

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ
БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ АЭРОКОСМИЧЕСКИЙ
УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)» (СГАУ)

Информатика

Электронный учебно-методический комплекс
по дисциплине в LMS Moodle

САМАРА
2012

УДК 004

Автор-составитель: **Сопченко Елена Вильевна**

Информатика [Электронный ресурс] : электрон. учеб.-метод. комплекс по дисциплине в LMS Moodle / Минобрнауки России, Самар. гос. аэрокосм. ун-т им. С. П. Королева (нац. исслед. ун-т); авт.-сост. Е.В.Сопченко. - Электрон. текстовые и граф. дан. - Самара, 2012. – 1 эл. опт. диск (CD-ROM).

В состав учебно-методического комплекса входят:

1. Курс лекций.
2. Методические указания к лабораторным работам.
3. Задания на практические работы.
4. Тесты для промежуточного контроля знаний.
5. Тесты для итогового контроля знаний.
6. Вопросы к экзамену.
7. Список использованных источников
8. Рабочая программа.

УМКД «Информатика» предназначен для студентов факультета информатики, обучающихся по направлениям подготовки бакалавров 010300.62 «Фундаментальная информатика и информационные технологии», 230100.62 «Информатика и вычислительная техника» и специальности 090303.65 Информационная безопасность автоматизированных систем в 1 семестре.

УМКД разработан на кафедре программных систем.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Самарский государственный аэрокосмический университет
имени академика С.П. Королёва (национальный исследовательский
университет)» (СГАУ)

Факультет информатики
Кафедра программных систем

Сопченко Е.В.

Курс лекций
«Информатика»

Учебное пособие

для студентов, обучающихся:

по направлению 010300.62 «Фундаментальные
информатика и информационные технологии»

по направлению 230100.62 «Информатика
и вычислительная техника»

по специальности 090303.65 «Информационная
безопасность автоматизированных систем»

Самара 2012

Содержание

Лекция 1 Информатика и ее связь с другими науками	4
Информационные революции	4
Основные подходы к определению информации.....	5
Информация и данные. Адекватность информации	5
Лекция 2 Знаковые системы. Семиотические аспекты информатики.....	6
Знаковые системы.....	6
Семиотические аспекты информатики	6
Адекватность информации	7
Меры информации.....	7
Лекция 3 Показатели качества информации. Классификация информации	10
Показатели качества информации	10
Классификация информации	10
Лекция 4 Сигнал: кодирование, декодирование. Информационные процессы и системы	12
Сигнал: кодирование, декодирование	12
Информационные процессы и системы	13
Лекция 5 Структурная схема ПК. Основные устройства ПК и их назначение. Память. Понятие чтения-записи.....	15
Структурная схема ПК	15
Системный блок: основные устройства и их назначение.....	16
Устройства обработки информации	17
Функции микропроцессора.....	17
Память.....	18
Лекция 6 Представление информации в компьютере. Системы счисления.....	20
Десятичная система счисления	20
Двоичная система счисления.....	20
Восьмеричная система счисления.....	22
Шестнадцатеричная система счисления	23
Лекция 7 Варианты представления информации в ПК. Кодирование текстовой, графической и звуковой информации	24
Кодирование числовой информации в ПК.....	24
Кодирование текстовых и символьных данных	27
Кодирование графических данных	28
Кодирование звуковой информации.....	30
Лекция 8 Основные подходы к определению количества информации. Понятие энтропии. Формула Шеннона, формула Хартли	31
Неопределенность, количество информации и энтропия.....	32
Формула Шеннона.....	34
Формула Хартли	35
Основные свойства логарифма.....	36
Количество информации, получаемой в процессе сообщения	37
Лекция 9 Элементы алгебры логики. Основные логические функции. Законы преобразования логических выражений.....	39
Основные понятия	39
Основные законы преобразования логических выражений	40
Лекция 10 Методы работы с процедурной информацией. Постановка задачи. Моделирование информации.....	41
Постановка задачи	41
Моделирование информации	42
Лекция 11 Построение алгоритма решения задачи.....	44
Графическая форма представления алгоритмов.....	44

Лекция 12 Стадии разработки программного продукта	47
Программирование решения задачи	47
Стадии разработки программного продукта.....	47
Схема взаимодействия специалистов, связанных с созданием и эксплуатацией программ ...	48
Характеристики программного продукта	49
Основные характеристики программ	50
Показатели качества программного продукта (ПП)	50
Лекция 13 Программное обеспечение компьютера	51
Состав и назначение программного обеспечения	51
Системное программное обеспечение.....	51
Операционные системы	53
Лекция 14 Базовые типы данных. Основные алгоритмы обработки.....	55
Простые типы данных	56
Основные алгоритмы обработки.....	56
Лекция 15 Вычислительные сети	58
Обобщенная структурная схема вычислительной сети	58
Топология сети.....	60
Лекция 16 Компьютерные вирусы и методы защиты информации от них	61
Классификация вирусов.....	61
Антивирусные программы.....	62
Лекция 17 Понятие несанкционированного доступа. Методы защиты информации.....	63
Программные системы защиты.....	63
Правовые методы защиты ПП и БД.....	63
Типы лицензий на программный продукт	64
Лекция 18 Криптографические методы защиты информации	66
Симметричные криптосистемы.....	66
Криптосистемы с открытым ключом.....	68
Электронная подпись	70
Управление ключами	71

Лекция 1

Информатика и ее связь с другими науками

Информационные революции

Изучение истории общества показывает, что проблема сбора, накопления, обработки и использования различных сведений и данных стояла перед человеком на всех этапах его развития. В течение длительного времени основными механизмами ее решения были органы чувств человека. Умственная деятельность, речь, слух помогали обрабатывать, хранить и передавать информацию от человека к человеку.

В истории развития цивилизации произошло несколько кардинальных изменений в сфере обработки информации, и, как следствие, преобразований общественных отношений, т.е. информационных революций.

1. *Первая* революция связана с изобретением письменности. Это привело к гигантскому качественному и количественному скачку в деле передачи знаний между поколениями.

2. *Вторая* революция (середина XVI в.) вызвана изобретением книгопечатания, которое радикально изменило индивидуальное общество, культуру, организацию деятельности.

3. *Третья* (конец XIX в.) обусловлена открытием электричества, благодаря которому появились телеграф, телефон, радио, позволяющие оперативно передавать информацию в любом объеме.

4. *Четвертая* (начало 50-х гг. XX в.) связана с появлением ЭВМ и другой информационной техники. Появление ЭВМ открыло эру механизации и автоматизации умственного труда, позволило хранить и обрабатывать огромные массы информации.

5. *Пятая* (70-е гг. XX в.) связана с изобретением микропроцессорной техники и появлением ПК. На микропроцессорах и интегральных схемах создаются компьютеры, компьютерные сети, системы передачи данных (информационные коммуникации). Изобретение ПК вызвало переход к широкому применению ЭВМ в информационных процессах в масштабах всего общества.

Именно масштабность преобразований, огромные возможности современного компьютерного мира вызвали бурное развитие науки информатики.

Термин *информатика* возник в 60-х годах во Франции. Французский термин *informatique* (информатика) образован путем слияния слов *information* (информация) и *automatique* (автоматика) и означает «автоматизированная переработка информации». В англоязычных странах этому термину соответствует синоним *computer science* (наука о компьютерной технике).

Информатику нельзя трактовать как предмет, изучающий лишь персональный компьютер.

Информатика и связанные с ней информационные технологии - необходимый атрибут профессиональной пригодности в современном обществе.

• **Информатика** - это фундаментальная наука, формирующая мировоззрение в информационной сфере и умение целенаправленно работать с информацией с помощью различных средств, в том числе, с помощью средств вычислительной техники. Информатика изучает процессы получения, хранения, обработки, передачи и защиты информации.

Информатика как наука формируется в настоящее время. Разрабатывается и осмысливается предмет науки, научные методы и средства научных исследований. В то же время информационный подход к изучению объектов самой различной природы играет большую роль в современном научном познании.

Основные подходы к определению информации

Ключевым понятием, изучаемым информатикой является *информация*.

|| Термин **информация** происходит от латинского *informato*, т.е. разъяснение, осведомление, изложение.

Относительная независимость информации от вещественно-энергетических носителей, возможность ее передачи и преобразование различными по физической природе сигналами безотносительно к ее содержанию, позволили объяснить некоторые процессы природы (например, наследственность) более адекватно, чем с помощью понятий вещества и энергии. Все это породило научно-философский интерес к информатике. Одним из центральных вопросов научных исследований является установление общности информационных процессов в органических и неорганических системах, в частности - между функциями человеческого мозга и технических систем.

В настоящее время сложились два подхода к определению информации.

■ **Философско-методологический.**

Согласно этому подходу, информация присутствует во всех материальных системах как органической, так и неорганической природы и является следствием свойства отражения - везде, где есть отражение, есть передача информации.

■ **Функциональный, или кибернетический подход.**

Согласно этому подходу, информация является свойством общества, живых существ и кибернетических устройств. Способность не только отражать внешний мир, но и использовать полученную информацию помогает живому организму адаптироваться к внешним условиям, обучаться и образовывать сложные сообщества.

Информация и данные. Адекватность информации

Наряду с понятием «*информация*» часто употребляют понятие «*данные*».

• **Данные** - это признаки или наблюдения, которые только хранятся, но не используются. В случае использования данных для уменьшения неопределенности о чем-то данные превращаются в информацию.

Лекция 2

Знаковые системы.

Семиотические аспекты информатики

Знаковые системы

Независимо от природы информации информационные процессы реализуются посредством знаковых систем.

• Под **знаком** понимается материальный предмет (явление, событие), который служит представлением другого предмета и используется для приобретения, хранения и передачи информации о последнем.

Примерами знаковых систем являются языки (русский, английский, китайский и т. д.), ноты, цифры, знаки дорожного движения и т. д.

Знаки делятся на символы - которые отображают значение знака (указатель направления - стрелка, геральдический знак (герб)) и диокритики - которые не имеют прямой зависимости между формой и значением или такая связь утрачена в результате изменения формы знака или переноса его значения на другие сущности. Это - буквы русского алфавита и арабские цифры, знаки математических операций. Если проследить историю букв, то можно определить связь их современного начертания с определенным значением (у древних финикийцев, изобретателей фонетической системы письма, на основании которой созданы латинский алфавит и кириллица, форма буквы - это стилизованное изображение предмета, название которого начинается с этой буквы). Часто знаки являются и символами и диокритиками одновременно (знаки дорожного движения).

Знаки и знаковые системы изучает наука семиотика.

Предметом семиотики являются вопросы связи знаков друг с другом и с представляемыми ими предметами и явлениями внешнего мира.

Семиотические аспекты информатики

В семиотике выделяют 3 аспекта: синтаксис, семантика и прагматика.

Синтаксис знаковых систем занимается изучением самих знаков.

Семантика изучает отношение между знаком и тем, что он представляет. Даже в простейшем случае между знаком и его обозначением есть субъект, который интерпретирует этот знак.

С точки зрения семантики каждое сочетание знаков является осмысленным или бессмысленным, причем осмысленность может допускать и неправильные с точки зрения синтаксиса построения (например, детская речь).

Однако истинность или ложность того или иного высказывания, т.е. достоверность отражения действительности знаковой системой, относится к области прагматики.

Прагматика изучает законы функционирования знаковой системы как средства коммуникации между субъектами. Она находится на стыке лингвистики, психологии, социологии и других наук.

Семиотические аспекты информатики очевидны. Конкретные приложения семиотики находят свою реализацию в знаковых системах кодирования

информации, в алгоритмических языках программирования, в автоматическом анализе и переводе текстов в системах искусственного интеллекта и т. д.

Для реализации информационных процессов особенно важны искусственные знаковые системы, которые созданы на формальной основе (например, алгоритмические языки программирования).

Основные отличия искусственных языков от естественных языков:

1. Отсутствие многозначности, каждая лексическая единица имеет ровно один смысл и наоборот.

2. Практическое отсутствие перефразирования, т.е. изменения формы высказывания при полном сохранении смысла.

3. Письменное представление.

4. Отсутствие избыточности.

5. Наличие четкой связи между лексической единицей языка и смыслом, который она представляет (оператор сравнения, оператор действия).

Адекватность информации

Основным свойством информации является ее адекватность.

• ***Адекватность информации*** - уровень соответствия информационного образа реальному объекту, процессу, явлению.

В соответствии с аспектами семиотики адекватность информации выражаются в трех формах:

а) синтаксическая (формально-структурные характеристики информации, не затрагивающие ее смысл). Обычно на таком уровне информацию называют данными (обезличенной информацией);

б) семантическая определяет степень соответствия образа объекта и самого объекта. При этом устанавливаются связи, отображающие смысл информации;

в) прагматическая - соответствует информации, цели её потребления (ценность, полезность информации и т. д.).

Для измерения информации используются два параметра:

- объём данных V_d - количество символов (разрядов) в сообщении;

- количество информации I - смысловое содержание данных.

В соответствии с формами адекватности разработаны меры информации.

Меры информации

1) Синтаксическая мера.

Количество информации I на синтаксическом уровне невозможно определить без рассмотрения понятия неопределённости состояния системы (энтропии).

Действительно, получение информации всегда связано с изменением степени неосведомлённости.

Пусть первоначально потребитель имеет априорные сведения о системе α . Мерой его неосведомлённости является функция $H(\alpha)$.

Для получения некоторого сообщения β получатель приобрёл некоторую дополнительную информацию $I_\beta(\alpha)$, уменьшившую его априорную

неосведомлённость так, что апостериорная (после β) неопределённость состояния системы стала $H_\beta(\alpha)$.

Тогда: $I_\beta(\alpha) = H(\alpha) - H_\beta(\alpha)$, т.е. количество информации измеряется изменением (уменьшением) неопределённости состояния системы.

Если в конце концов $H_\beta(\alpha)=0$, то первоначальное неполное знание заменяется полным знанием и $I_\beta(\alpha)=H(\alpha)$. Иными словами энтропия системы $H(\alpha)$ может быть мерой недостающей информации.

$H(\alpha)$ системы, имеющей N возможных состояний, согласно формуле Шеннона

$$H \stackrel{\text{Ш}}{=} - \sum_{i=1}^N P_i \cdot \log P_i,$$

где P_i - вероятность того, что система находится в i -м состоянии.

Для случая, когда $P_i = \frac{1}{N}$ (равновероятно)

$$H \stackrel{\text{Ш}}{=} - \sum_{i=1}^N \frac{1}{N} \cdot \log \frac{1}{N}.$$

В компьютере информация кодируется в той или иной системе счисления, тогда $N = m^n$, где m - основание системы счисления, n - число разрядов в сообщении.

Коэффициент (степень) информативности сообщения определяется отношением:

$$Y = \frac{I}{V_D} \quad \frac{\text{количество информации}}{\text{объем информации}}$$

2) Семантическая мера информации.

Для измерения смыслового содержания информации, т.е. её количества на семантическом уровне, наибольшее признание получила тезаурусная мера, которая связывает семантические свойства информации со способностью пользователя принимать поступившее сообщение.

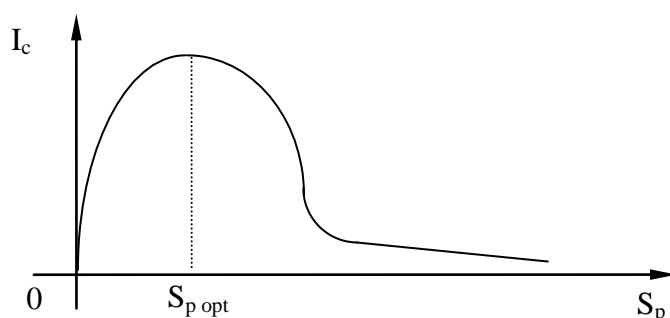
• **Тезаурус** - это совокупность сведений, которыми располагает пользователь или система.

Пусть S -смысловое содержание информации.

S_p -тезаурус пользователя.

I_c - количество семантической информации, воспринимаемой пользователем и включаемое им в тезаурус.

Тогда характер зависимости I_c от S_p имеет вид:



При $S_p = 0$ пользователь не воспринимает, не понимает поступающую информацию, так как у него нет никаких знаний.

При $S_p \rightarrow \infty$ (пользователь всё знает, и поступающая информация ему не нужна).

Максимальное количество семантической информации I_c пользователь приобретает при согласовании ее смыслового содержания S с тезаурусом S_p ($S_p = S_{p\ opt}$), когда поступающая информация понятна пользователю и несет ему ранее не известные сведения (отсутствующие в тезаурусе).

Таким образом, количество семантической информации в сообщении является величиной относительной. Одно и то же сообщение может иметь смысловое содержание для компетентного пользователя и быть бессмысленным для некомпетентного.

Коэффициент содержательности семантической информации.

$$C = \frac{I_c}{V_D} \quad - \text{ мера семантической информации.}$$

3) Прагматическая мера информации.

Эта мера определяет полезность (ценность) информации для достижения пользователем поставленной цели. Это тоже величина относительная, обусловленная особенностью использования информации. Ценность информации измеряется в тех же единицах, что и целевая функция (денежное выражение, время обработки информации, емкость памяти и т.д.).

Лекция 3

Показатели качества информации. Классификация информации

Показатели качества информации

1. Репрезентативность - правильность отбора и формирования информации.
2. Содержательность - $C = \frac{I_c}{V_D}$ - семантическая емкость.
3. Достаточность (полнота) - минимальный, но достаточный для принятия правильного решения состав информации.
4. Доступность - возможность выполнения процедур получения и преобразования информации.
5. Актуальность - степень сокращения ценности информации.
6. Своевременность - поступления не позже заранее назначенного момента времени.
7. Точность - степень близости получаемой информации реальному состоянию объекта.
8. Достоверность - свойство отражать реально существующие объекты с необходимой точностью (доверительная вероятность).
9. Устойчивость - способность реагировать на изменение исходных данных без нарушения необходимой точности.

Классификация информации

Любой объект реального мира характеризуется информацией различного рода.



Место возникновения.

Входная информация - информация, поступающая на вход системы.

Выходная информация - информация, выходящая из системы.

Внутренняя информация возникает внутри объекта.

Внешняя информация образуется за пределами объекта.

Стабильность.

Постоянная (условно-постоянная) информация - это неизменная и многократно используемая в течение длительного периода времени информация.

Переменная информация - это текущая информация, отражающая изменения количественных и качественных характеристик объекта.

Стадия обработки.

Первичная информация возникает непосредственно в процессе деятельности объекта и регистрируется на начальной стадии.

Вторичная информация получается в процессе обработки первичной информации и может быть промежуточной и результатной.

Промежуточная информация используется в качестве исходных данных для последующих расчетов.

Результатная информация получается в процессе обработки первичной и промежуточной информации.

Способ отражения.

Текстовая информация - совокупность алфавитных, цифровых и специальных символов.

Графическая информация - это различного рода графики, диаграммы, схемы, рисунки и т.д.

Лекция 4

Сигнал: кодирование, декодирование. Информационные процессы и системы

Сигнал: кодирование, декодирование

Сигнал – это знак, физический процесс или явление, несущие сообщение о каком-либо событии, состоянии объекта либо передающие команды управления, оповещения и т. д. Посредством сигналов можно представить любое, сколь угодно сложное событие [19].

По своей физической природе сигнал может быть механическим (например, деформация, изменение давления), тепловым (изменение температуры), световым (вспышка света, зрительный образ), электрическим (изменение силы тока, напряжения), электромагнитным (радиоволны), звуковым (акустические колебания) и др.

Сигналы могут преобразовываться (без изменения несомой ими информации) из одного вида в другой, более удобный для последующей передачи, восприятия, хранения, переработки либо целенаправленного изменения информации, содержащейся в сообщении; преобразование непрерывных сигналов в дискретные называется квантованием сигнала (при этом неизбежна некоторая потеря информации).

Понятие сигнала впервые было чётко сформулировано в кибернетике — как единство четырёх компонентов, непременно присутствующих в сигнале, несущем информацию о конкретном событии: физического носителя сигнала; формы выражения сигнала (синтаксиса); смысла интерпретации сигнала (семантики); правил приписывания различного смысла одному и тому же сигналу (прагматики). Задача установления общих закономерностей и взаимосвязи синтаксиса, семантики и прагматики решается семиотикой. Общие закономерности преобразования и передачи сигнала вне зависимости от их физической природы изучаются в теории информации (см. Информации теория).

Все исходные сигналы, поступающие от объекта, можно разделить на две большие группы [20]:

- сигналы статические, которые отображают устойчивые состояния некоторых объектов и могут быть представлены, например, в виде определенного положения элемента системы, текста в документе, определенного состояния электронного устройства и т. д.,
- сигналы динамические, для которых характерно быстрое изменение во времени, отображающее, например, изменение электрических параметров системы.

Динамические и статические сигналы имеют свои области использования. Статические сигналы существенное место занимают при подготовке, регистрации и хранении информации. Динамические используются в основном для передачи информации.

По характеру изменения сигналов во времени различают сигналы непрерывные и дискретные. Непрерывный сигнал отображается некоторой непрерывной функцией и физически представляет собой непрерывно

изменяющиеся значения колебаний. Дискретный сигнал характеризуется конечным множеством значений и в зависимости от исходного состояния принимает значения, связанные с определенным состоянием системы.

- **Кодирование** – это представление одного набора знаков другим с помощью кода.

Процедуры кодирования используются для обеспечения возможности передачи сообщений по каналам связи.

- **Код** – это правило отображения одного набора объектов или знаков в другой набор знаков без потери информации.

Кодирование должно быть однозначным, т.е. между символом и его кодом должно быть установлено взаимнооднозначное соответствие. Такое соответствие называют **кодовой таблицей**.

С помощью кодирования сообщение представляется в форме, которая позволяет осуществить передачу его по каналам связи. Дискретное сообщение можно изобразить в виде некоторой последовательности цифр или букв, при этом каждая цифра или буква представляет собой одно сообщение. С помощью кода каждая цифра или буква отображаются некоторым набором импульсов, которые составляют кодовую комбинацию.

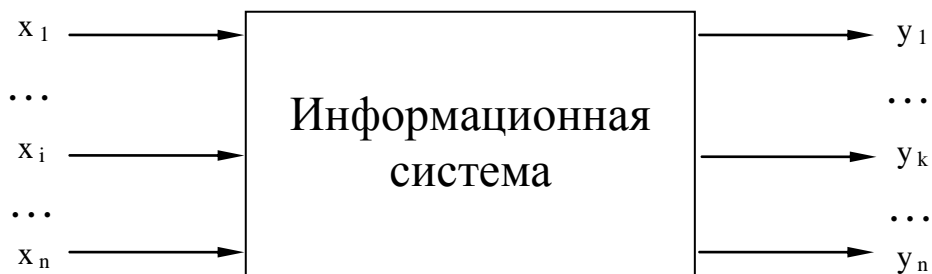
Основное требование, предъявляемое к кодовым комбинациям, состоит в возможности различения их на приемной стороне при определенных воздействиях помех в каналах связи [20].

Информационные процессы и системы

Все чаще в настоящее время в силу профессиональной необходимости люди становятся компонентами различных информационных систем.

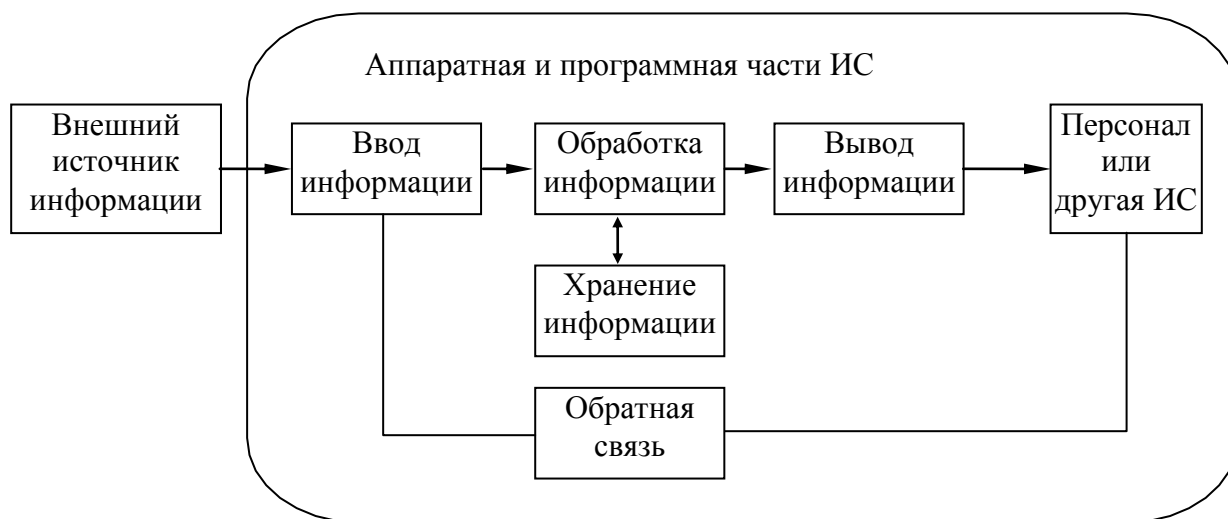
- **Информационная система** - совокупность средств, методов и персонала, используемых для приема, хранения, обработки и передачи информации, т.е. для организации информационных процессов.

Любая информационная система может быть представлена в виде «черного ящика», у которого на вход подается совокупность n входных символов x_i , а на выходе образуется некоторая совокупность из m выходных сигналов y_i .



- **Информационным процессом** называют процесс установления связи между двумя объектами материального мира: источником информации и приемником информации.

При этом информационные процессы возникают на любом из этапов взаимодействия компонентов информационной системы:



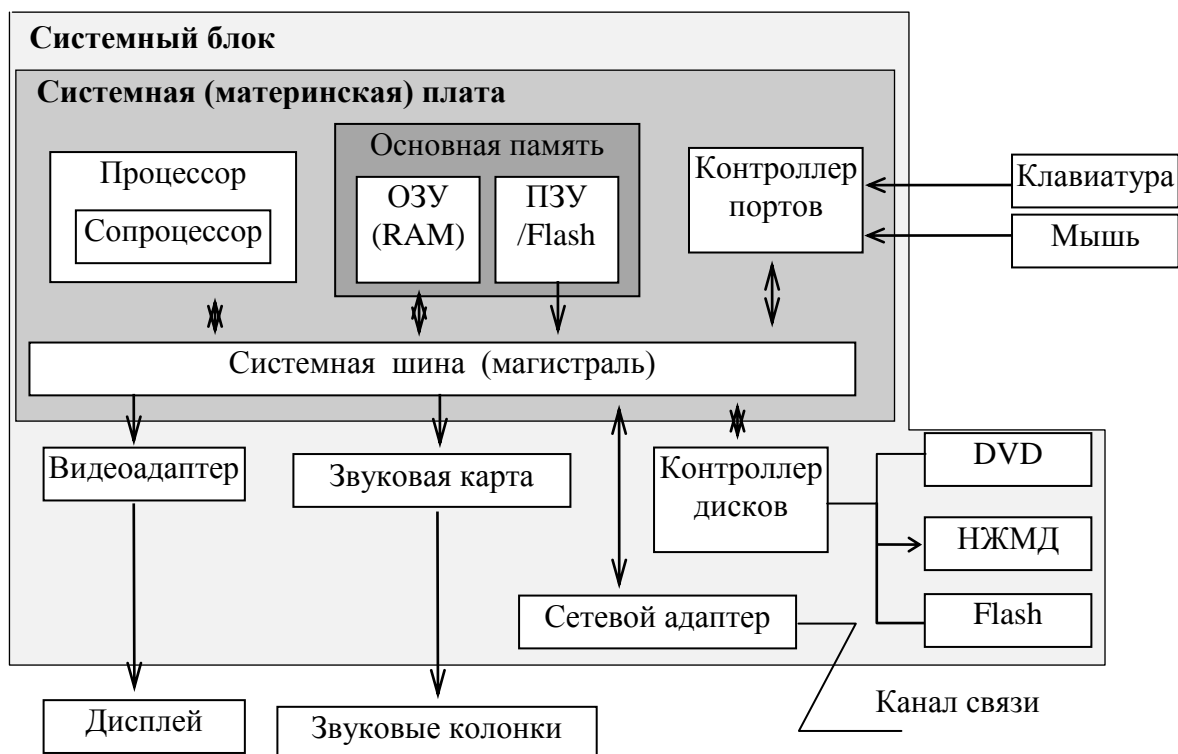
Компьютер является универсальной информационной системой, поскольку позволяет реализовать все известные информационные процессы:

- сбор (ввод информации);
- хранение;
- обработка;
- вывод;
- передача информации.

Лекция 5

Структурная схема ПК. Основные устройства ПК и их назначение. Память. Понятие чтения-записи

Структурная схема ПК



ОЗУ - оперативное запоминающее устройство;
ПЗУ - постоянное запоминающее устройство;
НЖМД - накопитель на жестком магнитном диске;
DVD - устройство для чтения компакт и DVD дисков;
Flash - устройство для чтения *Flash*-памяти.

На структурной схеме персонального компьютера можно выделить следующие блоки, отвечающие за выполнение основных информационных процессов:

- системный блок (хранение и обработка информации);
- устройства ввода информации (клавиатура, мышь);
- устройства вывода информации (дисплей, звуковые колонки);
- устройства передачи информации (сетевой адаптер, модем).

Системный блок: основные устройства и их назначение

Системный блок включает в себя:

- системную (материнскую) плату;
- контроллеры внешних устройств;
- устройства хранения информации.

На системной плате находятся устройства, называемые **внутренними**. Все устройства на плате подключаются к системной шине (магистралам).

• **Шина** - набор проводов, конструктивно выполненных в едином корпусе (пластмассовой обмотке) и предназначенных для совместного использования.

Системная шина включает в себя:

- *шину данных*, используемую для параллельной передачи всех разрядов числового кода операнда;
- *шину адреса*, предназначенную для параллельной передачи всех разрядов числового кода адреса ячейки ОП или порта ввода-вывода внешнего устройства;
- *шину управления*, используемую для передачи управляющих сигналов во все блоки ПК;
- *шину питания*, предназначенную для подключения блоков ПК к системе электропитания.

Внутри системного блока встроен блок питания. Он преобразует входное напряжение 220 В в напряжение в диапазоне -5 В до +5 В, необходимое для правильного функционирования схем компьютера.

Системная шина обеспечивает 3 направления передачи информации:

- 1) между МП и основной памятью (ОП);
- 2) между МП и портами ввода-вывода внешних устройств;
- 3) между ОП и портами ввода-вывода внешних устройств (в режиме прямого доступа к памяти).

Управление системной шиной осуществляется либо непосредственно, либо через контроллер шины. Обмен информацией между внешними устройствами и системной шиной выполняется с использованием ASCII-кодов.

Шины адреса, данных и управления служат для организации взаимодействия между устройствами ПК.

Каждое устройство имеет свой уникальный адрес в компьютере.

Когда процессору необходимо обменяться информацией с каким-либо устройством, то сначала он выставляет на шину адреса адрес этого устройства. Затем по шине управления процессор запрашивает о готовности устройства к обмену, если устройство готово (т.е. выставило на шину управления сигнал подтверждения готовности), то процессор выставляет данные на шину данных. Когда обмен информацией закончен, сигнал готовности устройства снимается, шины адреса и данных вновь свободны и готовы к новому обмену информацией.

Остальные устройства, расположенные вне системной платы называются **внешними** (в том числе и накопители на жестких и гибких магнитных дисках). Для подключения внешних устройств к системной шине всегда используются специальные адаптирующие устройства, называемые **контроллерами** (картами, адаптерами) внешних устройств.

• **Внешние устройства** (ВУ) обеспечивают взаимодействие ПК с окружающей средой: пользователями, объектами управления и другими компьютерами.

ВУ разделяются на следующие классы:

- внешние запоминающие устройства (ВЗУ);
- устройства ввода информации;
- устройства вывода информации;
- средства связи и телекоммуникации.

Устройства обработки информации

Основным устройством обработки информации является процессор.

• **Микропроцессор** (МП) – центральный блок ПК, предназначенный для управления работой всех блоков машины и для преобразования информации, т.е. выполнения арифметических и логических операций над информацией.

Первый МП был выпущен в США в 1971 году фирмой Intel – МП 4004. В настоящее время МП фирмы Intel и Intel-подобные являются наиболее популярными и распространенными.

Процессор является “мозгом“ компьютера. Он состоит из двух основных устройств - устройства управления (УУ) и арифметико-логического устройства (АЛУ).

УУ формирует сигналы управления АЛУ, памятью, внешними устройствами, выдает эти сигналы на шину управления. Кроме этого, оно принимает с шины управления сигналы от внешних устройств и обрабатывает их. УУ следит за ходом выполнения арифметических и логических операций.

АЛУ выполняет арифметические и логические операции над числовой и символьной информацией.

Устройство для выполнения высокоскоростных операций для чисел с плавающей точкой - сопроцессор - встроен внутрь процессора, представляет вместе с процессором единый функциональный блок, имеет свою систему команд и работает параллельно с основным МП, но под его управлением.

• **Внутренние регистры** – служат для кратковременного хранения, записи и выдачи информации, непосредственно используемой в вычислениях в ближайшее время работы ПК.

• **Регистры** – быстродействующие ячейки памяти различной длины (8÷64 разрядов).

Функции микропроцессора

1. Чтение и дешифрация команд из ОП.
2. Чтение данных из ОП и регистров адаптеров ВУ.
3. Прием и обработка запросов и команд от адаптеров на обслуживание ВУ.
4. Обработка данных и их запись в ОП и регистры адаптеров ВУ.
5. Выработка управляющих сигналов для всех узлов и блоков ПК.

Разрядность шины данных МП определяет разрядность ПК в целом; разрядность шины адреса МП – его адресное пространство.

• **Адресное пространство** – это максимальное количество ячеек ОП, которое может быть непосредственно адресовано микропроцессором.

Адресация ячеек памяти может быть реализована по-разному; существуют 4 типа адресации ОП: (зависит от процессора и ОП)

1. Линейное адресное пространство - каждый байт пронумерован подряд - это и есть адрес.
2. Сегментированное адресное пространство, вся память разделена на блоки (сегменты) переменной длины, а сегмент на байт.
3. Страничное адресное пространство - вся память разделена на блоки (страницы) постоянной длины, а страницы - на байты.
4. Сегментно-страничное адресное пространство - вся память разделена на сегменты, которые разделены на страницы, которые содержат пронумерованные байты.

Память

• **Запоминающее устройство** или просто **память** предназначена для хранения информации или команд программ.

Вся информация внутри ЭВМ хранится в виде нулей и единиц. Самой маленькой ячейкой памяти является 1 бит. Один бит хранит либо 0 или 1.

8 бит = 1 байт

• **1 байт**- это минимальное количество информации, которое можно очистить или записать в память.

Любая память представляет из себя совокупность байтов.

Каждый байт имеет свой уникальный адрес. Чтобы извлечь информацию из этого байта или записать ее туда, необходимо указать адрес этого байта.

• **Запись в память** - это размещение информации по указанному адресу и хранение ее там определенное время. При этом предыдущая информация, хранившаяся в этой ячейке памяти, стирается. Новая информация будет храниться в этой ячейке памяти до тех пор, пока туда не будет записана другая информация.

• **Чтение из памяти** - это выборка информации из ячейки памяти по указанному адресу. При этом копия информации передается в требуемое устройство, а сама информация остается в ячейке памяти. Таким образом, к указанной ячейке памяти можно обращаться сколько угодно большое число раз до тех пор, пока в эту ячейку не будет записана новая информация.

Любая память характеризуется следующими параметрами:

- **емкость** - максимальное количество хранимой информации, выраженное в байтах.

1 Кбайт = 1024 байт = 2^{10} байт,

1 Мегабайт = 1024К = 2^{20} байт,

1 Гигабайт = 1024М = 2^{30} байт.

С помощью одного байта можно закодировать $2^8 = 256$ символов, но хранится в 1 байте будет только 1 символ.

- *быстродействие* - это время обращения к памяти определяемые временем считывания или времени записи.

В зависимости от этих характеристик всю память можно разделить на внутреннюю и внешнюю.

Внутренняя память обладает наибольшим быстродействием и ограниченной емкостью.

Внешняя память предназначена для долговременного хранения информации вне зависимости от того, включено ли питание у компьютера или нет.

Лекция 6

Представление информации в компьютере. Системы счисления

• **Система счисления** - способ наименования и изображения чисел с помощью символов, имеющих определенные количественные значения.

В зависимости от назначения и применения различают внутренние коды (для представления данных в ЭВМ), коды, предназначенные для обмена данными и их передачи по каналам связи и коды для специальных применений.

Внутренние коды базируются на использовании позиционных систем счисления с основанием P . В общем случае число Z_P может быть представлено в виде:

$$Z_P = A_N * B^N + A_{N-1} * B^{N-1} + \dots + A_1 * B^1 + A_0 * B^0,$$

где N - номер позиции или разряд.

A_0, A_1, \dots, A_N - разрядные коэффициенты, которые могут принимать значения цифр, соответствующей системы счисления.

Основные системы счисления, используемые для представления информации в ЭВМ

Система счисления	Основание P	Символы A_i
Двоичная	2	0, 1
Восьмеричная	8	0,1,2,3,4,5,6,7
Десятичная	10	0,1,2,3,4,5,6,7,8,9
Шестнадцатеричная	16	0,1,2,3,4,5,6,7,8,9, A, B, C, D, E, F 10 11 12 13 14 15

Десятичная система счисления

Используемые символы: 0,1,2,3,4,5,6,7,8,9.

Примеры чисел: 245_{10} , 38_{10} , 1379_{10} .

Степени числа 10:

$$10^0 = 1$$

$$10^1 = 10$$

$$10^2 = 100 \text{ и т.д.}$$

Нумерация разрядов начинается с нуля справа налево.

Представление числа в виде степеней числа 10.

$$245_{10} = 2 \cdot 10^2 + 4 \cdot 10^1 + 5 \cdot 10^0.$$

Двоичная система счисления

Используемые символы: 0,1.

Примеры чисел: 10111_2 , 1111_2 , 01101010_2 .

Степени числа 2:

$2^0 = 1$	$2^5 = 32$	$2^{10} = 1024$
$2^1 = 2$	$2^6 = 64$	$2^{11} = 2048$
$2^2 = 4$	$2^7 = 128$	$2^{12} = 4096$
$2^3 = 8$	$2^8 = 256$	
$2^4 = 16$	$2^9 = 512$	

Нумерация разрядов начинается с нуля справа налево.

Представление числа в виде степеней числа 2.

$$10111_{10} = 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 16 + 4 + 2 + 1 = 23_{10}.$$

Перевод из десятичной системы счисления в двоичную: необходимо представить десятичное число в виде суммы степеней числа 2; если какая-либо степень отсутствует в сумме, в соответствующем разряде двоичного числа будет 0, если присутствует, то 1.

Например:

$$456 = 256 + 128 + 64 + 8 = 2^8 + 2^7 + 2^6 + 2^3 = \begin{matrix} & & & & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ & & & & & & & & & & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 2. \end{matrix}$$

Правила сложения двоичных чисел:

$$\begin{aligned} 0 + 0 &= 0 \\ 0 + 1 &= 1 \\ 1 + 0 &= 1 \\ 1 + 1 &= (1) 0 \end{aligned}$$

Примеры.

Представление отрицательных чисел в двоичном коде.

Для того, чтобы получить отрицательное двоичное число, необходимо:

- 1) инвертировать все разряды положительного двоичного числа;
- 2) прибавить к младшему разряду единицу.

Пример: Вычислить в двоичной форме.

$$123_{10} - 38_{10} = 85_{10}.$$

1 этап. Переводим числа 123 и 38 в двоичную форму.

$$123_{10} = 64 + 32 + 16 + 8 + 2 + 1 = 2^6 + 2^5 + 2^4 + 2^3 + 2^1 + 2^0 = 1111011$$

$$38_{10} = 32 + 4 + 2 = 2^5 + 2^2 + 2^1 = \begin{matrix} & & & & 5 & 4 & 3 & 2 & 1 & 0 \\ & & & & 1 & 0 & 0 & 1 & 1 & 0 & 2 \end{matrix} = \begin{matrix} & & & & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ & & & & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 2 \end{matrix}$$

(Выравниваем количество разрядов).

2 этап. Переводим 38_{10} в -38_{10} :

- 1) инвертируем все разряды

$$0100110$$

↓

$$1011001$$

- 2) прибавляем к младшему разряду единицу

$$1011001$$

$$\underline{0000001}$$

$$1011010$$

$$-38_{10} = 1011010_2$$

3 этап. Выполняем сложение $123_{10} + - 38_{10}$:

$$\begin{array}{r} 1111011 \\ + 1011010 \\ \hline (1)1010101 \end{array} \quad (\text{старшая единица в разряд переноса})$$

4 этап. Переводим $1010101_2 = 2^6 + 2^4 + 2^2 + 2^0 = 64 + 16 + 4 + 1 = 85_{10}$.

Умножение двоичных чисел:

Пример: $15 \cdot 15 = 225$.

$$\begin{array}{r} 1111 \\ 1111 \\ \hline 1111 \\ 1111 \\ 1111 \\ \hline 1111 \\ 1111 \\ \hline 11100001 \end{array}$$

Восьмеричная система счисления

Используемые символы: 0,1,2,3,4,5,6,7.

Примеры чисел: 123_8 , 10_8 , 437_8 .

Степени числа 8:

$$8^0 = 1$$

$$8^1 = 8$$

$$8^2 = 64$$

$$8^3 = 512$$

$$8^4 = 4096$$

Нумерация разрядов начинается с нуля справа налево.

Представление числа в виде степеней числа 8.

$$123_8 = 1 \cdot 8^2 + 2 \cdot 8^1 + 3 \cdot 8^0 = 64 + 16 + 3 = 83_{10}.$$

Перевод из десятичной системы счисления в восьмеричную: необходимо представить десятичное число в виде суммы степеней числа 8; при этом степени могут включены несколько раз.

$$500_{10} = 7 \cdot 8^2 + 6 \cdot 8^1 + 4 \cdot 8^0 = 764_8.$$

Перевод из восьмеричной системы счисления в двоичную: каждая восьмеричная цифра заменяется триадой (своим двоичным представлением в трех разрядах)

$$573_8 = 101\ 111\ 011_2$$

Перевод из двоичной системы счисления в восьмеричную: каждая триада заменяется на соответствующую восьмеричную цифру, при этом выделение триад начинается с младшего (нулевого) разряда. Недостающие старшие разряды дополняются нулями.

$$1\ 111\ 011\ 101\ 011_2 = 001\ 111\ 011\ 101\ 011_2 = 17353_8.$$

Шестнадцатеричная система счисления

Используемые символы: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F.

A - 10 D - 13

B - 11 E - 14

C - 12 F - 15

Примеры чисел: 123₁₆, A10₁₆, 4F₁₆, CD₁₆, E2A₁₆.

Степени числа 16:

$$16^0 = 1$$

$$16^1 = 16$$

$$16^2 = 256$$

$$16^3 = 4096$$

Нумерация разрядов начинается с нуля справа налево.

Представление числа в виде степеней числа 16.

$$E2A_8 = 14 \cdot 16^2 + 2 \cdot 16^1 + 10 \cdot 16^0 = 14 \cdot 256 + 2 \cdot 16 + 10 \cdot 1 = 3584 + 32 + 10 = 3626_{10}.$$

Перевод из десятичной системы счисления в шестнадцатеричную:

необходимо представить десятичное число в виде суммы степеней числа 16; при этом степени могут включены несколько раз.

$$901_{10} = 3 \cdot 16^2 + 8 \cdot 16^1 + 5 \cdot 16^0 = 385_{16}.$$

Перевод из шестнадцатеричной системы счисления в двоичную: каждая шестнадцатеричная цифра заменяется тетрадой (своим двоичным представлением в четырех разрядах)

$$2F3D_{16} = 0010\ 1111\ 0011\ 1101_2$$

Перевод из двоичной системы счисления в шестнадцатеричную: каждая тетрада заменяется на соответствующую шестнадцатеричную цифру, при этом выделение тетрад начинается с младшего (нулевого) разряда. Недостающие старшие разряды дополняются нулями.

$$11\ 1101\ 0001\ 1011_2 = 0011\ 1101\ 0001\ 1011_2 = 3D1B_8.$$

Лекция 7

Варианты представления информации в ПК. Кодирование текстовой, графической и звуковой информации

Виды информации, обрабатываемые в компьютере:

- числовая;
- текстовая,
- графическая,
- звуковая.

Несмотря на исходную форму, вся информация в компьютере представляется в числовой форме.

Кодирование числовой информации в ПК

Существует несколько вариантов представления чисел в ПК. Числа могут быть целые и дробные, положительные и отрицательные.

Целые положительные числа от 0 до 255 можно представить непосредственно в двоичной системе счисления, при этом они будут занимать один байт в памяти компьютера.

Число	Двоичный код
0	0000 0000
1	0000 0001
2	0000 0010
3	0000 0011
...	...
255	1111 1111

Целые отрицательные числа представлены особым образом: знак отрицательного числа кодируется обычно старшим битом, нуль интерпретируется как плюс, единица как минус. Поскольку один бит будет занят, то одним байтом могут быть закодированы целые числа в интервале от -127 до +127. Такой способ представления целых чисел называется **прямым кодом**.

Также существует способ кодирования отрицательных целых чисел в **обратном коде**. В этом случае положительные числа совпадают с положительными числами в прямом коде, а отрицательные получаются в результате вычитания из двоичного числа 1 0000 0000 соответствующего положительного числа, например, число -7 получит код 1111 1000. Целые числа больших диапазонов представляются в двухбайтовых и четырехбайтовых адресах памяти.

В вычислительных машинах применяются две формы представления **дробных двоичных чисел**:

- в естественной форме или форме с фиксированной запятой (точкой);
- в нормальной форме или форме с плавающей запятой (точкой).

С фиксированной запятой все числа изображаются в виде последовательности цифр с постоянным для всех чисел положением запятой, отделяющей целую часть от дробной.

Пример. Пусть число представлено в виде $m:n$, где m - фиксированное число разрядов в целой части числа (до запятой), n - фиксированное число разрядов в дробной части числа (после запятой).

Например, $m = 3$, $n = 6$, тогда числа, записанные в такую разрядную сетку, имеют вид:

- 213,560000; + 004,021025; - 000,007345.

Однако такое представление используется в основном для целых чисел, поскольку при выходе результата какой-либо операции за границы такой разрядной сетки дальнейшие вычисления теряют смысл.

С плавающей запятой все числа изображаются в виде двух групп цифр. Первая группа цифр называется мантиссой, вторая - порядком. Причем абсолютная величина мантиссы должна быть меньше 1, а порядок - целым числом.

В общем виде число в форме с плавающей запятой может быть представлено в виде:

$$N = \pm MP^{\pm r}$$

где M - мантисса числа ($|M| < 1$);

r - порядок числа (r - целое число);

P - основание системы счисления.

Пример. Числа из предыдущего примера имеют вид:

- 0,21356 $\cdot 10^{-3}$; + 0,4021025 $\cdot 10^{-1}$; - 0,734500 $\cdot 10^{-2}$.

Нормальная форма представления имеет огромный диапазон отображения чисел и является основой в современных ПК.

Кроме двоичной системы счисления также широкое распространение получила двоично-десятичная система счисления. В этой системе все десятичные цифры отдельно кодируются четырьмя двоичными цифрами и в таком виде последовательно записываются друг за другом.

Полям называют последовательность нескольких бит или байтов.

В ПК могут обрабатываться поля постоянной и переменной длины.

Поля постоянной длины:

1 байт;

слово - 2 байта;

двойное слово - 4 байта;

расширенное слово - 8 байт;

слово длиной 10 байт.

Поля переменной длины могут иметь любой размер от 0 до 256 байт, но обязательно кратный целому числу байтов.

Пример.

1) Двойное слово - 4 байта = 32 бита

1 бит	8 бит	23 бита
S	порядок	мантисса

2) Расширенное слово - 8 байт = 64 бита

1 бит	11 бит	52 бита
S	порядок	мантисса

3) Слово длиной 10 байт - 80 бит

1 бит	15 бит	64 бита
S	Порядок	мантисса

При этом S- поле знака:

если $S = 0$, число ≥ 0

если $S = 1$, число < 0 .

Двоично-десятичное кодирование чисел

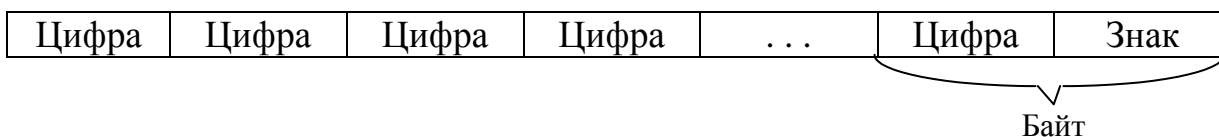
Двоично-кодированные десятичные числа могут быть представлены в ПК полями переменной длины в так называемых упакованном и распакованном форматах.

В упакованном формате для каждой десятичной цифры отводится по 4 двоичных разряда (полбайта), при этом знак числа кодируется в крайнем правом полубайте числа:

1100 – знак «+»,

1101 – знак «-».

Структура поля упакованного формата:



Упакованный формат используется обычно при выполнении операций сложения и вычитания двоично-десятичных чисел.

В распакованном формате для каждой десятичной цифры отводится по целому байту, при этом старшие полубайты каждого байта (кроме самого младшего) в ПК заполняются кодом 0011 (в соответствии с ASCII- кодом), а в младших (левых полубайтах) обычным образом кодируются десятичные цифры. Старший полубайт самого младшего (правого) байта используется для кодирования знака числа.

Распакованный формат используется обычно при выполнении операций умножения и деления двоично-десятичных чисел.

Распакованный формат представления двоично-десятичных чисел является следствием использования в ПК кода ASCII (American Standard Code for Information Interchange - Американский стандартный код для обмена информацией). Основным стандартом для кодирования символов использует шестнадцатеричные коды 00 - 7F, расширение стандарта - 80 - FF. Основным стандартом является международным и используется для кодирования управляющих символов, цифр и букв латинского алфавита; в расширении стандарта кодируются символы псевдографики и буквы национального алфавита.

Таким образом, можно составить обобщенную таблицу кодировки десятичных чисел в различных системах кодирования.

10 с/с	2 с/с	8 с/с	16 с/с	2/10 с/с	ASCII		
0	00000000	0	0	0000	0011	:	0000
1	00000001	1	1	0001	0011	:	0001
2	00000010	2	2	0010	0011	:	0010
3	00000011	3	3	0011	0011	:	0011
4	00000100	4	4	0100	0011	:	0100
5	00000101	5	5	0101	0011	:	0101
6	00000110	6	6	0110	0011	:	0110
7	00000111	7	7	0111	0011	:	0111
8	00001000	10	8	1000	0011	:	1000
9	00001001	11	9	1001	0011	:	1001
10	00001010	12	A	0001 : 0000	0011 :	0001 :	0011 : 0000
11	00001011	13	B	0001 : 0001	0011 :	0001 :	0011 : 0001
12	00001100	14	C	0001 : 0010	0011 :	0001 :	0011 : 0010
13	00001101	15	D	0001 : 0011	0011 :	0001 :	0011 : 0011
14	00001110	16	E	0001 : 0100	0011 :	0001 :	0011 : 0100
15	00001111	17	F	0001 : 0101	0011 :	0001 :	0011 : 0101

Кодирование текстовых и символьных данных

В двоичной системе счисления кодирование символов основывается на сопоставлении каждому из них определенной группы двоичных знаков. Двоичное кодирование символьных данных производится заданием кодовых таблиц, в которых каждому символу ставится в соответствие одно- или двухбайтовый код. Восемью двоичными разрядами достаточно для кодирования 256 различных символов. Этого количества достаточно, чтобы выразить все символы английского и русского алфавита, а также знаки препинания, символы основных арифметических операций и некоторые специальные символы [20].

Наиболее популярная таблица ASCII-кодов разработана в 1981 году.

Коды с 0 до 127 составляют базовую (основную) таблицу, коды со 128 по 255 — расширенную (дополнительную) таблицу. Дополнительная таблица отдана национальным алфавитам и символам псевдографики.

Символы с 0 до 31 относятся к служебным кодам. Если эти коды используются в символьном тексте программы, они считаются пробелами. При использовании в операциях ввода-вывода они имеют самостоятельное значение.

Аналогичные системы кодирования текстовых данных были разработаны и в других странах. Так, в СССР действовала система кодирования КОИ-8 (восьмизначный код обмена информацией). Компанией Microsoft была введена кодировка символов русского языка, известная как кодировка Windows-1251.

Во многих азиатских странах 256 кодов не хватило. В 1991 году производители программных продуктов (Microsoft, IBM, Apple) выработали единый стандарт Unicode 3.0. Этот код построен по 31-битной схеме. Все текстовые документы в этой кодировке вдвое длиннее, зато она содержит буквы латинского и многих национальных алфавитов, спецсимволы и т. п.

Кодирование графических данных

Различают три вида компьютерной графики: растровую, векторную и фрактальную. Они отличаются принципами формирования изображения при отображении на экране монитора или при печати на бумаге.

Если графические объекты формируются в виде множества точек (пикселей) разных цветов и разных яркостей, то это называется растровой графикой. В растровой графике общепринятым на сегодняшний день считается представление черно-белых иллюстраций в виде комбинации точек с 256 градациями серого цвета. Эти мельчайшие точки образуют характерный узор, называемый растром. Точно так же изображают информацию периферийные устройства печати. Основным элементом растрового изображения является точка. Если изображение экранное, то эта точка называется пикселем. В зависимости от того, на какое графическое разрешение экрана настроена операционная система компьютера, на экране могут размещаться изображения 640x480, 800x600, 1024x768 и более пикселей. При кодировании растровых изображений в памяти компьютера должна храниться информация о каждом пикселе. С размером изображения непосредственно связано его разрешение. Этот параметр измеряется в dpi (dots per inch — точек на дюйм).

У растровых изображений два основных недостатка. Во-первых, очень большие объемы данных. Для активных работ с большеразмерными иллюстрациями типа журнальной полосы требуются компьютеры с большими размерами оперативной. Во-вторых, растровые изображения невозможно значительно увеличить без серьезных искажений. Эффект искажения при увеличении точек раstra называется пикселизацией.

Расчет количества памяти, выделяемого под растровое изображение размером X на Y пикселей, производят по формуле:

$$\text{Память (в битах)} = X * Y * n,$$

где n – количество бит на 1 пиксель (глубина цвета).

Общее количество цветов **K** в этом случае вычисляют по формуле:

$$K = 2^n$$

В отличие от растровой *в векторной графике* изображение представляет собой совокупность простых элементов (графических примитивов): прямых линий, дуг, окружностей, эллипсов, прямоугольников и т. п. Положение и форма графических примитивов задаются в системе графических координат, связанных с экраном. В векторной графике объём памяти, занимаемой, какой-либо фигурой, не зависит от ее размеров, а зависит от количества параметров, по которым эта фигура будет строиться. Перед выводом на экран каждого объекта программа векторной графики производит вычисление координат экранных точек в изображении объекта. Векторная графика лишена недостатков растровой, но в ней сложно создавать художественные иллюстрации, поэтому чаще всего её используют для чертёжных и проектно-конструкторских работ.

Фрактальная графика, как и векторная, – вычисляемая, но отличается от неё тем, что никакие объекты в памяти компьютера не хранятся. Изображение строится по уравнениям, поэтому ничего, кроме формулы, хранить не надо. Изменение коэффициентов в уравнении позволяет получить совершенно другую картину.

Для кодирования цветных графических изображений применяется принцип декомпозиции – разложение произвольного цвета на основные составляющие. Существует множество различных типов цветовых моделей, но в компьютерной графике, как правило, применяется не более, трёх. Эти модели известны под названиями: RGB, CMYK и HSB [20]:

- **Цветовая модель RGB.** Любой цвет на экране (в проходящем свете) можно представить с использованием трех основных цветов: красного (Red), зеленого (Green) и синего (Blue). Совмещение всех трех цветов в равных пропорциях дает нейтральный серый цвет, который при большей яркости стремится к белому цвету, а при меньшей – к черному. Метод получения нового оттенка суммированием яркостей составляющих компонент называется *аддитивным*. Чаще всего на каждый цвет отводят по 1 байту (8 бит), что дает 24-битную глубину цвета. Режим представления цветной графики с использованием 24 двоичных разрядов называется *полноцветным* (True Color).

- **Цветовая модель CMYK.** Эту модель используют для подготовки печатных изображений, видимых в отражённом цвете. В этом случае увеличение количества краски приводит не к увеличению визуальной яркости, а к её уменьшению, т.к. более толстый слой краски поглощает больше света и меньше отражает. Поэтому для подготовки печатных изображений используется не аддитивная модель, а *субтрактивная* (вычитающая) модель. Цветовыми компонентами этой модели являются не основные цвета, а дополнительные, т. е. те, которые получаются в результате вычитания основных цветов из белого: голубой (Cyan), пурпурный (Magenta) и желтый (Yellow). Так как цветные красители по отражающим свойствам не одинаковы, то для повышения контрастности применяется ещё черный (Black) цвет. В типографиях цветные изображения печатаются в несколько приемов. Накладывая на бумагу поочередно голубой, пурпурный, жёлтый и чёрный отпечатки, получают полноцветную иллюстрацию.

- **Цветовая модель HSB.** Если модель RGB наиболее удобна для компьютера, модель CMYK – для типографии, то модель HSB наиболее удобна для человека. В модели HSB также три компонента: оттенок цвета (Hue), насыщенность (Saturation) и яркость цвета (Brightness). Регулируя эти три компонента, можно получить столь же много произвольных цветов, как и при работе с другими моделями. Эта модель удобна для применения в тех графических редакторах, которые ориентированы не на обработку готовых изображений, а на их создание. Значение цвета выбирается как вектор, выходящий из центра окружности. Точка в центре соответствует белому цвету, а точки по периметру – чистым цветам. Направление вектора определяет цветовой оттенок и задаётся в модели HSB в угловых градусах. Длина вектора определяет насыщенность цвета. Яркость цвета задается на отдельной оси, нулевая точка которой имеет черный цвет.

Создавать и обрабатывать цветные изображения принято в модели RGB. При печати рисунка RGB на цветном принтере драйвер принтера преобразует рисунок в цветовую модель CMYK. В случае, когда не требуется очень высокое качество отображения цвета, применяется режим High Color, который кодирует одну точку растра двумя байтами (16 разрядов дают 2^{16} цветов).

Режим, который при кодировании одной точки растра использует один байт, называется *индексным*, в нём различаются 256 цветов. Этого недостаточно, чтобы передать весь диапазон цветов. Код каждой точки при этом выражает собственно не цвет, а некоторый номер цвета из таблицы цветов, называемой *палитрой*. Палитра должна прикладываться к файлам с графическими данными и используется при воспроизведении изображения.

Кодирование звуковой информации

Методы работы со звуковой информацией пришли в вычислительную технику наиболее поздно. В итоге они далеки от стандартизации. Отдельные компании разработали свои корпоративные стандарты, однако можно выделить два основных подхода:

- **Метод частотной модуляции** (метод FM — Frequency Modulation) основан на разложении сигнала в виде суперпозиции элементарных гармоник с разными фазами, частотами и амплитудами. В природе звуковые сигналы имеют непрерывный спектр. Их разложение в гармонические ряды и представление в виде дискретных цифровых сигналов выполняют специальные устройства – аналого-цифровые преобразователи (АЦП). При воспроизведении происходит обратное преобразование – цифроаналоговое (ЦАП). Конструктивно АЦП и ЦАП находятся в звуковой карте компьютера. При таких преобразованиях неизбежны потери информации, связанные с методом кодирования. Метод компактен, но качество звучания не очень высокое и соответствует качеству звучания простейших электромзыкальных инструментов.
- **Метод таблично-волнового синтеза** (Wave-Table) заключается в том, что образцы звуков для множества различных музыкальных инструментов (сэмплы) хранятся в особых таблицах. Числовые коды выражают тип инструмента, высоту тона, продолжительность и интенсивность звука, динамику его изменения и другие особенности. Затем при моделировании звуковой информации эти образцы смешиваются. Качество звука, полученное в результате синтеза, приближается к качеству звучания реальных музыкальных инструментов.

При оцифровке звукового сигнала (моно) расчет количества памяти, выделяемого под звук длительностью t секунд и частотой дискретизации v , производят по формуле:

$$\text{Память (в битах)} = t * v * n,$$

где n –глубина звука.

Общее количество уровней звука K в этом случае вычисляют по формуле:

$$K = 2^n.$$

При оцифровке звукового сигнала (стерео) расчет количества памяти, выделяемого под звук длительностью t секунд и частотой дискретизации v , производят по формуле:

$$\text{Память (в битах)} = 2 * t * v * n.$$

Лекция 8

Основные подходы к определению количества информации. Понятие энтропии. Формула Шеннона, формула Хартли

Количество информации есть *количественная мера разнообразия*. Это может быть разнообразие исходов конкретной ситуации; разнообразие элементов некоторой системы; разнообразие сигнала, воспринятого в процессе конкретного сообщения¹.

Любое сообщение между источником и приемником информации имеет некоторую продолжительность во времени (длину сообщения). При этом количество информации, воспринятой приемником в результате сообщения, характеризуется в итоге вовсе не длиной сообщения, а разнообразием сигнала, порожденного в приемнике этим сообщением.

Память носителя информации имеет некоторую *физическую ёмкость*, в которой она способна *накапливать образы*, и *количество накопленной в памяти информации*, характеризуется в итоге именно *разнообразием* заполнения этой ёмкости. Для объектов неживой природы это разнообразие их истории, для живых организмов это разнообразие их опыта.

Разнообразие человеческого опыта, отражается в мироощущении: однообразие сужает кругозор, разнообразие – расширяет. Сидение в четырех стенах сужает мир до размера квартиры, наличие в квартире телевизора – расширяет мир до размеров планеты. Тяга к путешествиям, общению, новым знакомствам, новому опыту, новым ощущениям это тяга к получению новых, ранее не известных впечатлений в память, новых образов, следовательно, новой информации.

Разнообразие необходимо при передаче информации. Нельзя нарисовать белым по белому, одного состояния недостаточно. Если ячейка памяти способна находиться *только в одном* (исходном) состоянии и не способна изменять свое состояние под внешним воздействием, это значит, что она не способна воспринимать и запоминать информацию. Информационная емкость такой ячейки равна 0.

Минимальное разнообразие обеспечивается наличием *двух* состояний. Если ячейка памяти способна принимать одно из двух состояний, то она обладает *минимальной информационной ёмкостью*. Два возможных состояния условно обозначаются как «0» и «1».

• **Бит** (*bit* - сокращение от англ. *binary digit* - двоичное число) - информационная ёмкость одной ячейки памяти, способной находиться в двух различных состояниях.

Бит принят за единицу измерения *количества информации* и *информационной энтропии*. На физическом уровне бит является ячейкой

¹ Сайт www.5ballov.qip.ru

памяти, которая в каждый момент времени находится в одном из двух состояний: «0» или «1».

Если каждая точка некоторого изображения может быть только либо черной, либо белой, такое изображение называют битовым, потому что каждая точка представляет собой ячейку памяти емкостью 1 бит. Лампочка, которая может либо «гореть», либо «не гореть» также символизирует бит. Классический пример, иллюстрирующий 1 бит информации – количество информации, получаемое в результате подбрасывания монеты – “орел” или “решка”.

Количество информации, равное 1 биту, можно получить в ответе на вопрос типа «да»/ «нет».

Если изначально вариантов ответов было больше двух, количество получаемой в конкретном ответе информации будет больше, чем 1 бит.

Если вариант ответа один, то это не вопрос, а утверждение, следовательно, получения информации не требуется, *неопределенности* нет, значит, количество информации равно 0.

Информационная ёмкость ячейки памяти, способной воспринимать информацию, не может быть меньше 1 бита, но *количество* получаемой информации может быть и меньше, чем 1 бит. Это происходит тогда, когда варианты ответов «да» и «нет» *не равновероятны*. Неравновероятность в свою очередь является следствием того, что некоторая предварительная (априорная) информация по этому вопросу уже имеется, полученная, допустим, на основании предыдущего жизненного опыта. Таким образом, во всех рассуждениях предыдущего абзаца следует учитывать одну очень важную оговорку: они справедливы только для равновероятного случая.

Количество информации мы будем обозначать символом I , вероятность обозначается символом P .

Неопределенность, количество информации и энтропия

Основоположителем теории информации является *Клод Шеннон*.

Понятие информации тесно связано с понятием неопределенности. Неопределенность возникает в ситуации выбора. Задача, которая решается в ходе снятия неопределенности – уменьшение количества рассматриваемых вариантов (уменьшение разнообразия), и в итоге выбор одного соответствующего ситуации варианта из числа возможных. Снятие неопределенности дает возможность принимать обоснованные решения и действовать. Поэтому получение информации - необходимое условие для снятия неопределенности.

Ситуация *максимальной неопределенности* предполагает наличие нескольких *равновероятных* альтернатив (вариантов), т.е. ни один из вариантов не является более предпочтительным. Причем, *чем больше равновероятных вариантов* наблюдается, тем больше неопределенность, тем сложнее сделать однозначный выбор и *тем больше информации* требуется для этого получить. Для N вариантов эта ситуация описывается следующим распределением вероятностей: $\{1/N, 1/N, \dots, 1/N\}$.

Минимальная неопределенность равна 0, т.е. эта ситуация *полной определенности*, означающая что выбор сделан, и вся необходимая информация получена. Распределение вероятностей для ситуации полной определенности выглядит так: $\{1, 0, \dots, 0\}$.

- **Энтропия (H)** – мера неопределенности, выраженная в битах.

Пусть система может находиться в двух альтернативных состояниях. На рисунке 8.1 показано поведение энтропии для случая двух альтернатив, при изменении соотношения их вероятностей ($p, (1-p)$).

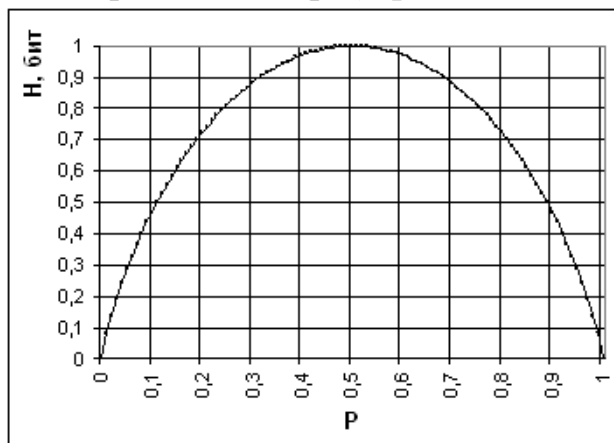


Рисунок 8.1 – Поведение энтропии для случая двух альтернатив

Максимального значения энтропия достигает в данном случае тогда, когда обе вероятности равны между собой и равны $\frac{1}{2}$, нулевое значение энтропии соответствует случаям ($p_0=0, p_1=1$) и ($p_0=1, p_1=0$).

Количество информации I и энтропия H характеризуют одну и ту же ситуацию, но с качественно противоположенных сторон. I – это количество информации, которое требуется для снятия неопределенности H. По определению Леона Бриллюэна информация есть отрицательная энтропия (негэнтропия).

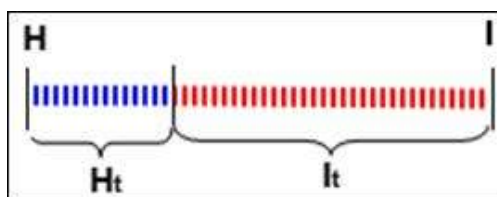


Рисунок 8.2 – Связь между энтропией и количеством информации.

Когда неопределенность снята полностью, количество полученной информации I равно изначально существовавшей неопределенности H .

При частичном снятии неопределенности, полученное количество информации и оставшаяся неснятой неопределенность составляют в сумме исходную неопределенность.

$$H_t + I_t = H.$$

По этой причине, формулы, которые будут представлены ниже для расчета энтропии H являются и формулами для расчета количества информации I , т.е. когда речь идет о полном снятии неопределенности, H в них может заменяться на I .

Формула Шеннона

В общем случае, энтропия H и количество получаемой в результате снятия неопределенности информации I зависят от исходного количества рассматриваемых вариантов N и априорных вероятностей реализации каждого из них $P: \{p_0, p_1, \dots, p_{N-1}\}$, т.е. $H=F(N, P)$. Расчет энтропии в этом случае производится по формуле Шеннона, предложенной им в 1948 году в статье "Математическая теория связи".

Формула Шеннона имеет следующий вид:

$$H = -\sum_{i=0}^{N-1} p_i \log_2(p_i) = \sum_{i=0}^{N-1} p_i \log_2\left(\frac{1}{p_i}\right) \quad (8.1)$$

Знак минус в формуле (1) не означает, что энтропия – отрицательная величина. Объясняется это тем, что $p_i \leq 1$ по определению, а логарифм числа меньшего единицы – величина отрицательная. По свойству логарифма $-\log(\alpha) = \log\left(\frac{1}{\alpha}\right)$, поэтому эту формулу можно записать и во втором варианте, без минуса перед знаком суммы.

$\log_2\left(\frac{1}{p_i}\right)$ интерпретируется как частное количество информации I_i , получаемое в случае реализации i -ого варианта. Энтропия в формуле Шеннона является средней характеристикой – математическим ожиданием распределения случайной величины $\{I_0, I_1, \dots, I_{N-1}\}$.

Напомним, что суммарная вероятность полной группы событий равна 1.

Формула Шеннона (1) совпала по форме с формулой Больцмана, полученной на 70 лет ранее для измерения термодинамической энтропии идеального газа. Эта связь между количеством информации и термодинамической энтропией послужила сначала причиной горячих дискуссий, а затем – ключом к решению ряда научных проблем. В самом общем случае энтропия понимается как мера неупорядоченности, неорганизованности материальных систем.

В соответствии со вторым законом термодинамики закрытые системы, т.е. системы лишенные возможности вещественно-энергетически-информационного обмена с внешней средой, стремятся, и с течением времени неизбежно приходят к естественному устойчивому равновесному внутреннему состоянию, что соответствует состоянию с максимальной энтропией. Закрытая система стремится к однородности своих элементов и к равномерности распределения энергии связей между ними. Т.е. в отсутствие информационного процесса материя самопроизвольно забывает накопленную информацию.

Приведем пример расчета энтропии по формуле Шеннона. Пусть в некотором учреждении состав работников распределяется так: $\frac{3}{4}$ – женщины, $\frac{1}{4}$ – мужчины. Тогда неопределенность, например, относительно того, кого вы встретите первым, зайдя в учреждение, будет рассчитана рядом действий, показанных в таблице 1.

Таблица 1.

	p_i	$1/p_i$	$I_i = \log_2(1/p_i)$, бит	$p_i * \log_2(1/p_i)$, бит
<i>Ж</i>	3/4	4/3	$\log_2(4/3) \approx 0,42$	$3/4 * 0,42 \approx 0,31$
<i>М</i>	1/4	4/1	$\log_2(4) = 2$	$1/4 * 2 = 0,5$
Σ	1			$H = 0,81$ бит

Если же априори известно, что мужчин и женщин в учреждении поровну (два равновероятных варианта), то при расчете по той же формуле мы должны получить неопределенность в 1 бит. Проверка этого предположения проведена в таблице 2.

Таблица 2.

	p_i	$1/p_i$	$I_i = \log_2(1/p_i)$, бит	$p_i * \log_2(1/p_i)$, бит
<i>Ж</i>	1/2	2	$\log_2(2) = 1$	$1/2 * 1 = 1/2$
<i>М</i>	1/2	2	$\log_2(2) = 1$	$1/2 * 1 = 1/2$
Σ	1			$H = 1$ бит

В частном случае, когда все варианты равновероятны, остается зависимость только от количества рассматриваемых вариантов, т.е. $H = F(N)$. В этом случае формула Шеннона значительно упрощается и совпадает с формулой Хартли, которая впервые была предложена американским инженером Ральфом Хартли в 1928 году, т.е. не 20 лет раньше.

Формула Хартли

Мы уже упоминали, что формула Хартли – частный случай формулы Шеннона для равновероятных альтернатив.

Подставив в формулу (1) вместо p_i его (в равновероятном случае не зависящее от i) значение $p_i = \frac{1}{N}$, получим:

$$H = \sum_{i=0}^{N-1} \frac{1}{N} \log_2 \left(\frac{N}{1} \right) = \frac{1}{N} * N * \log_2(N) = \log_2(N)$$

Таким образом, формула Хартли выглядит очень просто:

$$\boxed{H = \log_2(N)} \quad (8.2)$$

Из нее явно следует, что чем больше количество альтернатив (N), тем больше неопределенность (H). Эти величины связаны в формуле (8.2) не линейно, а через двоичный логарифм. Логарифмирование по основанию 2 и приводит количество вариантов к единицам измерения информации – битам.

Заметьте, что энтропия будет являться целым числом лишь в том случае, если N является степенью числа 2, т.е. если N принадлежит ряду: $\{1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048 \dots\}$

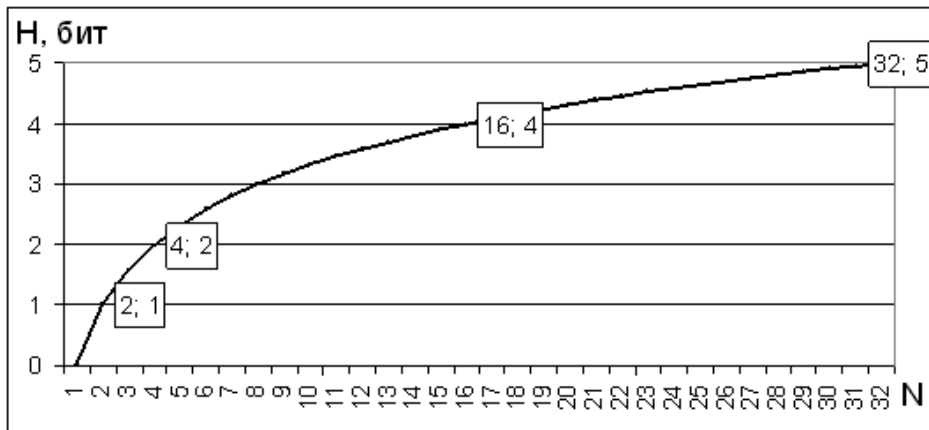


Рисунок 8.3 – Зависимость энтропии от количества равновероятных вариантов выбора (равнозначных альтернатив)

Понятие логарифма

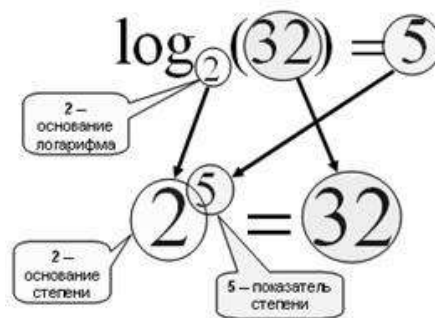


Рисунок 8.4 – Нахождение логарифма b по основанию a

Логарифм b по основанию a - это *степень*, в которую нужно возвести a , чтобы получить b .

Логарифм по основанию 2 называется *двоичным*:

$$\log_2(8)=3 \Rightarrow 2^3=8$$

$$\log_2(10)=3,32 \Rightarrow 2^{3,32}=10$$

Логарифм по основанию 10 —называется *десятичным*:

$$\log_{10}(100)=2 \Rightarrow 10^2=100$$

Основные свойства логарифма

1. $\log(1)=0$, т.к. любое число в нулевой степени дает 1;
2. $\log(a^b)=b \cdot \log(a)$;
3. $\log(a \cdot b)=\log(a)+\log(b)$;
4. $\log(a/b)=\log(a)-\log(b)$;
5. $\log(1/b)=0-\log(b)=-\log(b)$.

Для решения обратных задач, когда известна неопределенность (H) или полученное в результате ее снятия количество информации (I) и нужно определить какое количество равновероятных альтернатив соответствует возникновению этой неопределенности, используют обратную формулу Хартли, которая выглядит еще проще:

$$N = 2^H \quad (8.3)$$

Например, если известно, что в результате определения того, что интересующий нас Коля Иванов живет на втором этаже, было получено 3 бита информации, то количество этажей в доме можно определить по формуле (3), как $N=2^3=8$ *этажей*.

Если же вопрос стоит так: “в доме 8 этажей, какое количество информации мы получили, узнав, что интересующий нас Коля Иванов живет на втором этаже?”, нужно воспользоваться формулой (8.2): $I=\log_2(8)=3$ *бита*.

Количество информации, получаемой в процессе сообщения

До сих пор мы приводили формулы для расчета энтропии (неопределенности) H , указывая, что H в них можно заменять на I , потому что количество информации, получаемое *при полном снятии неопределенности* некоторой ситуации, количественно равно начальной энтропии этой ситуации.

Но *неопределенность может быть снята только частично*, поэтому количество информации I , получаемой из некоторого сообщения, вычисляется как *уменьшение энтропии, произошедшее в результате получения данного сообщения*.

$$I = H_{\text{до}} - H_{\text{после}} \quad (8.4)$$

Для *равновероятного случая*, используя для расчета энтропии формулу Хартли, получим:

$$I = \log_2(N_{\text{до}}) - \log_2(N_{\text{после}}) = \log_2\left(\frac{N_{\text{до}}}{N_{\text{после}}}\right) \quad (8.5)$$

Второе равенство выводится на основании свойств логарифма. Таким образом, в равновероятном случае I зависит от того, *во сколько раз* изменилось количество рассматриваемых вариантов выбора (рассматриваемое разнообразие).

Исходя из (5) можно вывести следующее:

$$I = \log_2\left(\frac{N_{\text{до}}}{1}\right) = \log_2(N_{\text{до}}) = H_{\text{до}}$$

Если $N_{\text{после}}=1$, то $I = H_{\text{до}}$ - полное снятие неопределенности, количество полученной в сообщении информации равно неопределенности, которая существовала до получения сообщения.

Если $N_{\text{после}}=N_{\text{до}}$, то $I = \log_2(1) = 0$ - неопределенности не изменилась, следовательно, информации получено не было.

Если $N_{\text{после}} < N_{\text{до}}$, то $\frac{N_{\text{до}}}{N_{\text{после}}} > 1 \Rightarrow I > 0$, если $N_{\text{после}} > N_{\text{до}}$, $\frac{N_{\text{до}}}{N_{\text{после}}} < 1 \Rightarrow I < 0$. Т.е. количество полученной информации будет положительной величиной, если в результате получения сообщения количество рассматриваемых альтернатив уменьшилось, и отрицательной, если увеличилось.

Если количество рассматриваемых альтернатив в результате получения

$$\frac{N_{до}}{N_{после}} = 2$$

сообщения уменьшилось вдвое, т.е. $N_{после}$, то $I = \log_2(2) = 1$ бит. Другими словами, получение 1 бита информации исключает из рассмотрения половину равнозначных вариантов.

Рассмотрим в качестве примера опыт с колодой из 36 карт.

4	♣									
	♠									
	♦									
	♥									
		6	7	8	9	10	В	Д	К	Т

Рисунок 8.5 - Иллюстрация к опыту с колодой из 36-ти карт

Пусть некто вынимает одну карту из колоды. Нас интересует, какую именно из 36 карт он вынул. Изначальная неопределенность, рассчитываемая по формуле (8.2), составляет $H = \log_2(36) \approx 5,17$ бит. Вытянувший карту сообщает нам часть информации. Используя формулу (8.5), определим, какое количество информации мы получаем из этих сообщений:

Вариант А. "Это карта красной масти".

$I = \log_2(36/18) = \log_2(2) = 1$ бит (красных карт в колоде половина, неопределенность уменьшилась в 2 раза).

Вариант В. "Это карта пиковой масти".

$I = \log_2(36/9) = \log_2(4) = 2$ бита (пиковые карты составляют четверть колоды, неопределенность уменьшилась в 4 раза).

Вариант С. "Это одна из старших карт: валет, дама, король или туз".

$I = \log_2(36) - \log_2(16) = 5,17 - 4 = 1,17$ бита (неопределенность уменьшилась больше чем в два раза, поэтому полученное количество информации больше одного бита).

Вариант Д. "Это одна карта из колоды".

$I = \log_2(36/36) = \log_2(1) = 0$ бит (неопределенность не уменьшилась - сообщение не информативно).

Вариант Е. "Это дама пик". $I = \log_2(36/1) = \log_2(36) = 5,17$ бит (неопределенность полностью снята).

Лекция 9

Элементы алгебры логики.

Основные логические функции.

Законы преобразования логических выражений

Для описания логики функционирования аппаратных и программных средств компьютера используется алгебра логики или булева алгебра.

Дж. Буль – английский математик 19 века. Булева алгебра оперирует с логическими переменными, которые могут принимать только 2 значения: истина и ложь, обозначаемые соответственно 1 и 0.

Основные понятия

- Совокупность значений логических переменных x_1, x_2, \dots, x_n называется **набором переменных**. Набор логических переменных удобно изображать в виде n -разрядного двоичного числа, каждый разряд которого равен значению одной из переменных. Количество наборов логических переменных в n двоичных разрядах равно 2^n .

- **Логической функцией** от набора логических переменных $f(x_1, x_2, \dots, x_n)$ называется функция, которая может также принимать только 2 значения: *истина* или *ложь*.

Любая логическая функция может быть задана с помощью таблицы истинности, в левой части которой записываются возможные наборы переменных, а в правой – соответствующие им значения функции.

В случае большого числа переменных, табличный способ становится громоздким. Поэтому, логические функции выражают через элементарные логические функции, которые легко задаются таблично. Как правило, это функции от одной или двух переменных.

Совокупность логических функций, с помощью которых можно выразить логическую функцию любой сложности, называются функционально полными системами логических функций.

Наиболее часто используемая система логических функций: инверсия ($\bar{}$, отрицание, NOT), конъюнкция (\wedge , логическое умножение, AND, &), дизъюнкция (\vee , логическое сложение, OR).

Инверсия	
x	\bar{x}
0	1
1	0

Конъюнкция		
x1	x2	$x1 \wedge x2$
0	0	0
0	1	0
1	0	0
1	1	1

Дизъюнкция		
x1	x2	$x1 \vee x2$
0	0	0
0	1	1
1	0	1
1	1	1

Логические переменные, объединенные знаками логических операций, составляют *логические выражения*. При вычислении логических выражений определено следующее старшинство выполнения логических операций: инверсия, конъюнкция, дизъюнкция. Для изменения порядка используют скобки.

$$f(x_1, x_2, x_3) = (x_1 \wedge x_2 \vee x_2 \vee x_3) \wedge x_1 \vee x_3;$$

$$f(0, 1, 1) = (0 \wedge 1 \vee 1 \vee 1) \wedge 0 \vee 1 = 0;$$

$$f(1, 0, 1) = (1 \wedge 0 \vee 0 \wedge 1) \wedge 1 \vee 1 = 1.$$

Основные законы преобразования логических выражений

В алгебре логики выполняются следующие основные законы, позволяющие производить тождественные преобразования логических выражений

1. Коммуникативный закон (переместительный):

$$x_1 \vee x_2 = x_2 \vee x_1$$

$$x_1 \wedge x_2 = x_2 \wedge x_1$$

2. Ассоциативный закон (сочетательный):

$$x_1 \vee (x_2 \vee x_3) = (x_1 \vee x_2) \vee x_3$$

$$x_1 \wedge (x_2 \wedge x_3) = (x_1 \wedge x_2) \wedge x_3$$

3. Дистрибутивный закон (распределительный):

$$x_1 \wedge (x_2 \vee x_3) = (x_1 \wedge x_2) \vee (x_1 \wedge x_3)$$

$$x_1 \vee (x_2 \wedge x_3) = (x_1 \vee x_2) \wedge (x_1 \vee x_3)$$

4. Правила де Моргана:

$$\overline{x_1 \vee x_2} = \overline{x_1} \wedge \overline{x_2}$$

$$\overline{x_1 \wedge x_2} = \overline{x_1} \vee \overline{x_2}$$

5. Правила операций с константами:

$$0 = \overline{1}; \quad \overline{1} = 0; \quad x \wedge 1 = x; \quad x \wedge 0 = 0; \quad x \vee 0 = x; \quad x \vee 1 = 1.$$

6. Правила операций переменной с ее инверсией:

$$x \vee \overline{x} = 1;$$

$$x \wedge \overline{x} = 0;$$

7. Закон поглощения:

$$x_1 \vee (x_1 \wedge x_2) = x_1;$$

$$x_1 \wedge (x_1 \vee x_2) = x_1.$$

8. Закон идемпотентности:

$$x \vee x = x;$$

$$x \wedge x = x;$$

9. Закон двойного отрицания:

$$\overline{\overline{x}} = x.$$

Схемные реализации операций, выполняемых элементарными логическими функциями, называются логическими элементами. С их помощью реализуются функции управления процессом обработки информации.

Лекция 10

Методы работы с процедурной информацией. Постановка задачи. Моделирование информации

Информацию можно рассматривать не только как совокупность фактических данных, но и как динамическую величину, то есть информацию действия. В таком аспекте информация необходима для решения некоторой задачи.

• **Задача** – это проблема, подлежащая решению.

Различают технологические и функциональные задачи.

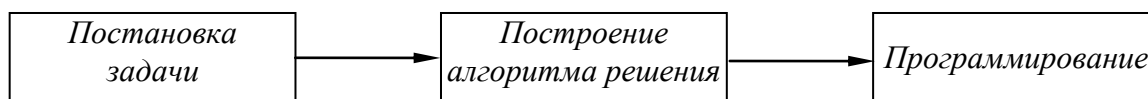
Технологические задачи – ставятся и решаются при организации технологического процесса обработки информации на компьютере. Технологические задачи являются основой для сервисных систем программного обеспечения в виде операционных систем, систем программирования, систем управления базами данных, табличных процессоров, текстовых процессоров и т.д., т.е. при разработке системного программного обеспечения.

Функциональные задачи – ставятся при реализации функций управления в рамках информационных систем предметных областей.

Например, задачи управления деятельностью предприятия или фирмы, управления поставками и перевозками продукции и т.д.

Функциональные задачи в совокупности образуют предметную область и полностью определяют ее специфику. Функциональные задачи решаются при разработке прикладного программного обеспечения.

Процесс решения задач на компьютере можно представить как последовательность трех этапов:



Постановка задачи

• **Постановка задачи** – это точная формулировка решения задачи на компьютере с описанием входной и выходной информации рассматриваемой предметной области.

Постановка задачи связана с конкретизацией основных параметров ее реализации, определением источников и структурой входной и выходной информации, востребованной пользователем.

Для облегчения решения сложной задачи ее необходимо разбивать на подзадачи.

К основным характеристикам задач относятся:

- цель решения задачи, ее место и связи с другими задачами;
- условия решения задачи с использованием средств вычислительной техники;
- требования к периодичности решения задачи;
- ограничения по срокам и точности выходной информации;
- состав и форма представления выходной информации;
- источники входной информации для решения задачи;
- пользователи задачи (кто ее решает и пользуется результатами).

Этап постановки задачи изначально является главным этапом проектирования любой программы, т.к. от того, как будет определена цель и выбрана модель описания предметной области, зависит правильное решение исходной задачи.

Моделирование информации

Для успешной организации информационных процессов с помощью технических и программных средств нужно представить информацию о реальном объекте в таком виде, чтобы ее можно было ввести в информационную систему, обработать, при необходимости сохранить и выдать по первому требованию пользователя. Для этого информацию необходимо смоделировать.

Введем несколько определений [1].

- **Модель** - это формальное (как правило, приближенное) описание изучаемого объекта или явления, отражающее интересующие нас аспекты.

- **Математическая модель** (формулы + методы + правила) является основой построения информационной модели.

- **Информационная модель** задачи подразумевает определение вида, структуры, последовательности обработки и подачи на экран информации, необходимой для работы программного продукта.

- **Входная информация** определяется как данные, поступающие на вход задачи и используемые для ее решения. Входной информацией служат рукописные документы, файлы базы данных, нормативно-справочная информация, входные сигналы от датчиков устройств и т.д.

- **Выходная информация** по задаче может быть представлена в виде документов, выданных на принтер, файла базы данных, графика на экране дисплея, управляющего сигнала и т.д.

- **Предметная область** (ПрО) - совокупность связанных между собой функций, задач управления, с помощью которых достигается выполнение поставленных целей. Любая предметная область имеет границы.

- **Модель данных** (МД) - это используемая знаковая система (способ абстрагирования предметной области) для обозначения структур данных и операций их обработки.

- **Концептуальная схема ПрО** - описание предметной области в терминах выбранной модели данных.

Распространённой моделью данных является устная или письменная речь. Концептуальная схема ПрО в этом случае выражается в виде совокупности текстов и/или фонограмм. Однако тексты и фонограммы плохо приспособлены для моделирования информации, имеющей сложную структуру. В этом случае используют модели данных, располагающие удобными средствами описания информационной структуры. Существуют различные модели данных для описания информационных структур (иерархические, сетевые, реляционные и т.д.).

Для описания ПрО используют три основных конструктивных элемента - сущность, атрибут и связь

- **Сущность** - это обобщённое понятие для обозначения множества однородных объектов ПрО, информацию о которых необходимо собирать и

хранить в информационной системе. Сущность определяется своим уникальным именем и перечнем атрибутов, характеризующих свойства сущности.

• **Атрибут** - это поименованная характеристика сущности, которая принимает значения из некоторого множества допустимых значений. Атрибуты моделируют свойства сущности.

Чтобы задать атрибут в модели, необходимо присвоить ему наименование, определить множество его допустимых значений и указать, для чего он используется. Имя атрибута связывают с его смысловым описанием. В этой связи совокупность имен атрибутов описывает свойства сущности.

• **Связь** - это обобщённое понятие, предназначенное для обозначения выделенного в ПрО отношения между двумя или более сущностями. Как и сущности, каждая категория пользователей выделяет связи в соответствии со своей концепцией ПрО.

Пример. Выделить информационные объекты учебного процесса.

Описывая предметную область «**Учебный процесс**», можно выделить несколько сущностей: **студент, преподаватель, предмет**.

Сущность **Студент** можно охарактеризовать следующими атрибутами:

ФИО, ДАТА_РОЖДЕНИЯ, АДРЕС, ДАТА_ПОСТУПЛЕНИЯ, НОМЕР_ЗАЧЕТНОЙ_КНИЖКИ, НОМЕР_ГРУППЫ.

Сущность **Преподаватель** характеризуется следующими атрибутами:

ФИО, АДРЕС, ДОЛЖНОСТЬ, КАФЕДРА, УЧЕНОЕ_ЗВАНИЕ, ТЕЛЕФОН_РАБОЧИЙ, ТЕЛЕФОН_ДОМАШНИЙ.

Сущность **Предмет** характеризуется следующими атрибутами:

НАЗВАНИЕ, КАФЕДРА, ЛЕКЦИИ, ПРАКТИЧЕСКИЕ_ЗАНЯТИЯ, ЛАБОРАТОРНЫЕ_РАБОТЫ, КУРСОВЫЕ_РАБОТЫ, ОТЧЕТНОСТЬ, СЕМЕСТР.

Между сущностями **Студент**, **Преподаватель** и **Предмет** существует связь **Экзамен**. Связь, как и сущность, может иметь атрибуты. Связь экзамен характеризуется, например, атрибутами **ОЦЕНКА** и **ДАТА_ПРОВЕДЕНИЯ**.

Предметная область определена, если известны существующие в ней объекты, их свойства и отношения (связи).

При описании той или иной ПрО желательно, чтобы соблюдались следующие требования:

- полнота охвата объектов (сущностей) рассматриваемой области;
- однозначность атрибутов;
- возможность включения новых объектов (сущностей).

Лекция 11

Построение алгоритма решения задачи

После разработки модели и определения применяемых математических методов строится алгоритм, т.е. строгая последовательность действий по решению поставленной задачи.

• **Алгоритм** – система точно сформулированных правил, однозначно определяющая процесс преобразования допустимых исходных данных (входной информации) в желаемый результат (выходную информацию) за конечное число шагов.

Алгоритм включает в себя также порядок диалога с пользователем, порядок использования устройств машины и т.д. При этом алгоритм не должен быть привязан к конкретному языку программирования, т.е. по нему можно написать программу на любом языке программирования.

Алгоритм должен обладать следующими свойствами:

• **дискретностью** – разбиением процесса обработки информации на более простые этапы (шаги), выполнение которых компьютером или человеком не вызывает затруднений;

• **определенностью** (детерминированностью) – однозначностью получаемого результата при одних и тех же исходных данных;

• **результативностью** – обязательным получением желаемого результата за конечное число шагов при допустимых исходных данных;

• **массовостью** – применимостью алгоритма для решения определенного класса задач.



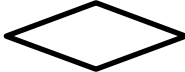



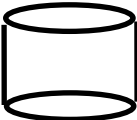
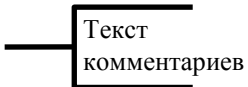
В алгоритме отражаются логика и способ формирования результатов решения с указанием необходимых расчетных формул, логических условий, соотношений для контроля достоверности выходных результатов. В алгоритме обязательно должны быть предусмотрены все ситуации, которые могут возникнуть в процессе решения задачи. В зависимости от степени детализации, поставленных целей, методов и технических средств решения задачи используются различные способы представления алгоритмов. На практике наиболее распространены следующие формы представления алгоритмов: содержательная (текстуальная) форма; графическая форма (блок-схема); программная (на языках программирования компьютера).

Остановимся подробнее на графическом представлении алгоритма.

Графическая форма представления алгоритмов

Является более компактной и наглядной по сравнению с текстовой формой. Алгоритм изображается в виде последовательности связанных между собой функциональных блоков, каждый из которых соответствует выполнению одного или нескольких действий. Такое графическое представление называется блок-схемой алгоритма.

Основные условные обозначения функциональных блоков, принятые в схемах алгоритмов:

<i>Графическое обозначение</i>	<i>Наименование</i>	<i>Пояснения</i>
	Пуск-останов	Начало, конец, прерывание процесса обработки данных.
	Процесс, действие	Операция, в результате которой изменяется значение данных.
	Условие	Разветвление алгоритма в зависимости от некоторых условий.
	Модификация	Начало циклической операции.
	Программа, подпрограмма	Часть алгоритма, требующая дополнительной детализации на последующих шагах.
	Ввод-вывод данных	Ввод-вывод без указаний конкретного носителя.
	Документ	Ввод-вывод данных, носителем которых является бумага.
	Дисплей	Ввод-вывод данных на дисплей.
	Магнитный диск	Ввод-вывод данных на магнитный диск.
	Комментарий	Связь между элементом схемы и пояснением к нему.
	Соединители	Связь между прерванными линиями на одной странице, связь между прерванными частями схем на разных стр.

Пример различных форм алгоритма

Задача: Даны три числа $A1$, $A2$, $A3$. Найти максимальное число A_{max} .

Содержательная форма представления алгоритма

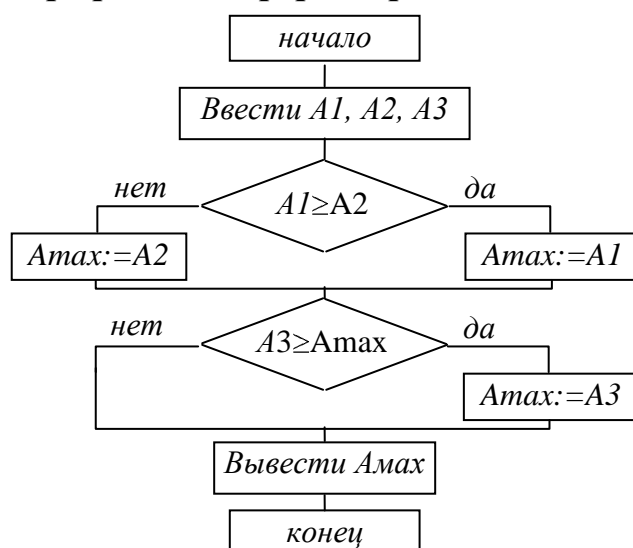
Шаг 1. Сравнить значения чисел $A1$ и $A2$; если $|A1| \geq |A2|$, то числу A_{max} присвоить значение $|A1|$, иначе числу A_{max} присвоить значение $|A2|$.

Шаг 2. Сравнить значения чисел $A3$ и A_{max} . Если $|A3| \geq |A_{max}|$, то числу A_{max} присвоить значение $|A3|$.

Шаг 3. Конец алгоритма.

Приведенное описание алгоритма является достаточно строгим и позволяет однозначно решить поставленную задачу. Однако, для достаточно сложных алгоритмов описание становится слишком громоздким и ненаглядным. Поэтому содержательную форму используют на начальных стадиях разработки алгоритмов, когда определяются основные этапы решения поставленной задачи.

Графическая форма представления алгоритма



Программная форма представления алгоритма

```
Program max_3;  
Var A1,A2,A3,Amax:integer;  
Begin  
  Writeln('Введите три числа');  
  Readln(A1,A2,A3);  
  if A1>=A2 then Amax:=A1  
    else Amax:=A2;  
  if A3>=Amax then Amax:=A3;  
  Writeln('Максимальное число ',Amax);  
End.
```


Лекция 12

Стадии разработки программного продукта

Программирование решения задачи

Алгоритм и его программная реализация тесно взаимосвязаны.

• **Программа** – упорядоченная последовательность инструкций компьютера (команд) для решения задачи.

• **Программное обеспечение** – совокупность программ обработки данных и необходимых для их эксплуатации документов.

• **Программирование** – теоретическая и практическая деятельность, связанная с созданием программ.

Программирование является собирательным понятием и может рассматриваться и как наука, и как искусство; на этом основан научно-практический подход к разработке программ.

Программа – результат интеллектуального труда, для которого характерно творчество, поэтому в любой программе присутствует индивидуальность ее разработчика, программа отражает определенную степень искусства программиста. Вместе с тем программирование предполагает и рутинные работы, которые могут и должны иметь строгий регламент выполнения и соответствовать стандартам.

Программирование базируется на комплексе научных дисциплин, направленных на исследование, разработку и применение методов и средств разработки программ. При разработке программ используются ресурсоемкие и наукоемкие технологии, высококвалифицированный интеллектуальный труд. По некоторым данным в середине 90-х годов в мире было занято программированием $\approx 2\%$ трудоспособного населения. Совокупный оборот в сфере создания программных средств достигает нескольких сот миллиардов долларов в год. В связи с этим весьма актуальным становится вопрос о разработке и применении эффективных технологий программирования.

Стадии разработки программного продукта

Этот этап представляет собой выполнение четырех взаимосвязанных задач:

- ◆ реализация алгоритма на языке программирования;
- ◆ отладка программы;
- ◆ тестирование программы;
- ◆ модификация и сопровождение программы.

• **Написание программы** - запись разработанного алгоритма на каком-либо языке программирования.

• **Отладка**. На этом этапе с помощью специальной программы (транслятора) в программе устраняются **синтаксические** ошибки (связанные с неправильной записью или употреблением языковых конструкций).

• **Тестирование.** На этом этапе устраняются *семантические* (смысловые) ошибки. Необходимо проводить тестирование в два этапа: на заведомо правильных, а затем на заведомо неправильных данных. В любом случае программа должна выдавать соответствующие результаты. Как правило, в программе должны быть отражены все особые случаи. Для сложных алгоритмов такая информация, как правило, неизвестна.

• **Модификация.** Программа постоянно модифицируется как на этапе разработки, так и в период опытной эксплуатации. Программы, рассчитанные на долгий срок применения, требуют периодического усовершенствования. Это может быть связано с дополнительными требованиями к ее возможностям, сменой оборудования и т.п. Поэтому необходимо, чтобы программа предоставляла возможность своей модификации без полной повторной реализации.

• **Сопровождение программного продукта** - исправление обнаруженных ошибок, а также поддержка работоспособности программного продукта при переходе на новые версии и внесении изменений.

Схема взаимодействия специалистов, связанных с созданием и эксплуатацией программ

Основная категория специалистов, занятых разработкой программ – это программисты. Программисты неоднородны по уровню квалификации, а также по характеру своей деятельности. Наиболее часто программисты делятся на системных и прикладных.

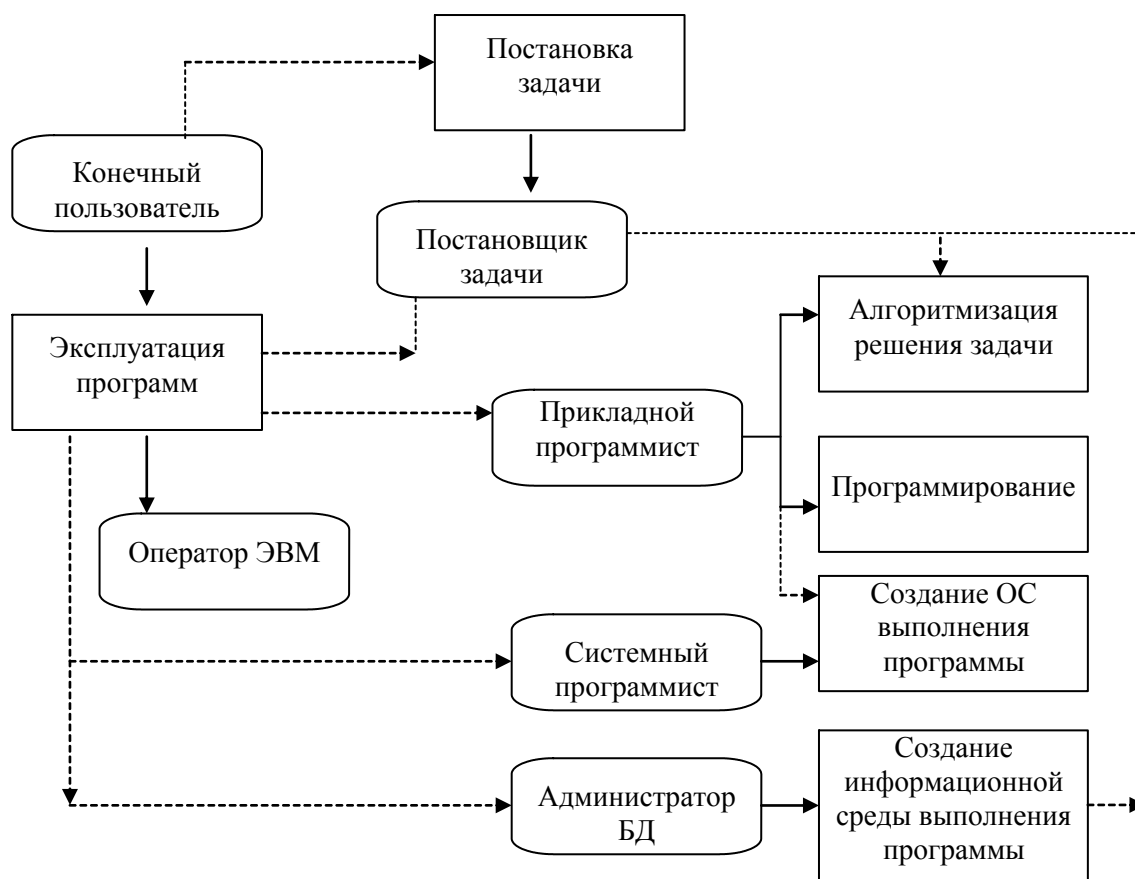
Системный программист занимается разработкой, эксплуатацией и сопровождением системного программного обеспечения, поддерживающего работоспособность компьютера и создающего сферу для выполнения программ, обеспечивающих реализацию функциональных задач.

Прикладной программист осуществляет разработку и отладку программ для решения функциональных задач.

В процессе создания программ на начальной стадии работ участвуют и специалисты – постановщики задач. При создании больших по масштабам и функциям обработки программ (например, операционных систем) – нужен программист-аналитик для анализа и проектирования комплекса взаимосвязанных программ.

Большинство информационных систем основано на работе с базами данных (БД). Если база данных является сложной и используется многими программистами, то возникает проблема организационной поддержки БД, которая выполняется администратором БД.

Основным потребителем программ служит конечный пользователь, который относится к категории пользователей-непрограммистов. Он должен иметь элементарные знания и навыки работы с компьютером.



Характеристики программного продукта

Все программы по характеру использования и категориям пользователей можно разделить на 2 класса: утилитарные программы и программные продукты (изделия).

- **Утилитарные программы (утилиты)** предназначены для использования самими разработчиками. Чаще всего это программные решения функциональных задач, не предназначенных для широкого распространения.

- **Программный продукт** – комплекс взаимосвязанных программ для решения определенной проблемы (задачи) массового спроса, подготовленный к реализации как любой вид промышленной продукции.

Программный продукт должен быть соответствующим образом подготовлен к эксплуатации, иметь необходимую техническую документацию, представлять сервис и гарантию надежной работы программы, иметь товарный знак изготовителя, а также желательно наличие кода государственной регистрации.

Программные продукты предназначены для широкого распространения и продажи.

Процесс создания программных продуктов достаточно долгий, он связан с изменениями технической и программной среды разработки и эксплуатации программ, с появлением и развитием самостоятельной отрасли – информационного бизнеса, для которого характерны разделение труда фирм-

разработчиков программ, их дальнейшая специализация, формирование рынка программных средств и информационных услуг.

Программный продукт разрабатывается на основе промышленной технологии выполнения проектных работ с применением современных инструментальных средств программирования. Специфика заключается в уникальности процесса разработки алгоритмов и программ, зависящего от характера обработки информации и используемых инструментальных средств.

Как правило, программные продукты требуют сопровождения, которое осуществляется специализированными фирмами-распространителями программ (дистрибьюторами), реже – фирмами-разработчиками. Сопровождение программ массового применения сопряжено с большими трудозатратами – исправление ошибок, создание новых версий программ и т.п.

Основные характеристики программ

1. Алгоритмическая сложность.
2. Состав и глубина проработки реализованных функций.
3. Полнота и системность функций.
4. Объем файлов программ.
5. Требования к ОС и техническим средствам со стороны программы.
6. Объем дисковой памяти.
7. Размер операционной памяти.
8. Тип процессора.
9. Версия ОС.
10. Наличие вычислительной сети и др.

Показатели качества программного продукта (ПП)

1. Мобильность – независимость ПП от технических средств обработки информации, ОС, сетевой технологии. Мобильный ПП пригоден для массового использования без каких-либо изменений.

2. Надежность – бесперебойность и устойчивость в работе, возможность диагностики возникающих ошибок.

3. Эффективность – минимально возможный расход вычислительных ресурсов и максимально возможное быстроедействие.

4. Модифицируемость - простота внесения изменений.

5. Коммуникативность – свойство интеграции с другими программами, обеспечения обмена данными в общих форматах представления.

6. Учет человеческого фактора – обеспечение дружелюбного интерфейса, наличие контекстно-зависимой подсказки или обучающей системы, хорошей документации.

Лекция 13

Программное обеспечение компьютера

Состав и назначение программного обеспечения

Процесс взаимодействия человека с компьютером организован устройством управления микропроцессора в соответствии с той программой, которую пользователь разработал и ввел в память компьютера. На начальном этапе развития вычислительной техники пользователь представлял свои программы на машинном языке в виде двоичных кодов, а устройство управления в зависимости от их содержания подключало нужные электронные цепи и схемы.

По мере усложнения задач и повышения требований к параметрам ЭВМ появилась потребность в более гибком органе управления, нежели существующее электронное устройство управления. Появились системные управляющие программы, которые, благодаря своей гибкости, взяли на себя большую часть функций устройства управления по организации процесса обработки информации на компьютере.

По мере развития машинные коды сменили языки программирования более высокого уровня, основа которых составляет простейшие слова английского языка. Появились такие языки программирования как BASIC, FORTRAN. Они позволяли программистам общаться с ЭВМ на более высоком уровне. Следующим этапом в развитии языков взаимодействия человека с компьютером стали алгоритмические языки. К ним относятся Паскаль, Си, Модула-2. Логические языки типа Пролог позволяют обрабатывать логическую и символьную информацию. Они явились прототипом для программ обработки и распознавания текстов.

Создание компьютера является только первым шагом на пути компьютеризации деятельности человека. Вся тяжесть поставленной задачи ложится на плечи программистов, основная задача которых - разработать комплекс взаимосвязанных и разнообразных по своим функциям программ, т. е. так называемое программное обеспечение.

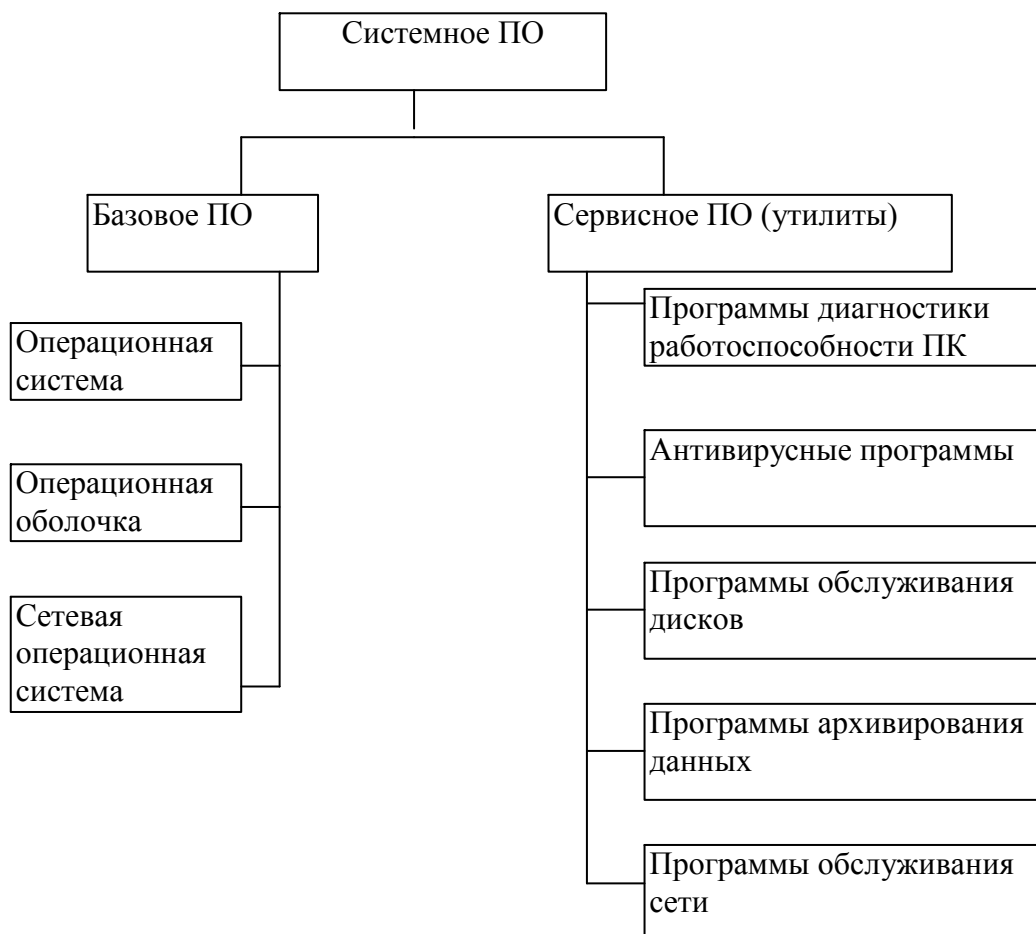
• **Программное обеспечение** - это совокупность программ, позволяющая организовать решение задач пользователя на компьютере.

Программное обеспечение является составной частью компьютера, и некоторая его часть поставляется вместе с аппаратурой. По назначению программное обеспечение можно разделить на классы:

- Системное ПО;
- Языки и системы программирования;
- Прикладное программное обеспечение;
- Уникальное ПО.

Системное программное обеспечение

• **Системное программное обеспечение** – совокупность программ и программных комплексов для обеспечения работы компьютера и сетей ЭВМ.



СПО организует процесс обработки информации в компьютере. Главную часть системного ПО составляет операционная система, основное назначение которой - управление процессом обработки информации в ЭВМ и организация связи пользователя с ЭВМ. В состав системного ПО входят также средства контроля и диагностики неисправностей в процессе работы ЭВМ, во время профилактики, при выходе ее из строя. Программы диагностики и контроля обеспечивают автоматический поиск ошибки с указанием неисправного блока. К системному ПО относятся также утилиты для выполнения вспомогательных технологических процессов (копирование, архивирование, восстановление файлов программ и БД, антивирусные программы и т.д.)

Системное ПО ориентировано, в основном, на профессионального пользователя: системного программиста, администратора сети, прикладного программиста, оператора. Однако, знание базовой технологии работы с этим классом ПП требуется и конечным пользователям. К ПП данного класса применяются высокие требования по надежности и технологичности работы, удобству и эффективности использования.

Языки и системы программирования

Позволяют разрабатывать программы на удобном символическом языке, а не в машинных кодах. Символические языки подразделяются на машинно-ориентированные языки, в которых отражена структура данного компьютера, и на алгоритмические языки, программирование на которых ведется безотносительно к конкретному типу компьютера.

В систему программирования входят трансляторы - комплекс программ, обеспечивающий автоматический перевод с символического языка на машинный язык и наоборот.

Прикладное программное обеспечение

Основное назначение прикладных программ - решать задачи пользователя. ППО состоит из библиотек стандартных программ и пакетов прикладных программ (ППП).

• **Библиотека стандартных программ** состоит из часто используемых программ вычисления функций, решения стандартных задач, выполнения распространенных операций обработки данных.

Составление подобных программ достаточно трудоемкое дело, поэтому они разработаны заранее и хранятся на жестком диске. При использовании в программах задается имя функции и значение аргумента.

Уникальное ПО

Комплекс программ, предназначенных для решения специализированных задач. В процессе разработки уникального ПО может быть создана своя уникальная операционная система ОС - основное назначение которой - обеспечить решение конкретной задачи и удовлетворить требованиям технического задания на данный пакет. Уникальные программы имеют, как правило, мощное математическое обеспечение, т.е. математические методы, алгоритмы, обеспечивающие решение поставленной задачи.

Операционные системы

Операционная система (ОС) - это комплекс программ, предназначенный для управления процессом обработки информации.

ОС – совокупность программных средств, обеспечивающая управление аппаратной частью компьютера и прикладными программами, а также их взаимодействие между собой и пользователем.

Управляющая программа всегда являлась основой работы ПК. Вначале это была непосредственно та программа, которую разрабатывал программист для решения своей задачи. Она последовательно подключала те технические устройства, которые необходимо использовать в данной задаче. При этом во время работы каждого конкретного устройства все остальные устройства простаивали, а особенно арифметико-логическое устройство, т.к. при решении задачи часто приходилось обращаться к внешним устройствам, скорость работы которых гораздо ниже.

Такой режим работы не мог удовлетворить пользователей, для которых требовались большие ресурсы компьютера. В процессе развития структуры ПК и расширение ее возможностей стали появляться специальные комплексы управляющих программ, которые могли работать независимо от программ пользователей и хранились постоянно во внутренней памяти ПК. Организация работы всех устройств компьютера стало возможным на основе функционирования операционной системы.

Основное назначение ОС состоит в обеспечении максимальной производительности компьютера.

Основные принципы построения ОС

1. Организация параллельной работы процессора, устройства ввода-вывода, внешней памяти (параллельная работа процессора и операционной памяти с использованием внешних устройств).
2. Распределить ресурсы компьютера так, чтобы были обеспечены постоянная загрузка всех устройств и их работа в той последовательности, которая необходима в решаемой задаче.

Однако созданные ОС настолько сложны, что работать в этой среде могут только высококвалифицированные программисты. Появилась новая задача вычислительной техники - создание более простого и удобного средства обработки информации пользователя, что привело к изменениям в структуре и функциях ОС. В персональном компьютере не столь важной становится функция максимальной загрузки процессора, хотя все также необходимо обеспечить параллельную работу центральной и периферийной частей. Операционная система состоит из модулей, т.е. из программ, выполняющих свою определенную функцию.

ОС для ПК делятся на:

- 1) одно- и многозадачные (в зависимости от числа параллельного выполняемых прикладных процессов);
- 2) одно- и многопользовательские (в зависимости от числа пользователей, одновременно работающих с ОС);
- 3) непереносимые и переносимые на другие типы компьютеров;
- 4) несетевые и сетевые, обеспечивающие работу в сетях ЭВМ;
- 5) операционные оболочки.

• ***Операционные оболочки*** – специальные программы, предназначенные для облегчения общения пользователя с командами ОС. Операционные оболочки имеют текстовый и графический варианты интерфейса пользователя. Эти программы существенно упрощают задание команды ОС для выполнения.

Лекция 14

Базовые типы данных. Основные алгоритмы обработки

Данные, относящиеся к какой-либо проблеме, являются абстрактным представлением объектов реального мира. Структура данных и алгоритмы их обработки неразрывно связаны между собой, т.к. выбор алгоритма часто очень сильно зависит от структуры данных [13].

Можно выделить следующие основные виды структур данных:

- ♦ простые (т.е. встроенные в язык) структуры – это основные конструкции, из которых строятся более сложные структуры-агрегаты. Простые структуры и агрегаты образуют **фундаментальные структуры**. Данные, представленные в виде фундаментальных структур, во время выполнения программы могут изменять значение, но изменить их строение нельзя;
- ♦ составные (**динамические**) **структуры** в процессе выполнения программы могут изменять как значение, так и строение;
- ♦ **иерархические структуры** - это динамические структуры, которые содержат данные, распределенные по уровням;
- ♦ **объектно-ориентированные** структуры воплощают концепцию совместной обработки данных и алгоритмов.

Фундаментальные типы данных подразделяются на простые или встроенные в язык программирования, и конструируемые пользователем, т.е. абстрактные.

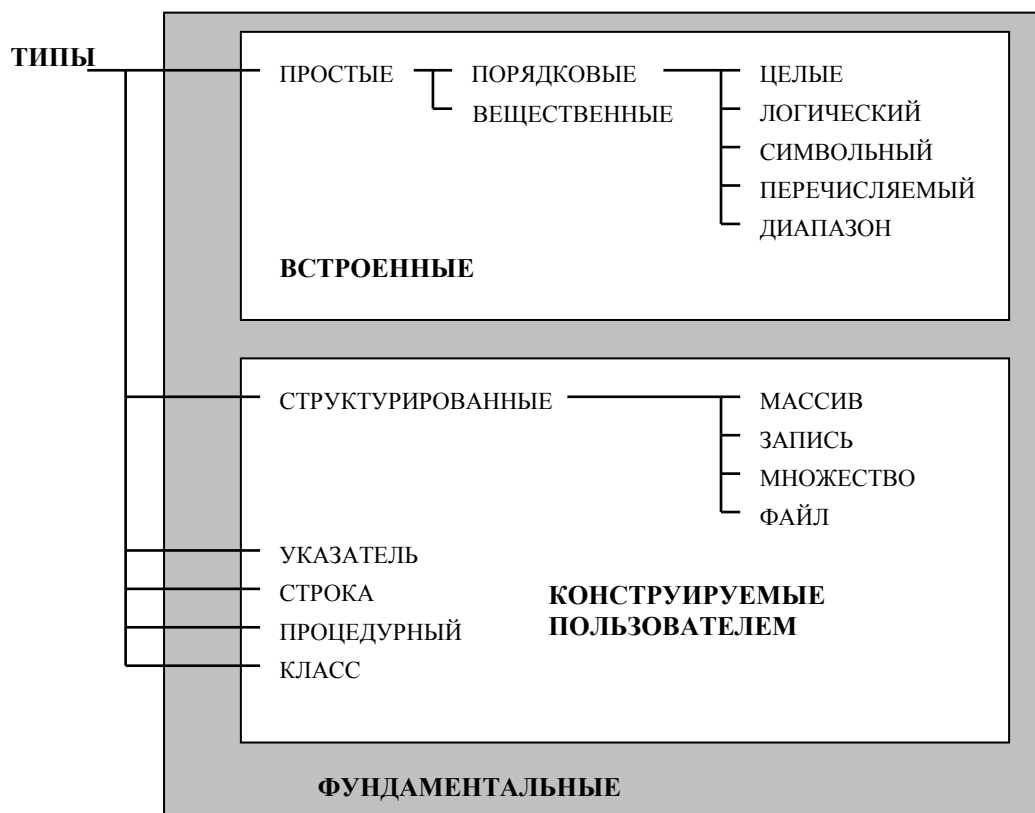


Рисунок 14.1 - Классификация типов данных в языке PASCAL

Простые типы данных

1) целый тип integer (-32168 ... 32167)

чаще всего используется для описания целочисленных массивов, счетчиков, индексов элементов.

Занимает в памяти 2 байта.

2) целый тип byte (0 ... 255)

используется в качестве индексов элементов одномерных и многомерных массивов, а также в качестве счетчика.

Mem=1 байт, поэтому является более предпочтительным, чем тип integer в диапазоне от 0 до 255.

3) вещественный тип real

используется при вычислении встроенных в Паскаль функций, любых операций, содержащих оператор вещественного деления (/).

4) символьный char

занимает в памяти 1 байт, имеет ASCII-кодировку, позволяет закодировать 256 символов, включая EN и RUS буквы, цифры, знаки препинания, символы псевдографики и управляющие символы.

5) строковый string

представляет собой цепочку длиной не более 255 символов. Со строкой можно работать как с массивом символов, так и как с целым числом.

6) логический boolean

Mem=1 байт. Переменные этого типа могут принимать только 2 значения: true & false.

Наиболее часто используется в виде т.н. программных флагов, которые показывают программисту, наступило то или иное событие в программе или нет.

Основные алгоритмы обработки

{Сумма, среднее двух чисел}

```
var a,b,s : integer;
    sr : real;
begin
  writeln('Введите 2 числа');  readln(a,b);
  s:=a+b;  sr:=s/2;
  writeln('Сумма равна = ',s);
  writeln('Среднее равно = ',sr:6:2);
  readln;
end.
```

{Максимум из двух чисел}

```
var a,b : integer;
begin
  writeln('Введите 2 числа');
  readln(a,b);
  if a>b then writeln('Максимум равен a = ',a)
  else if a<b then writeln('Максимум равен b = ',b)
  else writeln('Числа равны, максимума не существует');
  readln;
end.
```

{Максимум из трех чисел}

```
var a,b,c : integer;
```

```
begin
```

```
  writeln('Введите 3 числа');  readln(a,b,c);
```

```
  if ((a-b>0) and (a-c>0)) then writeln('Максимум равен a = ',a);
```

```
  if ((b-a>0) and (b-c>0)) then writeln('Максимум равен b = ',b);
```

```
  if ((c-a>0) and (c-b>0)) then writeln('Максимум равен c = ',c);
```

```
  readln;
```

```
end.
```

{"Какое из чисел x, y, z находится посередине?"}

```
var a,b,c : integer;
```

```
begin
```

```
  writeln('Введите 3 числа');
```

```
  readln(a,b,c);
```

```
  if (((b>a) and (b<c)) or ((b>c) and (b<a))) then writeln('Посередине находится b = ',b);
```

```
  if (((a>b) and (a<c)) or ((a>c) and (a<b))) then writeln('Посередине находится a = ',a);
```

```
  if (((c>a) and (c<b)) or ((c>b) and (c<a))) then writeln('Посередине находится c = ',c);
```

```
  readln;
```

```
end.
```

Лекция 15

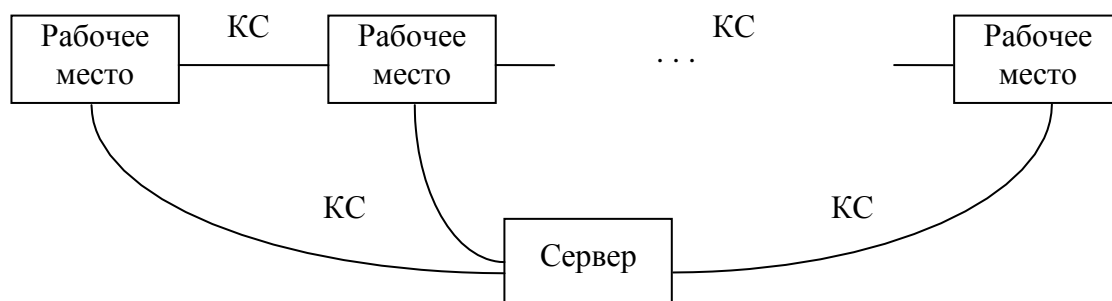
Вычислительные сети

Возрастающие потоки информации заставляют искать пути наиболее эффективного использования персональных компьютеров. В частности, это привело к созданию мощных вычислительных центров.

• **Вычислительная сеть** - это совокупность компьютеров и каналов связи, предназначенных для совместного использования. В сети решаются прикладные задачи пользователей по обработке информации и задачи управления сетью. Все компьютеры находятся в постоянной физической и информационной связи друг с другом.

Создание вычислительных сетей послужило мощным импульсом для организации больших систем по обработке информации.

Обобщенная структурная схема вычислительной сети



КС — канал связи

Самым дорогим из элементов компьютерной сети является сервер (Server), так как он имеет самые большие накопители на жёстких магнитных дисках.

Современные серверы позволяют подключать несколько жёстких дисков. Кроме хранения большого объёма информации на сервер возложены управляющие функции:

- распределение общих ресурсов(памяти, процессоры) между рабочими местами,
- управление каналами связи.

• **Рабочие места** - это персональные компьютеры, оснащённые различными средствами ввода, вывода информации(сканеры, принтеры, стримеры). Рабочие места могут иметь жёсткие диски для хранения собственной информации, а могут и не иметь.

В последнем случае вся информация хранится на сервере.

Персональные компьютеры включенные в сеть могут работать и самостоятельно, если сеть по каким-либо причинам не функционирует.

Каналы связи также являются дорогой частью вычислительной сети.

Различают два вида каналов связи:

1. Коаксиальный кабель.

Используется для соединения рабочих мест на удалённых расстояниях. Однако обладает пониженной прочностью в местах соединения. Если в сети, построенной на основе коаксиального кабеля, происходит разрыв в одном месте, то вся сеть перестаёт функционировать.

2. Витая пара.

Представляет собой два изолированных провода, перевитых между собой. Как правило, в канале связи есть две витых пары, заключённых в единый корпус - одна пара предназначена для приёма информации, вторая пара - для выдачи.

Несмотря на простоту устройства, витая пара обладает повышенной надёжностью и более высоким быстродействием по сравнению с коаксиальным кабелем.

Вычислительная сеть, построенная с помощью витых пар функционирует даже тогда, когда из строя выходят один или более компьютеров.

Различают два вида вычислительных сетей:

1. Глобальные.

2. Локальные.

Примером *глобальной сети*, соединяющей компьютеры, находящиеся на большом расстоянии друг от друга, может служить INTERNET.

Локальные вычислительные сети (ЛВС) обладают значительно большей эффективностью по сравнению с глобальными сетями. Они обеспечивают высокую скорость передачи информации и большую надёжность каналов связи за счет уменьшения их длины.

Как правило ЛВС обладает следующими особенностями:

- 1) ЛВС расположена на территории одного предприятия на территории от 1 до 2 километров.
- 2) ЛВС соединяет физически независимые рабочие места.
- 3) Любое рабочее место общаться через сеть с любым другим рабочим местом.
- 4) В ЛВС используют дешёвые средства связи. (например, коаксиальный кабель.

ЛВС позволяет организовать 2 вида обработки информации::

1. Удаленная обработка.
2. Распределенная обработка.

Удаленная обработка заключается в том, что все данные о предметной области, а также программы их обработки расположены на сервере, т.е. удалены от рабочих мест. Когда на каком-либо рабочем месте возникает необходимость обработки информации, то в память этого рабочего места по каналам связи из сервера передаются данные и программы их обработки. После завершения обработки все эти компоненты возвращаются обратно на сервер. Кроме того, к удаленной обработке можно отнести тот случай, когда рабочие места передают собранные данные на сервер, где и происходит их дальнейшая обработка и хранение.

Распределенная обработка получила наибольшее распространение в последнее время. Эта обработка заключается в том, что каждое рабочее место хранит программы обработки данных, а данные получает либо от устройств ввода, либо от сервера. После обработки данные вновь возвращаются на сервер.

ЛВС позволяет решить проблемы обмена текстовой, графической и звуковой информации, увеличить электронный документооборот.

Топология сети

Различают основные виды организации ЛВС:

1. Звездообразная.
2. Шинная.
3. Петлевая.
4. Кольцевая.
5. Древоподобная.

- Способ соединения компонентов сети называется ***топологией сети***.

Вид топологии определяется техническими функциональными характеристиками первоначального размещения рабочих мест. Рабочие места должны находиться на расстоянии более 3 метров от сервера и друг от друга, тогда влияние помех на каналы связи от соседних рабочих мест будут минимальными.

Чаще всего используют кольцевую, шинную и древоподобную топологию.

В кольцевой топологии каждый угол сети связан только с двумя другими. Информация в кольцевой сети передается только в одном направлении.

В шинной сети информация передается от устройств на шине, где распространяется с очень большой скоростью. Каждый узел подключается к шине через специальный интерфейсный разъем. Подключение к шине может осуществляться в любом месте и не влиять при этом на работу остальных устройств. Благодаря этому шинная топология получила наиболее широкое применение.

Лекция 16

Компьютерные вирусы и методы защиты информации от них

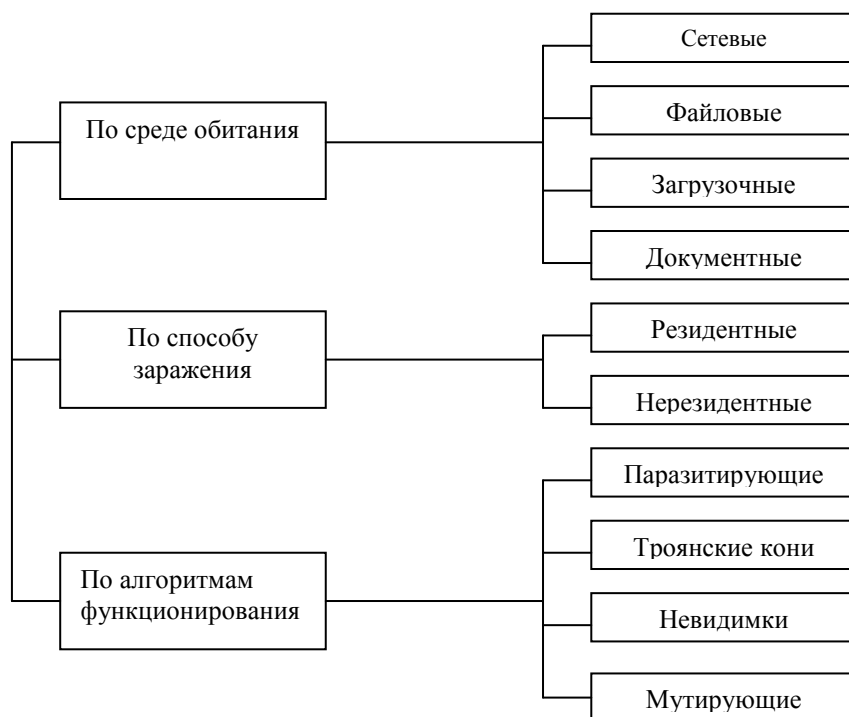
Компьютерным *вирусом* называется программа, способная самостоятельно создавать свои копии и внедряться в другие программы, в системные области дисковой памяти компьютера, распространяться по каналам связи [20].

Целью применения вирусов является нарушение работы программ, порча файловых систем, нарушение нормальной работы пользователей. Вирусы имеют три стадии существования:

- пассивная стадия;
- стадия размножения;
- активная стадия.

Классификация вирусов

Все вирусы можно классифицировать по следующей схеме:



Сетевые вирусы используют для своего распространения команды и протоколы телекоммуникационных сетей.

Файловые чаще всего внедряются в исполняемые файлы с расширениями *exe* и *com.*, в драйверы внешних устройств и библиотеки.

Загрузочные внедряются в загрузочный сектор дискеты или в главную загрузочную запись жесткого диска. Вирус применяет программу загрузки операционной системы, запуская файлы, необходимые для осуществления несанкционированного доступа.

Документные (или *макровирусы*) внедряются в текстовые файлы.

Резидентные после завершения инфицированной программы остаются в оперативной памяти и продолжают свои деструктивные действия.

Нерезидентные удаляются из памяти вместе с инфицированной программой.

Паразитирующие вирусы изменяют содержание зараженных файлов.

Троянские кони маскируются под полезные программы, однако наряду, с полезными функциями, вирус нарушает работу компьютерной системы или собирает информацию, циркулирующую в системе.

Вирусы-невидимки прячутся при попытке их обнаружить, перехватывая запрос антивирусной программы. Они либо временно удаляются из обрабатываемого файла, либо подставляют вместо себя незаражённые участки программы.

Мутлирующие вирусы периодически изменяют свой программный код.

Признаки наличия вирусов на персональном компьютере:

- отказ в работе персонального компьютера или его отдельных компонентов;
- отказ в загрузке операционной системы;
- заметное замедление работы компьютера;
- нарушение работы отдельных программ;
- искажение размеров или исчезновение файлов;
- уменьшение доступной части оперативной памяти.

Основной канал проникновения вирусов в персональный компьютер — коммуникационные сети и съёмные носители информации. Для исключения проникновения вирусов необходимо:

- осуществлять автоматический входной контроль информации, поступающей в сеть, сетевым экраном (брандмауэром);
- удалять, не читая, электронную почту, полученную от неизвестных источников;
- ограничить число пользователей с правом записи на жесткий диск;
- проверять все съёмные носители специальной антивирусной программой.

Антивирусные программы

Антивирусные программы условно делятся на пять типов: *детекторы, ревизоры, фильтры, доктора и вакцины*.

Детекторы ищут вирусы в памяти и при обнаружении сообщают об этом ; пользователю. Ревизоры запоминают исходное состояние программ, каталогов, системных областей и периодически сравнивают их с текущими значениями. При изменении контролируемых параметров сообщают это пользователю. Фильтры выявляют подозрительные процедуры (изменение загрузочных записей, атрибутов файлов и т. п.). Доктора не только обнаруживают, но и удаляют известные им вирусы. Вакцины модифицируют файл или диск так, что он воспринимается вирусом уже зараженным.

Лекция 17

Понятие несанкционированного доступа. Методы защиты информации

Программные продукты и базы данных являются предметом интеллектуального труда специалистов высокой квалификации. Процесс проектирования ПП основан на использовании наукоемких технологий, требует применения дорогостоящей вычислительной техники, значительных материальных и трудовых затрат. Это обуславливает необходимость защиты интересов разработчика ПП и БД от несанкционированного их использования.

Программное обеспечение является объектом защиты также и в связи со сложностью и трудоемкостью восстановления его работоспособности, значимостью программного обеспечения для работы информационной сети.

Самый простой и доступный метод защиты ПП и БД – ограничение доступа. Контроль доступа обеспечивается путем:

- ◆ парольной защиты программ при их запуске;
- ◆ использование ключевой дискеты для запуска программ;
- ◆ ограничения программ или данных, доступных пользователям; могут использоваться и криптографические методы защиты информации БД или программ.

Программные системы защиты

Предотвращают нелегальное использование ПП и БД. Программа выполняется только при опознании некоторого уникального не копируемого элемента – ключа:

- дискеты с записью ключа;
- характеристики аппаратуры ПК;
- специальное устройство (электронный ключ), подключаемое к ПК и предназначенное для выдачи опознавательного кода.

Для идентификации «запускающих дискет» применяются следующие методы:

- 1) нанесение повреждений на поверхность дискеты («лазерная дыра»), которая с трудом может быть воспроизведена в несанкционированной копии дискеты;
- 2) нестандартное форматирование запускающей дискеты.

Правовые методы защиты ПП и БД

- 1) Патентная защита.
- 2) Закон о производственных секретах.
- 3) Лицензионные соглашения и контракты.
- 4) Закон об авторском праве.

Различают 2 категории прав:

1. Экономические права, дающие выгоды от продажи ПП и БД.
2. Моральные права, обеспечивающие защиту личности автора и его произведения.

Во многих цивилизованных странах несанкционированное копирование программ в целях продажи или бесплатного распространения рассматривается как государственное преступление, карается штрафом или тюремным заключением. К сожалению, само авторское право не обеспечивает защиту новой идеи, концепции, методологии и технологии разработки программ, поэтому требуются дополнительные меры их защиты.

Патентная защита устанавливает приоритет в разработке и использовании нового подхода или метода, примененного при разработке программ, удостоверяет их оригинальность.

Статус производственного секрета для программы ограничивает круг лиц, допущенных к ее эксплуатации, а также определяет меру их ответственности за разглашение секретов.

Лицензионные соглашения распространяются на все аспекты правовой охраны программных продуктов, включая авторское право, патентную защиту, производственные секреты. Наиболее часто используются лицензионные соглашения на передачу авторских прав.

• **Лицензия** - договор на передачу одним лицом (лицензиаром) другому лицу (лицензиату) права на использование имени, продукции, технологии или услуги.

В лицензионном соглашении оговариваются все условия эксплуатации программ, в том числе – создание копий. На каждой копии программы должны быть те же отметки, что и на оригинале:

- ◆ знак авторского права (обычно ©) и название разработчика, года выпуска программы, прочих ее атрибутов;
- ◆ знак патентной защиты или производственного секрета;
- ◆ торговые марки, соответствующие использованным в программе другим программным изделиям (обычно - TM и название фирмы-разработчика ПП);
- ◆ символ зарегистрированного права на распространение программного продукта (обычно ®).

Типы лицензий на программный продукт

1. **Исключительная лицензия** – продажа всех имущественных прав на ПП и БД, покупателю лицензии предоставляется исключительное право на их использование, а автор или владелец патента отказывается от самостоятельного их применения или предоставления другим лицам. Это самый дорогой вид лицензии, к нему прибегают для монопольного владения с целью извлечения дополнительной прибыли, либо с целью прекращения существования на рынке данного ПП.

2. **Простая лицензия** – лицензиар предоставляет право лицензиату использовать ПП и БД, оставляя за собой применять их и предоставлять на аналогичных условиях неограниченному числу лиц (лицензиат при этом не может сам выдавать сублицензии, может лишь продать копии приобретенного ПП и БД). Такой вид лицензии приобретает дилер (торговец) либо фирмы-производители, использующие купленные лицензии как сопутствующий товар к основному виду деятельности. Например, производители и фирмы, торгующие

компьютерами, осуществляют продажу с установленным лицензионным ПО (ОС, текстовый редактор, электронная таблица, графический редактор и т.д.).

3. *Этикеточная лицензия* – лицензия на одну копию ПП или БД. Данный тип лицензии применяется при розничной продаже. Каждый официальный покупатель заключает лицензионное соглашение с продавцом на их использование, но при этом сохраняется авторское право разработчика.

Экономические отношения между лицензиаром и лицензиатом могут строиться различным образом: за право пользования ПП или БД выплачивается единовременное вознаграждение (поушальный платеж) или выплачиваются периодические отчисления лицензиару в виде роялти – фиксированной ставки в определенные интервалы времени в течении действия лицензионного соглашения, как правило, % от стоимости ПП или БД.

Закон об охране ПП и компьютерных БД автором признает физическое лицо, в результате творческой деятельности которого они созданы. Автору независимо от его имущественных прав принадлежат личные авторские права: авторство, имя, неприкосновенность (целостность) программ или БД.

Авторское право действует с момента создания ПП или БД в течении всей жизни автора и 50 лет после его смерти.

Автор может:

- ◆ выпускать в свет;
- ◆ воспроизводить в любой форме, любыми способами;
- ◆ распространять;
- ◆ модифицировать;

осуществлять любое иное использование ПП или БД.

Авторское право не связано с правом собственности на материальный носитель.

Имущественные права на ПП или БД могут быть переданы частично или полностью другим физическим или юридическим лицам по договору. Имущественные права относятся к категории наследуемых. Если ПП или БД созданы в порядке выполнения служебных обязанностей, имущественные права принадлежат работодателю.

ПП или БД могут использоваться третьими лицами – пользователями на основании договора с правообладателем.

Лицо, правомерно владеющее экземпляром программы или БД, вправе, без получения дополнительного разрешения правообладателя, осуществлять любые действия, связанные с функционированием ПП или БД в соответствии с ее назначением, в том числе:

- ◆ устанавливать 1 экземпляр, если не предусмотрено иное соглашение с правообладателем, ПП или БД на компьютер;
- ◆ исправлять явные ошибки;
- ◆ адаптировать ПП или БД;
- ◆ изготавливать страховые копии.

Лекция 18

Криптографические методы защиты информации

Криптографические методы защиты информации - это специальные методы шифрования, кодирования или иного преобразования информации, в результате которого ее содержание становится недоступным без предъявления ключа криптограммы и обратного преобразования. Криптографический метод защиты, безусловно, самый надежный метод защиты, так как охраняется непосредственно сама информация, а не доступ к ней (например, зашифрованный файл нельзя прочесть даже в случае кражи носителя). Данный метод защиты реализуется в виде программ или пакетов программ².

Современная криптография включает в себя четыре крупных раздела:

- симметричные криптосистемы;
- криптосистемы с открытым ключом;
- электронная подпись;
- управление ключами.

Симметричные криптосистемы

В симметричных криптосистемах и для шифрования, и для дешифрования используется один и тот же ключ. (Шифрование - преобразовательный процесс: исходный текст, который носит также название открытого текста, заменяется зашифрованным текстом, дешифрование - обратный шифрованию процесс. На основе ключа зашифрованный текст преобразуется в исходный);

В симметричных криптосистемах существуют 4 основных класса преобразований:

- **подстановка** - символы шифруемого текста заменяются символами того же или другого алфавита в соответствии с заранее определенным правилом;

Один из самых простых методов шифрования – это **шифрование заменой** (подстановкой). Символы шифруемого текста заменяются другими символами, взятыми из одного (моноалфавитная подстановка) или из нескольких (полиалфавитная подстановка) алфавитов.

Такой шифр имеет низкую стойкость, т. к. зашифрованный таким образом текст имеет те же самые статистические характеристики, что и исходный, поэтому по частотному анализу можно восстановить таблицу замены. Сообщение шифруют таким способом только тогда, когда оно достаточно короткое.

Использование полиалфавитных подстановок повышает стойкость шифра. Смена алфавитов производится последовательно и циклически: первый символ заменяется соответствующим символом, первого алфавита, второй — второго

² edu.ria-link.ru>Научно-практическая конференция>files/356.doc

алфавита и т. д., пока не будут исчерпаны все алфавиты. После этого использование алфавитов повторяется.

- **перестановка** - символы шифруемого текста переставляются по некоторому правилу в пределах заданного блока передаваемого текста;

Более сложным является *шифрование методом перестановки*. Метод заключается в перестановке по определенным правилам символов шифруемого текста внутри блока определённых размеров. Алгоритм шифрования и дешифрования таков:

1. Выбирается размер блока шифрования: m строк, n столбцов.
2. Выбирается ключ шифра — последовательность столбцов 1, 2, 3... n и, полученная случайной перестановкой.
3. Шифруемый текст записывается последовательными строками в блок $m \times n$.
4. Зашифрованный текст выписывается колонками в последовательности возрастания номеров колонок, задаваемых номерами ключевой последовательности.
5. Заполняется новым текстом новый такой же блок и т. д.

Пример. Зашифруем таким образом сообщение: **Жил-был у бабушки серенький козлик.**

Выберем $n = 7$ и $m = 5$. Пусть ключ будет такой: 7-5-3-2-6-1-4.

Таблица «Блок шифрования сообщения»

Ключ	7	5	3	2	6	1	4
Номера n и m	1	2	3	4	5	6	7
1	ж	и	л	–	б	ы	л
2	пробел	у	пробел	б	а	б	у
3	ш	к	и	пробел	с	е	р
4	е	н	ь	к	и	й	пробел
5	к	о	з	л	и	к	пробел

Зашифрованное сообщение выглядит так: **ыбейк-б клл иьзлур иукнобасишж шек.**

Дешифрование выполняется в следующем порядке:

1. В сообщении выделяется блок символов размером $m \times n$.
2. Выделенная часть сообщения разбивается на n групп по m символов в группе. Группы записываются в те же столбцы таблицы, номера которых совпадают с номерами ключа.
3. Расшифрованный текст читается по строкам таблицы.

В случае нашего примера весь текст $7 \times 5 \Rightarrow 35$ символов — длина выделяемого блока. Разобьем его на семь групп по пять символов в каждой группе.

1	2	3	4	5	6	7
ыбейк	–б клл иьз	лур	иукно	басии	ж шек	

Распишем эти группы по номеру ключа в таблице. В результате, опять получим исходную таблицу шифрования.

- аналитическое преобразование - шифруемый текст преобразуется по некоторому аналитическому правилу, например гаммирование - заключается в наложении на исходный текст некоторой псевдослучайной последовательности, генерируемой на основе ключа;
- комбинированное преобразование - представляют собой последовательность (с возможным повторением и чередованием) основных методов преобразования, применяемую к блоку (части) шифруемого текста. Блочные шифры на практике встречаются чаще, чем “чистые” преобразования того или иного класса в силу их более высокой криптостойкости. Российский и американский стандарты шифрования основаны именно на этом классе.

Криптосистемы с открытым ключом

В системах с открытым ключом используются два ключа - открытый и закрытый, которые математически связаны друг с другом. Информация шифруется с помощью открытого ключа, который доступен всем желающим, а расшифровывается с помощью закрытого ключа, известного только получателю сообщения. (Ключ - информация, необходимая для беспрепятственного шифрования и дешифрования текстов);

Как бы ни были сложны и надежны криптографические системы - их слабое мест при практической реализации - проблема распределения ключей. Для того, чтобы был возможен обмен конфиденциальной информацией между двумя субъектами ИС, ключ должен быть сгенерирован одним из них, а затем каким-то образом опять же в конфиденциальном порядке передан другому. Т.е. в общем случае для передачи ключа опять же требуется использование какой-то криптосистемы. Для решения этой проблемы на основе результатов, полученных классической и современной алгеброй, были предложены системы с открытым ключом. Суть их состоит в том, что каждым адресатом ИС генерируются два ключа, связанные между собой по определенному правилу. Один ключ объявляется открытым, а другой закрытым. Открытый ключ публикуется и доступен любому, кто желает послать сообщение адресату. Секретный ключ сохраняется в тайне. Исходный текст шифруется открытым ключом адресата и передается ему. Зашифрованный текст в принципе не может быть расшифрован тем же открытым ключом. Дешифрование сообщения возможно только с использованием закрытого ключа, который известен только самому адресату. Криптографические системы с открытым ключом используют так называемые необратимые или односторонние функции, которые обладают следующим свойством: при заданном значении x относительно просто вычислить значение $f(x)$, однако если $y = f(x)$, то нет простого пути для вычисления значения x . Множество классов необратимых функций и порождает все разнообразие систем с открытым ключом. Однако не всякая необратимая функция годится для использования в реальных ИС. В самом определении необратимости присутствует неопределенность. Под необратимостью понимается не теоретическая необратимость, а практическая невозможность вычислить

обратное значение используя современные вычислительные средства за обозримый интервал времени. Поэтому чтобы гарантировать надежную защиту информации, к системам с открытым ключом (СОК) предъявляются два важных и очевидных требования:

1. Преобразование исходного текста должно быть необратимым и исключать его восстановление на основе открытого ключа.
2. Определение закрытого ключа на основе открытого также должно быть невозможным на современном технологическом уровне. При этом желательна точная нижняя оценка сложности (количества операций) раскрытия шифра.

Алгоритмы шифрования с открытым ключом получили широкое распространение в современных информационных системах. Так, алгоритм RSA стал мировым стандартом де-факто для открытых систем. Вообще же все предлагаемые сегодня криптосистемы с открытым ключом опираются на один из следующих типов необратимых преобразований:

1. Разложение больших чисел на простые множители;
2. Вычисление логарифма в конечном поле;
3. Вычисление корней алгебраических уравнений.

Здесь же следует отметить, что алгоритмы криптосистемы с открытым ключом (СОК) можно использовать в следующих назначениях:

1. Как самостоятельные средства защиты передаваемых и хранимых данных.
2. Как средства для распределения ключей.

Алгоритмы СОК более трудоемки, чем традиционные криптосистемы. Поэтому часто на практике рационально с помощью СОК распределять ключи, объем которых как информации незначителен. А потом с помощью обычных алгоритмов осуществлять обмен большими информационными потоками. Один из наиболее распространенных - система с открытым ключом - RSA. Криптосистема RSA, разработанная в 1977 году и получила название в честь ее создателей: Рона Ривеста, Ади Шамира и Леонарда Эйделмана. Они воспользовались тем фактом, что нахождение больших простых чисел в вычислительном отношении осуществляется легко, но разложение на множители произведения двух таких чисел практически невыполнимо. Доказано (теорема Рабина), что раскрытие шифра RSA эквивалентно такому разложению. Поэтому для любой длины ключа можно дать нижнюю оценку числа операций для раскрытия шифра, а с учетом производительности современных компьютеров оценить и необходимое на это время. Возможность гарантированно оценить защищенность алгоритма RSA стала одной из причин популярности этой СОК на фоне десятков других схем. Поэтому алгоритм RSA используется в банковских компьютерных сетях, особенно для работы с удаленными клиентами (обслуживание кредитных карточек).

Бывают системы с открытым ключом. В таких системах для шифрования используют один ключ, а для дешифрования — другой. Первый ключ публикуется и используется всеми пользователями системы, шифрующей данные. Второй ключ — секретный, причем этот ключ дешифрования не может быть определен из открытого ключа шифрования. Методы с открытым ключом называют асимметричными методами.

Наряду с ними существуют симметричные методы, где для шифрования и дешифрования используется один ключ, который естественно является секретным.

Во всех методах одна из самых острых проблем обеспечения секретности является безопасная передача ключа пользователю. Обычно ключ также шифруется и передается зашифрованным сообщением.

При шифровании сообщения может быть использована хэш-функция. Это такой прием или метод, который отображает сообщение любой длины в строку фиксированного размера, причем эта функция не имеет обратной, т. е, по сжато, изображению восстановить исходное сообщение невозможно.

Электронная подпись

Системой электронной подписи называется присоединяемое к тексту его криптографическое преобразование, которое позволяет при получении текста другим пользователем проверить авторство и подлинность сообщения.

В чем состоит проблема аутентификации данных? В конце обычного письма или документа исполнитель или ответственное лицо обычно ставит свою подпись. Подобное действие обычно преследует две цели. Во-первых, получатель имеет возможность убедиться в истинности письма, сличив подпись с имеющимся у него образцом. Во-вторых, личная подпись является юридическим гарантом авторства документа. Последний аспект особенно важен при заключении разного рода торговых сделок, составлении доверенностей, обязательств и т.д. Если подделать подпись человека на бумаге весьма непросто, а установить авторство подписи современными криминалистическими методами - техническая деталь, то с подписью электронной дело обстоит иначе. Подделать цепочку битов, просто ее скопировав, или незаметно внести нелегальные исправления в документ сможет любой пользователь. С широким распространением в современном мире электронных форм документов (в том числе и конфиденциальных) и средств их обработки особо актуальной стала проблема установления подлинности и авторства безбумажной документации. В разделе криптографических систем с открытым ключом было показано, что при всех преимуществах современных систем шифрования они не позволяют обеспечить аутентификацию данных. Поэтому средства аутентификации должны использоваться в комплексе и криптографическими алгоритмами.

При обмене электронными документами очень важным вопросом является установление авторства, подлинности и целостности информации. Цифровая подпись по функции аналогична рукописной и обладает всеми её признаками:

- удостоверяет, что подписанный текст исходит от лица, поставившего подпись;
- не дает лицу, подписавшему текст, отказаться от обязательств;
- гарантирует целостность подписанного текста.

Цифровая подпись работает следующим образом. Цифровая подпись шифруется с применением методов открытого ключа и связывает содержимое документа, самой подписи и пары ключей. На этапе формирования цифровой подписи генерируются два ключа: секретный и открытый. С помощью хэш-функции, применённой ко всему документу, вычисляется небольшое число,

характеризующее весь текст в целом. Это число, зашифрованное закрытым ключом, и есть электронная подпись.

При проверке электронная подпись расшифровывается открытым ключом. К полученному открытому документу, применяется хэш-функция, и результат её работы сравнивается с присланной электронной подписью. Всякое изменение документа приведет к несовпадению ключа, полученного хэш-функцией и присланного, т. е. к доказательству факта несанкционированного доступа к информации.

Управление ключами

Это процесс системы обработки информации, содержанием которых является составление и распределение ключей между пользователями.

Кроме выбора подходящей для конкретной ИС криптографической системы, важная проблема - управление ключами. Как бы ни была сложна и надежна сама криптосистема, она основана на использовании ключей. Если для обеспечения конфиденциального обмена информацией между двумя пользователями процесс обмена ключами тривиален, то в ИС, где количество пользователей составляет десятки и сотни управление ключами - серьезная проблема. Под ключевой информацией понимается совокупность всех действующих в ИС ключей. Если не обеспечено достаточно надежное управление ключевой информацией, то завладев ею, злоумышленник получает неограниченный доступ ко всей информации. Управление ключами - информационный процесс, включающий в себя три элемента:

- генерацию ключей;
- накопление ключей;
- распределение ключей.

Рассмотрим, как они должны быть реализованы для того, чтобы обеспечить безопасность ключевой информации в ИС.

Генерация ключей

В самом начале разговора о криптографических методах было сказано, что не стоит использовать неслучайные ключи с целью легкости их запоминания. В серьезных ИС используются специальные аппаратные и программные методы генерации случайных ключей. Как правило используют датчики ПСЧ. Однако степень случайности их генерации должна быть достаточно высоким. Идеальными генераторами являются устройства на основе “натуральных” случайных процессов. Например случайным математическим объектом являются десятичные знаки иррациональных чисел, которые вычисляются с помощью стандартных математических методов.

Накопление ключей

Под накоплением ключей понимается организация их хранения, учета и удаления. Поскольку ключ является самым привлекательным для злоумышленника объектом, открывающим ему путь к конфиденциальной информации, то вопросам накопления ключей следует уделять особое внимание. Секретные ключи никогда не должны записываться в явном виде на носителе,

который может быть считан или скопирован. В достаточно сложной ИС один пользователь может работать с большим объемом ключевой информации, и иногда даже возникает необходимость организации мини-баз данных по ключевой информации. Такие базы данных отвечают за принятие, хранение, учет и удаление используемых ключей. Итак, каждая информация об используемых ключах должна храниться в зашифрованном виде. Ключи, зашифровывающие ключевую информацию называются мастер-ключами. Желательно, чтобы мастер-ключи каждый пользователь знал наизусть, и не хранил их вообще на каких-либо материальных носителях. Очень важным условием безопасности информации является периодическое обновление ключевой информации в ИС. При этом переназначаться должны как обычные ключи, так и мастер-ключи. В особо ответственных ИС обновление ключевой информации желательно делать ежедневно. Вопрос обновления ключевой информации связан и с третьим элементом управления ключами - распределением ключей.

Распределение ключей

Распределение ключей - самый ответственный процесс в управлении ключами. К нему предъявляются два требования:

- Оперативность и точность распределения;
- Скрытность распределяемых ключей.

В последнее время заметен сдвиг в сторону использования криптосистем с открытым ключом, в которых проблема распределения ключей отпадает. Тем не менее распределение ключевой информации в ИС требует новых эффективных решений. Распределение ключей между пользователями реализуются двумя разными подходами:

1. Путем создания одного ли нескольких центров распределения ключей. Недостаток такого подхода состоит в том, что в центре распределения известно, кому и какие ключи назначены и это позволяет читать все сообщения, циркулирующие в ИС. Возможные злоупотребления существенно влияют на защиту.
2. Прямой обмен ключами между пользователями информационной системы. В этом случае проблема состоит в том, чтобы надежно удостоверить подлинность субъектов. Для обмена ключами можно использовать криптосистемы с открытым ключом, используя тот же алгоритм RSA.

В качестве обобщения сказанного о распределении ключей следует сказать следующее. Задача управления ключами сводится к поиску такого протокола распределения ключей, который обеспечивал бы:

- возможность отказа от центра распределения ключей;
- взаимное подтверждение подлинности участников сеанса;
- подтверждение достоверности сеанса механизмом запроса-ответа, использование для этого программных или аппаратных средств;
- использование при обмене ключами минимального числа сообщений.

Основные направления использования криптографических методов - передача конфиденциальной информации по каналам связи (например, электронная

почта), установление подлинности передаваемых сообщений, хранение информации (документов, баз данных) на носителях в зашифрованном виде.

Требования к криптосистемам

Процесс криптографического закрытия данных может осуществляться как программно, так и аппаратно. Аппаратная реализация отличается существенно большей стоимостью, однако ей присущи и преимущества: высокая производительность, простота, защищенность и т.д. Программная реализация более практична, допускает известную гибкость в использовании. Для современных криптографических систем защиты информации сформулированы следующие общепринятые требования:

- зашифрованное сообщение должно поддаваться чтению только при наличии ключа;
- число операций, необходимых для определения использованного ключа шифрования по фрагменту шифрованного сообщения и соответствующего ему открытого текста, должно быть не меньше общего числа возможных ключей;
- число операций, необходимых для расшифровывания информации путем перебора всевозможных ключей должно иметь строгую нижнюю оценку и выходить за пределы возможностей современных компьютеров (с учетом возможности использования сетевых вычислений);
- знание алгоритма шифрования не должно влиять на надежность защиты;
- незначительное изменение ключа должно приводить к существенному изменению вида зашифрованного сообщения даже при использовании одного и того же ключа;
- структурные элементы алгоритма шифрования должны быть неизменными;
- дополнительные биты, вводимые в сообщение в процессе шифрования, должен быть полностью и надежно скрыты в шифрованном тексте;
- длина шифрованного текста должна быть равной длине исходного текста;
- не должно быть простых и легко устанавливаемых зависимостей между ключами, последовательно используемыми в процессе шифрования;
- любой ключ из множества возможных должен обеспечивать надежную защиту информации;
- алгоритм должен допускать как программную, так и аппаратную реализацию, при этом изменение длины ключа не должно вести к качественному ухудшению алгоритма шифрования.

Реализация криптографических методов

Проблема реализации методов защиты информации имеет два аспекта:

- разработку средств, реализующих криптографические алгоритмы;
- методику использования этих средств.

Каждый из рассмотренных криптографических методов могут быть реализованы либо программным, либо аппаратным способом. Возможность программной реализации обуславливается тем, что все методы криптографического преобразования формальны и могут быть представлены в

виде конечной алгоритмической процедуры. При аппаратной реализации все процедуры шифрования и дешифрования выполняются специальными электронными схемами. Наибольшее распространение получили модули, реализующие комбинированные методы. Большинство зарубежных серийных средств шифрования основано на американском стандарте DES. Отечественные же разработки, такие как, например, устройство КРИПТОН, использует отечественный стандарт шифрования. Основным достоинством программных методов реализации защиты является их гибкость, т.е. возможность быстрого изменения алгоритмов шифрования. Основным же недостатком программной реализации является существенно меньшее быстродействие по сравнению с аппаратными средствами (примерно в 10 раз). В последнее время стали появляться комбинированные средства шифрования, так называемые программно-аппаратные средства. В этом случае в компьютере используется своеобразный "криптографический сопроцессор" - вычислительное устройство, ориентированное на выполнение криптографических операций (сложение по модулю, сдвиг и т.д.). Меняя программное обеспечение для такого устройства, можно выбирать тот или иной метод шифрования. Такой метод объединяет в себе достоинства программных и аппаратных методов.

Таким образом, выбор типа реализации криптозащиты для конкретной ИС в существенной мере зависит от ее особенностей и должен опираться на всесторонний анализ требований, предъявляемых к системе защиты информации.

Идентификация и аутентификация

Идентификацию и аутентификацию можно считать основой программно-технических средств безопасности. Идентификация и аутентификация - это первая линия обороны, "проходная" информационного пространства организации.

Идентификация позволяет субъекту - пользователю или процессу, действующему от имени определенного пользователя, назвать себя, сообщив свое имя. Посредством аутентификации вторая сторона убеждается, что субъект действительно тот, за кого себя выдает. В качестве синонима слова "аутентификация" иногда используют сочетание "проверка подлинности". Субъект может подтвердить свою подлинность, если предъявит по крайней мере одну из следующих сущностей:

- нечто, что он знает: пароль, личный идентификационный номер, криптографический ключ и т.п.;
- нечто, чем он владеет: личную карточку или иное устройство аналогичного назначения;
- нечто, что является частью его самого: голос, отпечатки пальцев и т.п., то есть свои биометрические характеристики;
- нечто, ассоциированное с ним, например координаты.

Главное достоинство парольной аутентификации - простота и привычность. Пароли давно встроены в операционные системы и иные сервисы. При правильном использовании пароли могут обеспечить приемлемый для многих организаций уровень безопасности. Тем не менее по совокупности

характеристик их следует признать самым слабым средством проверки подлинности. Надежность паролей основывается на способности помнить их и хранить в тайне. Ввод пароля можно подсмотреть. Пароль можно угадать методом грубой силы, используя, быть может, словарь. Если файл паролей зашифрован, но доступен на чтение, его можно перекачать к себе на компьютер и попытаться подобрать пароль, запрограммировав полный перебор.

Пароли уязвимы по отношению к электронному перехвату - это наиболее принципиальный недостаток, который нельзя компенсировать улучшением администрирования или обучением пользователей. Практически единственный выход - использование криптографии для шифрования паролей перед передачей по линиям связи.

Тем не менее, следующие меры позволяют значительно повысить надежность парольной защиты:

- наложение технических ограничений (пароль должен быть не слишком коротким, он должен содержать буквы, цифры, знаки пунктуации и т.п.);
- управление сроком действия паролей, их периодическая смена;
- ограничение доступа к файлу паролей;
- ограничение числа неудачных попыток входа в систему, что затруднит применение метода грубой силы;
- обучение и воспитание пользователей;
- использование программных генераторов паролей, которые, основываясь на несложных правилах, могут породить только благозвучные и, следовательно, запоминающиеся пароли.

Перечисленные меры целесообразно применять всегда, даже если наряду с паролями используются другие методы аутентификации, основанные, например, на применении токенов.

Токен - это предмет или устройство, владение которым подтверждает подлинность пользователя. Различают токены с памятью (пассивные, которые только хранят, но не обрабатывают информацию) и интеллектуальные токены (активные).

Самой распространенной разновидностью токенов с памятью являются карточки с магнитной полосой. Для использования подобных токенов необходимо устройство чтения, снабженное также клавиатурой и процессором. Обычно пользователь набирает на этой клавиатуре свой личный идентификационный номер, после чего процессор проверяет его совпадение с тем, что записано на карточке, а также подлинность самой карточки. Таким образом, здесь фактически применяется комбинация двух способов защиты, что существенно затрудняет действия злоумышленника.

Необходима обработка аутентификационной информации самим устройством чтения, без передачи в компьютер - это исключает возможность электронного перехвата.

Иногда (обычно для физического контроля доступа) карточки применяют сами по себе, без запроса личного идентификационного номера.

Как известно, одним из самых мощных средств в руках злоумышленника является изменение программы аутентификации, при котором пароли не только

проверяются, но и запоминаются для последующего несанкционированного использования.

Интеллектуальные токены характеризуются наличием собственной вычислительной мощности. Они подразделяются на интеллектуальные карты, стандартизованные ISO и прочие токены. Карты нуждаются в интерфейсном устройстве, прочие токены обычно обладают ручным интерфейсом (дисплеем и клавиатурой) и по внешнему виду напоминают калькуляторы. Чтобы токен начал работать, пользователь должен ввести свой личный идентификационный номер.

По принципу действия интеллектуальные токены можно разделить на следующие категории:

- Статический обмен паролями: пользователь обычным образом доказывает токену свою подлинность, затем токен проверяется компьютерной системой;
- Динамическая генерация паролей: токен генерирует пароли, периодически изменяя их. Компьютерная система должна иметь синхронизированный генератор паролей. Информация от токена поступает по электронному интерфейсу или набирается пользователем на клавиатуре терминала;
- Запросно-ответные системы: компьютер выдает случайное число, которое преобразуется криптографическим механизмом, встроенным в токен, после чего результат возвращается в компьютер для проверки. Здесь также возможно использование электронного или ручного интерфейса. В последнем случае пользователь читает запрос с экрана терминала, набирает его на клавиатуре токена (возможно, в это время вводится и личный номер), а на дисплее токена видит ответ и переносит его на клавиатуру терминала.

Управление доступом

Средства управления доступом позволяют специфицировать и контролировать действия, которые субъекты - пользователи и процессы могут выполнять над объектами - информацией и другими компьютерными ресурсами. Речь идет о логическом управлении доступом, который реализуется программными средствами. Логическое управление доступом - это основной механизм многопользовательских систем, призванный обеспечить конфиденциальность и целостность объектов и, до некоторой степени, их доступность путем запрещения обслуживания неавторизованных пользователей. Задача логического управления доступом состоит в том, чтобы для каждой пары (субъект, объект) определить множество допустимых операций, зависящее от некоторых дополнительных условий, и контролировать выполнение установленного порядка. Простой пример реализации таких прав доступа - какой-то пользователь (субъект) вошедший в информационную систему получил право доступа на чтение информации с какого-то диска(объект), право доступа на модификацию данных в каком-то каталоге(объект) и отсутствие всяких прав доступа к остальным ресурсам информационной системы.

Контроль прав доступа производится разными компонентами программной среды - ядром операционной системы, дополнительными средствами

безопасности, системой управления базами данных, посредническим программным обеспечением (таким как монитор транзакций) и т.д.

Протоколирование и аудит

Под протоколированием понимается сбор и накопление информации о событиях, происходящих в информационной системе. Например - кто и когда пытался войти в систему, чем завершилась эта попытка, кто и какими информационными ресурсами пользовался, какие и кем модифицировались информационные ресурсы и много других.

Аудит - это анализ накопленной информации, проводимый оперативно, почти в реальном времени, или периодически.

Реализация протоколирования и аудита преследует следующие главные цели:

- обеспечение подотчетности пользователей и администраторов;
- обеспечение возможности реконструкции последовательности событий;
- обнаружение попыток нарушений информационной безопасности;
- предоставление информации для выявления и анализа проблем.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Самарский государственный аэрокосмический университет
имени академика С.П. Королёва (национальный исследовательский
университет)» (СГАУ)

Факультет информатики
Кафедра программных систем

Сопченко Е.В.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к лабораторным работам по курсу
«ИНФОРМАТИКА»

для студентов, обучающихся:

по направлению 010300.62 «Фундаментальные
информатика и информационные технологии»

по направлению 230100.62 «Информатика
и вычислительная техника»

по специальности 090303.65 «Информационная
безопасность автоматизированных систем»

Самара 2012

СОДЕРЖАНИЕ

Лабораторная работа 1 – Структура программы на Паскале	3
Лабораторная работа 2 – Обработка одномерных массивов	6
Лабораторная работа 3 - Обработка двумерных массивов	9
Лабораторная работа 4 – Использование процедур и функций	13
Лабораторная работа 6 – Обработка числовых рядов	22
Лабораторная работа 7 – Сортировка массивов	23
Лабораторная работа 8 – Текстовые файлы	24
Лабораторная работа 9 – Файлы записей	31

Лабораторная работа 1 – Структура программы на Паскале

Алгоритм и его программная реализация тесно взаимосвязаны.

- **Программа** – упорядоченная последовательность инструкций компьютера (команд) для решения задачи.

- **Программное обеспечение** – совокупность программ обработки данных и необходимых для их эксплуатации документов.

- **Программирование** – теоретическая и практическая деятельность, связанная с созданием программ.

Программирование является собирательным понятием и может рассматриваться и как наука, и как искусство; на этом основан научно-практический подход к разработке программ.

Программа – результат интеллектуального труда, для которого характерно творчество, поэтому в любой программе присутствует индивидуальность ее разработчика, программа отражает определенную степень искусства программиста. Вместе с тем программирование предполагает и рутинные работы, которые могут и должны иметь строгий регламент выполнения и соответствовать стандартам.

Структура программы на языке Паскаль

```
Program <имя программы>;  
uses <имена модулей>; {Раздел использования модулей}  
type <имя типа> = <определение типа>; {Раздел описания собственных типов}  
var <переменная>: <имя типа>; {Раздел описания переменных}  
  
{Раздел описания подпрограмм - процедур и функций}  
procedure <имя процедуры [(список параметров)]>;  
  var <список переменных процедуры>;  
  begin  
    {Тело процедуры}  
  end;  
  
function <имя функции [(список параметров)]>: <тип результата>;  
  var <список переменных процедуры>;  
  begin  
    {Тело функции}  
  end;  
  
Begin {Основная программа}  
  {Тело программы}  
End.
```

Основными элементами программы на Паскале являются:

1) зарезервированные слова - это слова, с помощью которых паскаль определяет, какие действия ему необходимо выполнять.

2) идентификаторы - это название пользовательских констант, переменных, подпрограмм, модулей и меток.

Идентификатор должен состоять из латинских букв, знаков подчеркивания (`_`) и арабских цифр.

Идентификатор всегда должен начинаться с латинской буквы или знака подчеркивания.

Идентификаторы не могут совпадать с зарезервированными словами.

Большие и малые буквы в идентификаторе Паскаль не различает, поэтому нужно следить за отсутствием дублирования имен идентификаторов.

3) Константы - это постоянные величины, которые не меняют своего значения в течение работы программы.

4) Переменные - это величины, которые используются для ввода и вывода данных, для промежуточных и результирующих расчетов.

Любой переменной должно быть присвоено начальное значение.

5) Метки - особые места в программе, позволяющие вернуться к ним из любого места в программе.

Являются наследием низкоуровневого программирования.

Метки в программе запрещены!

6) подпрограммы - являются структурно обособленными элементами главной программы, начинаются со слов `procedure` и `function`.

описываются в программе один раз и могут быть вызваны произвольное количество раз.

7) комментарии -

необходимы для того, чтобы комментировать собственные действия.

Крупные структурные блоки в программе желательно снабжать комментариями.

{ <...> } - простой комментарий.

(* <...> *) - сложный комментарий, который может включать в себя несколько простых комментариев.

Операции в Паскале

{Операции указаны в порядке убывания приоритета}

1) унарные (`not`, `@` {операция взятия адреса}) - применяются к одному операнду ;

2) мультипликативные (`*` / and `div mod`) ;

3) аддитивные (`+` - or `xor`);

4) операции отношения (`=` `<>` `<` `>` `<=` `>=` `in` {для проверки имеется ли элемент во множестве});

Операторы ввода - это операторы чтения данных с клавиатуры.

`read`

`readln` {за счет `~ln` переводит курсор на следующую строку}

Одним оператором могут быть считаны такие типы, как string и все простые типы.

Если необходимо считать массив, то нужно считывать его каждый элемент. То есть оператор read должен выполняться в цикле.

Операторы вывода

write

writeln {за счет ~ln переводит курсор на следующую строку}

Для вещественных чисел предусмотрена особая форма записи оператора write(ln).

Пусть x имеет тип real. Тогда мы можем вывести его таким образом: write(x:8:2), где:

8- это общее число символов на число x, включая запятую;

2- это кол-во знаков после запятой.

Основные типы данных

1) целый тип integer (-32168 ... 32167)

чаще всего используется для описания целочисленных массивов, счетчиков, индексов элементов.

Занимает в памяти 2 байта.

2) целый тип byte (0 ... 255)

используется в качестве индексов элементов одномерных и многомерных массивов, а также в кач-ве счетчика.

Mem=1 байт, поэтому является более предпочтительным, чем тип integer в диапазоне от 0 до 255.

3) вещественный тип real

используется при вычислении встроенных в Паскаль функций, любых операций, содержащих оператор вещественного деления (/).

4) символьный char

занимает в памяти 1 байт, имеет ASCII-кодировку, позволяет закодировать 256 символов, включая EN и RUS буквы, цифры, знаки препинания, символы псевдографики и управляющие символы.

5) строковый string

представляет собой цепочку длиной не более 255 символов. Со строкой можно работать как с массивом символов, так и как с целым числом.

6) логический boolean

Mem=1 байт. Переменные этого типа могут принимать только 2 значения: true & false.

Наиболее часто используется в виде т.н. программных флагов, которые показывают программисту, наступило то или иное событие в программе или нет.

Лабораторная работа 2 – Обработка одномерных массивов

Язык программирования Паскаль
<p><u>Описание</u></p> <p><i>1 способ:</i> задается целочисленный массив фиксированной длины 10</p> <pre>var A: array [1..10] of integer; i:integer; {индекс элементов массива}</pre> <p><i>2 способ:</i> задается целочисленный массив фиксированной длины 10. В программе в качестве длины массива используется константа n.</p> <pre>const n=10; var A: array [1..n] of integer; i:integer; {индекс элементов массива}</pre> <p><i>3 способ:</i> задается целочисленный массив длиной не более 10 (nmax). В программе в качестве фактической длины массива используется переменная константа n.</p> <pre>const nmax=10; type vector = array [1..nmax] of integer; var A:vector; i:integer; {индекс элементов массива} n:integer; {фактическое количество элементов массива}</pre>
<p><u>Ввод</u></p> <pre>for i:=1 to n do readln(A[i]);</pre>
<p><u>Вывод</u></p> <pre>for i:=1 to n do writeln(A[i]);</pre>

Задание к лабораторной работе:

Составить алгоритм и написать программу:

1. Для одномерного целочисленного массива из 20 элементов решить задачу поиска максимального элемента и его номера.
2. Для одномерного целочисленного массива из 20 элементов решить задачу поиска суммы положительных элементов.
3. Для одномерного целочисленного массива из 20 элементов решить задачу поиска количества четных элементов.
4. Для одномерного целочисленного массива из 20 элементов решить задачу поиска количества элементов, больших своего предыдущего.
5. Для одномерного целочисленного массива из 20 элементов решить задачу поиска среднего арифметического положительных элементов.
6. Для одномерного целочисленного массива из 20 элементов решить задачу поиска суммы четных элементов.

7. Для одномерного целочисленного массива из 20 элементов решить задачу поиска среднего арифметического нечетных элементов.
8. Для одномерного целочисленного массива из 20 элементов решить задачу поиска количества ненулевых элементов.
9. Для одномерного целочисленного массива из 20 элементов решить задачу поиска количества элементов, неравных своему предыдущему.
10. Для одномерного целочисленного массива из 20 элементов решить задачу поиска номера первого отрицательного элемента.
11. Для одномерного целочисленного массива из 20 элементов решить задачу поиска суммы нечетных элементов.
12. Для одномерного целочисленного массива из 20 элементов решить задачу поиска номера последнего отрицательного элемента.
13. Для одномерного целочисленного массива из 20 элементов решить задачу поиска количества нулевых элементов.
14. Для одномерного целочисленного массива из 20 элементов решить задачу поиска среднего арифметического четных элементов.
15. Для одномерного целочисленного массива из 20 элементов решить задачу поиска количества элементов, меньших своего предыдущего.
16. Для одномерного целочисленного массива из 20 элементов решить задачу поиска номера первого положительного элемента.
17. Для одномерного целочисленного массива из 20 элементов решить задачу поиска суммы ненулевых элементов.
18. Для одномерного целочисленного массива из 20 элементов решить задачу поиска количества отрицательных элементов.
19. Для одномерного целочисленного массива из 20 элементов решить задачу поиска номера последнего положительного элемента.
20. Для одномерного целочисленного массива из 20 элементов решить задачу поиска среднего арифметического отрицательных элементов.
21. Для одномерного целочисленного массива из 20 элементов решить задачу поиска номера первого нечетного элемента.
22. Для одномерного целочисленного массива из 20 элементов решить задачу поиска количества элементов, равных своему предыдущему.
23. Для одномерного целочисленного массива из 20 элементов решить задачу поиска количества положительных элементов.
24. Для одномерного целочисленного массива из 20 элементов решить задачу поиска номера первого нулевого элемента.
25. Для одномерного целочисленного массива из 20 элементов решить задачу поиска номера последнего четного элемента.

26. Для одномерного целочисленного массива из 20 элементов решить задачу поиска среднего арифметического ненулевых элементов.
27. Для одномерного целочисленного массива из 20 элементов решить задачу поиска номера первого четного элемента.
28. Для одномерного целочисленного массива из 20 элементов решить задачу поиска количества нечетных элементов.
29. Для одномерного целочисленного массива из 20 элементов решить задачу поиска суммы отрицательных элементов.
30. Для одномерного целочисленного массива из 20 элементов решить задачу поиска минимального элемента и его номера.

Лабораторная работа 3 - Обработка двумерных массивов

Язык программирования Паскаль
<p><u>Описание</u></p> <p><i>1 способ:</i> задается целочисленная матрица размером 10 x 10 элементов.</p> <pre>var A: array [1..10, 1..10] of integer; i,j:integer; {индексы элементов массива}</pre>
<p><i>2 способ:</i> задается целочисленная матрица размером 10 x 10 элементов. В программе в качестве длины массива используется константа n.</p> <pre>const n=10; var A: array [1..n, 1..n] of integer; i,j:integer; {индексы элементов массива}</pre>
<p><i>3 способ:</i> задается целочисленная матрица размером не более 10 x 10 элементов (nmax x nmax). В программе в качестве фактической длины массива используется переменная константа n.</p> <pre>const nmax=10; var A: array [1..nmax] of integer; i,j:integer; {индексы элементов массива} n,m:integer; {фактическое количество строк и столбцов матрицы}</pre>
<p><u>Ввод</u></p> <pre>Readln(n,m); for i:=1 to n do for j:=1 to m do readln(A[i]);</pre>
<p><u>Вывод</u></p> <pre>for i:=1 to n do for j:=1 to m do writeln(A[i]);</pre>

Задания к лабораторной работе № 3:

1. Используя процедуры и функции из целочисленной матрицы A (n x m) получить вектор, состоящий из сумм элементов соответствующих строк матрицы.
2. Используя процедуры и функции из целочисленной матрицы A (n x m) получить вектор, элементами которого будут номера первых отрицательных элементов строк матрицы.
3. Используя процедуры и функции из целочисленной матрицы A (n x m) получить вектор, элементами которого будут TRUE, если в соответствующей строке есть хотя бы один отрицательный элемент, и FALSE, если иначе.
4. Используя процедуры и функции из целочисленной матрицы A (n x m) получить вектор, состоящий из номеров минимальных четных элементов строк матрицы.

5. Используя процедуры и функции из целочисленной матрицы A ($n \times m$) получить вектор, состоящий из сумм элементов, не превосходящих по значению заданное пользователем число в соответствующих строках матрицы.
6. Используя процедуры и функции из целочисленной матрицы A ($n \times m$) получить вектор, элементами которого будут номера последних отрицательных элементов строк матрицы.
7. Используя процедуры и функции из целочисленной матрицы A ($n \times m$) получить вектор, состоящий из номеров минимальных элементов соответствующих строк матрицы.
8. Используя процедуры и функции из целочисленной матрицы A ($n \times m$) получить вектор, состоящий из сумм элементов, превосходящих по значению заданное пользователем число в соответствующих строках матрицы.
9. Используя процедуры и функции из целочисленной матрицы A ($n \times m$) получить вектор, состоящий из номеров минимальных нечетных элементов строк матрицы.
10. Используя процедуры и функции из целочисленной матрицы A ($n \times m$) получить вектор, состоящий из сумм элементов соответствующих столбцов матрицы.
11. Используя процедуры и функции из целочисленной матрицы A ($n \times m$) получить вектор, элементами которого будут номера первых положительных элементов строк матрицы.
12. Используя процедуры и функции из целочисленной матрицы A ($n \times m$) получить вектор, элементами которого будет TRUE, если в строке матрицы количество нулевых элементов равно количеству ненулевых, и FALSE – иначе.
13. Используя процедуры и функции из целочисленной матрицы A ($n \times m$) получить вектор, состоящий из номеров максимальных четных элементов строк матрицы.
14. Используя процедуры и функции из целочисленной матрицы A ($n \times m$) получить вектор, состоящий из произведений элементов соответствующих строк матрицы.
15. Используя процедуры и функции из целочисленной матрицы A ($n \times m$) получить вектор, элементами которого будут TRUE, если в соответствующей строке есть хотя бы один положительный элемент, и FALSE, если иначе.

- 16.Используя процедуры и функции из целочисленной матрицы $A (n \times m)$ получить вектор, состоящий из максимальных элементов соответствующих строк матрицы.
- 17.Используя процедуры и функции из целочисленной матрицы $A (n \times m)$ получить вектор, элементами которого будут номера последних положительных элементов строк матрицы.
- 18.Используя процедуры и функции из целочисленной матрицы $A (n \times m)$ получить вектор, состоящий из номеров максимальных нечетных элементов строк матрицы.
- 19.Используя процедуры и функции из целочисленной матрицы $A (n \times m)$ получить вектор, состоящий из произведений элементов соответствующих столбцов матрицы.
- 20.Используя процедуры и функции из целочисленной матрицы $A (n \times m)$ получить вектор, состоящий из сумм положительных элементов соответствующих столбцов матрицы.
- 21.Используя процедуры и функции из целочисленной матрицы $A (n \times m)$ получить вектор, элементами которого будут TRUE, если в соответствующей строке есть хотя бы один нулевой элемент, и FALSE, если иначе.
- 22.Используя процедуры и функции из целочисленной матрицы $A (n \times m)$ получить вектор, элементами которого будут номера первых четных элементов соответствующих строк матрицы.
- 23.Используя процедуры и функции из целочисленной матрицы $A (n \times m)$ получить вектор, состоящий из номеров минимальных элементов соответствующих строк матрицы.
- 24.Используя процедуры и функции из целочисленной матрицы $A (n \times m)$ получить вектор, состоящий из сумм отрицательных элементов соответствующих строк матрицы.
- 25.Используя процедуры и функции из целочисленной матрицы $A (n \times m)$ получить вектор, элементами которого будет TRUE, если в строке матрицы количество четных (по значению) элементов равно количеству нечетных (нулевые не учитывать), и FALSE – иначе.
- 26.Используя процедуры и функции из целочисленной матрицы $A (n \times m)$ получить вектор, состоящий из средних арифметических элементов соответствующих строк матрицы.
- 27.Используя процедуры и функции из целочисленной матрицы $A (n \times m)$ получить вектор, элементами которого будут TRUE, если в

соответствующей строке есть хотя бы один ненулевой элемент, и FALSE, если иначе.

- 28.Используя процедуры и функции из целочисленной матрицы A ($n \times m$) получить вектор, состоящий из сумм отрицательных элементов соответствующих столбцов матрицы.
- 29.Используя процедуры и функции из целочисленной матрицы A ($n \times m$) получить вектор, состоящий из номеров максимальных элементов соответствующих строк матрицы.
- 30.Используя процедуры и функции из целочисленной матрицы A ($n \times m$) получить вектор, состоящий из номеров последних четных элементов соответствующих строк матрицы.

Лабораторная работа 4 – Использование процедур и функций

В языке Паскаль, как и в большинстве языков программирования, предусмотрены средства, позволяющие оформлять вспомогательный алгоритм как подпрограмму. Это бывает необходимо тогда, когда какой-либо алгоритм неоднократно повторяется в программе или имеется возможность использовать некоторые фрагменты уже разработанных ранее алгоритмов. Кроме того, подпрограммы применяются для разбиения крупных программ на отдельные смысловые части в соответствии с модульным принципом в программировании.

Для использования алгоритма в качестве подпрограммы ему необходимо присвоить имя и описать алгоритм по правилам языка Паскаль. В дальнейшем, при необходимости вызвать его в программе, делают вызов подпрограммы упоминанием в нужном месте имени соответствующего алгоритма со списком входных и выходных данных. Такое упоминание приводит к выполнению входящих в подпрограмму операторов, работающих с указанными данными. После выполнения подпрограммы работа продолжается с той команды, которая непосредственно следует за вызовом подпрограммы.

В языке Паскаль имеется два вида подпрограмм - **процедуры** и **функции**.

Описание и вызов процедур и функций

Структура описания процедур и функций до некоторой степени похожа на структуру Паскаль-программы: у них также имеются заголовок, раздел описаний и исполняемая часть. Раздел описаний содержит те же подразделы, что и раздел описаний программы: описания констант, типов, меток, процедур, функций, переменных. Исполняемая часть содержит собственно операторы процедур.

Одна и та же подпрограмма может вызываться неоднократно, выполняя одни и те же действия с разными наборами входных данных. Параметры, использующиеся при записи текста подпрограммы в разделе описаний, называют **формальными**, а те, что используются при ее вызове - **фактическими**.

Формат описания процедуры имеет вид:

```
procedure имя процедуры (формальные параметры) ;  
    раздел описаний процедуры  
begin  
    исполняемая часть процедуры  
end;
```

Формат описания функции:

```
function имя функции (формальные параметры) : тип  
результата;  
    раздел описаний функции  
begin  
    исполняемая часть функции  
  
end;
```

Формальные параметры в заголовке процедур и функций записываются в виде:

```
var имя параметра: имя типа
```

и отделяются друг от друга точкой с запятой. Ключевое слово `var` может отсутствовать (об этом далее). Если параметры однотипны, то их имена можно перечислять через запятую, указывая общее для них имя типа. При описании параметров можно использовать только стандартные имена типов, либо имена типов, определенные с помощью команды `type`. Список формальных параметров может отсутствовать.

Вызов процедуры производится оператором, имеющим следующий формат:

```
имя процедуры (список фактических параметров) ;
```

Список фактических параметров - это их перечисление через запятую. При вызове фактические параметры подставляются вместо формальных, стоящих на тех же местах в заголовке. Таким образом происходит передача входных параметров, затем выполняются операторы исполняемой части процедуры, после чего происходит возврат в вызывающий блок. Передача выходных параметров происходит непосредственно во время работы исполняемой части.

Вызов функции в Турбо Паскаль может производиться аналогичным способом, кроме того имеется возможность осуществить вызов внутри какого-либо выражения. В частности имя функции может стоять в правой части оператора присваивания, в разделе условий оператора `if` и т.д.

Для обмена информацией между процедурами и функциями и другими блоками программы существует механизм **входных** и **выходных параметров**. Входными параметрами называют величины, передающиеся из вызывающего блока в подпрограмму (исходные данные для подпрограммы), а выходными - передающиеся из подпрограммы в вызывающий блок (результаты работы подпрограммы).

Для передачи в вызывающий блок выходного значения функции в исполняемой части функции перед возвратом в вызывающий блок необходимо поместить следующую команду:

имя функции := результат;

При **вызове процедур и функций** необходимо соблюдать следующие **правила**:

- количество фактических параметров должно совпадать с количеством формальных;
- соответствующие фактические и формальные параметры должны совпадать по порядку следования и по типу.

Заметим, что имена формальных и фактических параметров могут совпадать. Это не приводит к проблемам, так как соответствующие им переменные все равно будут различны из-за того, что хранятся в разных областях памяти. Кроме того, все формальные параметры являются временными переменными - они создаются в момент вызова подпрограммы и уничтожаются в момент выхода из нее.

Передача параметров

В стандарте языка Паскаль передача параметров может производиться двумя способами - по значению и по ссылке. Параметры, передаваемые по значению, называют **параметрами-значениями**, передаваемые по ссылке - **параметрами-переменными**. Последние отличаются тем, что в заголовке процедуры (функции) перед ними ставится служебное слово `var`.

При первом способе (*передача по значению*) значения фактических параметров копируются в соответствующие формальные параметры. При изменении этих значений в ходе выполнения процедуры (функции) исходные данные (фактические параметры) измениться не могут. Поэтому таким способом передают данные только из вызывающего блока в подпрограмму (т.е. входные параметры). При этом в качестве фактических параметров можно использовать и константы, и переменные, и выражения.

При втором способе (*передача по ссылке*) все изменения, происходящие в теле процедуры (функции) с формальными параметрами,

приводят к немедленным аналогичным изменениям соответствующих им фактических параметров. Изменения происходят с переменными вызывающего блока, поэтому по ссылке передаются выходные параметры. При вызове соответствующие им фактические параметры могут быть только переменными.

Выбор способа передачи параметров при создании процедуры (функции) происходит в соответствии со сказанным выше: входные параметры нужно передавать по значению, а выходные - по ссылке. Практически это сводится к расстановке в заголовке процедуры (функции) описателя var при всех параметрах, которые обозначают результат работы подпрограммы. Однако, в связи с тем, что функция возвращает только один результат, в ее заголовке использовать параметры-переменные не рекомендуется.

Локальные и глобальные идентификаторы

Использование процедур и функций в Паскале тесно связано с некоторыми особенностями работы с идентификаторами (именами) в программе. В частности, не все имена всегда доступны для использования. Доступ к идентификатору в конкретный момент времени определяется тем, в каком блоке он описан.

Имена, описанные в заголовке или разделе описаний процедуры или функции называют **локальными** для этого блока. Имена, описанные в блоке, соответствующем всей программе, называют **глобальными**. Следует помнить, что формальные параметры процедур и функций всегда являются локальными переменными для соответствующих блоков.

Основные правила работы с глобальными и локальными именами:

- Локальные имена доступны (считаются известными, "видимыми") только внутри того блока, где они описаны. Сам этот блок, и все другие, вложенные в него, называют **областью видимости** для этих локальных имен.
- Имена, описанные в одном блоке, могут совпадать с именами из других, как содержащих данный блок, так и вложенных в него. Это объясняется тем, что переменные, описанные в разных блоках (даже если они имеют одинаковые имена), хранятся в разных областях оперативной памяти.

Глобальные имена хранятся в области памяти, называемой **сегментом данных** (статическим сегментом) программы. Они создаются на этапе компиляции и действительны на все время работы программы.

В отличие от них, локальные переменные хранятся в специальной области памяти, которая называется **стек**. Они являются временными, так как

создаются в момент входа в подпрограмму и уничтожаются при выходе из нее.

Имя, описанное в блоке, "закрывает" совпадающие с ним имена из блоков, содержащие данный. Это означает, что если в двух блоках, один из которых содержится внутри другого, есть переменные с одинаковыми именами, то после входа во вложенный блок работа будет идти с локальной для данного блока переменной. Переменная с тем же именем, описанная в объемлющем блоке, становится временно недоступной и это продолжается до момента выхода из вложенного блока.

Рекомендуется все имена, которые имеют в подпрограммах чисто внутреннее, вспомогательное назначение, делать локальными. Это предохраняет от изменений глобальные объекты с такими же именами.

Задание к лабораторной работе:

1. Переработать задание из лабораторной работы 2 с использованием процедур и функций.
2. Переработать задание из лабораторной работы 3 с использованием процедур и функций.

Лабораторная работа 5 – Обработка строковых и символьных переменных

Тип `STRING` (строка) в Турбо Паскале широко используется для обработки текстов. Он во многом похож на одномерный массив символов, описываемый директивой `array [1..255] of char`, однако, в отличие от последнего, количество символов в строке-переменной может меняться.

Длину строки можно ограничить любой константой порядкового типа (но не более 255), указав это в описании строки:

```
s:string[35];
```

Если ограничение не указано, то длина строки принимается максимально возможной, а именно 255 символов.

Нулевой байт строки хранит информацию о ее фактической длине.

Строка в Турбо Паскале трактуется как цепочка символов. К любому символу в строке можно обратиться точно так же, как к элементу одномерного массива `array [1..255] of char`, например:

```
Var s : String;  
begin  
  ...  
  if s[5] = 'A' then...  
end.
```

Над строками определены следующие функции:

1. `LENGTH (S)` – функция типа `INTEGER`; возвращает длину строки `S`.
2. `POS (Substr, S)` – функция типа `INTEGER`; отыскивает в строке `S` первое вхождение подстроки `Substr` и возвращает номер позиции, с которой она начинается; если подстрока не найдена, возвращается ноль.
3. `CONCAT(S1 [,S2, ... , Sn])` – функция типа `STRING`; возвращает строку, представляющую собой сцепление строк-параметров `S1, S2, ..., Sn`.

Аналогичный результат можно получить выполнив `S:=S1+S2+...+Sn`

4. `COPY(S, Index, Count)` – функция типа `STRING`; копирует из строки `S` `Count` символов, начиная с символа с номером `Index`.
5. `DELETE (S, Index, Count)` – процедура; удаляет `COUNT` символов из строки `S`, начиная с символа с номером `Index`.

6. INSERT (Substr, S, Index) – процедура; вставляет подстроку Substr в строку S, начиная с символа с номером Index.
7. STR(X [; Width [: Dec]], S) – процедура, которая преобразует число X любого вещественного или целого типов в строку символов S так, как это делает процедура WRITELN перед выводом; параметры Width и Dec, если они присутствуют, задают формат преобразования: Width определяет общую ширину поля, выделенного под соответствующее символьное представление вещественного или целого числа X, а Dec – количество символов в дробной части (этот параметр имеет смысл только в том случае, когда X- вещественное число).
8. VAL(S, X, Code) – процедура; преобразует строку символов S во внутреннее представление целой или вещественной переменной X, которое определяется типом этой переменной; параметр Code содержит ноль, если преобразование прошло успешно, и тогда в X помещается результат преобразований, в противном случае он содержит номер позиции в строке S, где обнаружен ошибочный символ, и в этом случае содержимое X не меняется; в строке S недопустимы незначащие пробелы;
9. UPCASE (Ch) – функция типа CHAR; возвращает для символьного выражения Ch, которое должно представлять собой строчную латинскую букву, соответствующую заглавную букву; если значением CH является любой другой символ (в том числе строчная буква русского алфавита), функция возвращает его без преобразования.

Пример: Удалить из строки все незначащие пробелы

```
var S : String; Len : Byte;
begin
  while Pos(' ', S)>0 do delete (S, Pos(' ', S),1);
  if S[1] = ' ' then delete (S, 1,1);
  if S[length(S)] = ' ' then delete (S, length(S),1);
end.
```

Операции отношения =, <>, >, <, >=, <= выполняются над двумя строками посимвольно, слева направо с учетом внутренней кодировки символов. Если одна строка меньше другой по длине, недостающие символы короткой строки заменяются значением CHR(0) .

Следующие операции отношения дадут значение TRUE:

```
'Turbo' <' Turbo Pascal'
'Паскаль' >'Turbo Pascal'
```

Задания к лабораторной работе № 5:

1. Дан массив A из N строк длиной M. Сформировать массив B, каждый элемент которого равен первому латинскому символу соответствующей строки.
2. Дан массив B из N строк длиной M. Сформировать массив V, каждый элемент которого равен последнему символу соответствующей строки.
3. Определить позицию первого и последнего вхождения в заданную строку A (N) какого-либо символа, введенного пользователем. Если строка не содержит данного символа, выдать сообщение.
4. Проверить, является ли данная строка из 6 символов "счастливой" (сумма первых трех цифр равна сумме трех последних). Выдать сообщение и суммы левой и правой половины.
5. Определить, является ли заданная строка идентификатором.
6. Определить, является ли заданная строка десятичной записью целого числа.
7. Определить, является ли заданная строка шестнадцатеричной записью целого числа.
8. Переписать заданную строку в обратном направлении.
9. Посчитать в строке количество слогов "на" и "при", предлогов "в".
10. Заменить в строке символов все слоги "на" слогом "при".
11. Посчитать в строке символов количество слов, написанных латинскими буквами. Слова в строке разделены пробелами, в конце текста точка.
12. Посчитать в строке символов и выдать на экран количество букв "а", "б", ..., "я".
13. Посчитать количество слов в строке, начинающихся на букву "в". Слова в строке разделены пробелами, в конце текста точка.
14. Посчитать количество слов в строке, заканчивающихся на букву "я". Слова в строке разделены пробелами, в конце текста точка.
15. Посчитать в строке символов и выдать на экран количество букв "а", "б", ..., "z".
16. Дан массив A из N строк длиной M. Сформировать массив B, каждый элемент которого равен символу с максимальным порядковым номером в таблице ASCII соответствующей строки.
17. Дан массив A из N строк длиной M. Сформировать массив B, каждый элемент которого равен символу с минимальным порядковым номером в таблице ASCII соответствующей строки.

18. Определить, является ли заданная строка двоичной записью целого числа.
19. Определить, является ли заданная строка восьмеричной записью целого числа.
20. Посчитать в строке символов и выдать на экран количество цифр "0", "1", ..., "9".
21. Проверить, является ли строка читаемой в прямом и обратном направлении (например, "а роза упала на лапу азора"), (не считая пробелов).
22. Посчитать, сколько и каких букв есть в заданной строке (например, "торт" - буква "т" -2, "о" -1, "р" -3).
23. Выдать на экран все символы и их порядковые номера из таблицы символов ASCII, находящиеся между двумя символами, введенными с клавиатуры.
24. Найти в строке символ "*". Разделить заданную строку на две строки (первая - до "*", вторая после "*").
25. Выдать на экран матрицу, состоящую из символа, введенного с клавиатуры (н-р, 2x3, символ 'М'). Размеры матрицы вводятся пользователем.
26. Найти в строке символ "!". Заменить все символы до него на "а", после него - на "я".
27. Посчитать в строке количество слов. Заменить каждое второе слово на "!!!!!"
28. Заменить в строке символов все слова на слово, введенное пользователем.
29. Посчитать в строке символов и выдать на экран количество всех символов (например, "а" - 5, "б" - 6, "f" - 2, "z"- 11, "?"- 3 и т.д.)
30. Посчитать количество слов в строке, начинающихся на слог "по". Слова в строке разделены пробелами, в конце текста точка.

Лабораторная работа 6 – Обработка числовых рядов

1. Реализуйте алгоритм вычисления суммы n первых членов ряда с заданной точностью.

$$\frac{1!}{x} + \frac{2!}{x^2} + \frac{3!}{x^3} + \frac{4!}{x^4} + \dots + \frac{n!}{x^n} + \dots$$

2. Реализуйте алгоритм вычисления суммы n первых членов ряда с заданной точностью.

$$1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^{n-1}}{(n-1)!} + \dots$$

3. Реализуйте алгоритм вычисления суммы n первых членов числового ряда с заданной точностью.

$$\sin x + \sin^2 x + \sin^3 x + \dots + \sin^n x + \dots$$

4. Реализуйте алгоритм вычисления суммы n первых членов числового ряда с заданной точностью.

$$1 + a x + a^2 x^2 + a^3 x^3 + \dots + a^n x^n + \dots$$

5. Реализуйте алгоритм вычисления последовательности с заданной точностью.

$$y_i, i = \overline{1, \dots, n} \equiv x + \sin(x + \sin(\dots(x + (\sin x))))$$

6. Реализуйте алгоритм вычисления суммы n первых членов числового ряда с заданной точностью.

$$1 + a^{-1} x + a^{-2} x^2 + a^{-3} x^3 + \dots + a^{-n} x^n + \dots$$

Лабораторная работа 7 – Сортировка массивов

1. Отсортировать столбцы двумерного массива чисел ($n \times m$) по возрастанию значений элементов в первой строке.
2. Отсортировать столбцы двумерного массива чисел ($n \times m$) по невозрастанию значений элементов в первой строке.
3. Отсортировать столбцы двумерного массива чисел ($n \times m$) по убыванию значений элементов в первой строке.
4. Отсортировать столбцы двумерного массива чисел ($n \times m$) по неубыванию значений элементов в первой строке.
5. Отсортировать строки двумерного массива чисел ($n \times m$) по возрастанию значений элементов в столбце с номером, заданным пользователем.
6. Отсортировать строки двумерного массива чисел ($n \times m$) по невозрастанию значений в столбце с номером, заданным пользователем.
7. Отсортировать строки двумерного массива чисел ($n \times m$) по убыванию значений элементов в столбце с номером, заданным пользователем.
8. Отсортировать строки двумерного массива чисел ($n \times m$) по неубыванию значений элементов в столбце с номером, заданным пользователем.
9. Отсортировать элементы каждой строки двумерного массива чисел ($n \times m$) по неубыванию значений элементов.
10. Отсортировать элементы каждой строки двумерного массива чисел ($n \times m$) по убыванию значений элементов.
11. Отсортировать элементы каждой строки двумерного массива чисел ($n \times m$) по возрастанию значений элементов.
12. Отсортировать элементы каждой строки двумерного массива чисел ($n \times m$) по невозрастанию значений элементов.
13. Отсортировать столбцы двумерного массива чисел ($n \times m$) по возрастанию значений элементов в строке с номером, заданным пользователем.
14. Отсортировать столбцы двумерного массива чисел ($n \times m$) по невозрастанию значений в строке с номером, заданным пользователем.
15. Отсортировать столбцы двумерного массива чисел ($n \times m$) по убыванию значений элементов в строке с номером, заданным пользователем.
16. Отсортировать столбцы двумерного массива чисел ($n \times m$) по неубыванию значений элементов в строке с номером, заданным пользователем.
17. Отсортировать элементы каждого столбца двумерного массива чисел ($n \times m$) по неубыванию значений элементов.
18. Отсортировать элементы каждого столбца двумерного массива чисел ($n \times m$) по убыванию значений элементов.
19. Отсортировать элементы каждого столбца двумерного массива чисел ($n \times m$) по возрастанию значений элементов.
20. Отсортировать элементы каждого столбца двумерного массива чисел ($n \times m$) по невозрастанию значений элементов.
21. Для каждой строки матрицы выдать сообщение о том, отсортирована ли строка по возрастанию, по убыванию или не отсортирована.
22. Для каждого столбца матрицы выдать сообщение о том, отсортирован ли столбец по возрастанию, по убыванию или не отсортирован.

Лабораторная работа 8 – Текстовые файлы

В программе:

- имя исходного и результирующего файла вводятся с клавиатуры. При вводе производится проверка на существование файла;
- использовать процедуры и функции с параметрами;
- файлы передавать в параметрах через VAR.

1. В исходном текстовом файле записаны строки, содержащие произвольные алфавитно-цифровые символы. Требуется написать программу, которая для каждой строки исходного файла будет выводить в результирующий файл последовательность строчных английских букв ("a", "b", ... "z") из входной последовательности и частоты их повторения. Печать должна происходить в алфавитном порядке. Например, пусть в одной из строк исходного файла содержатся следующие символы:

fhb5kbfыshfm

m4md

В этом случае в результирующем файле должно быть:

b-2, f -3, h-2, k-1, m-1, s-1

d-1, m-2

2. В исходном текстовом файле записаны строки, содержащие последовательность символов, среди которых встречаются и цифры. Требуется написать, которая для каждой строки исходного файла будет составлять и выводить в результирующий файл из тех цифр, которые не встречаются во входных данных, максимальное число. При составлении итогового числа каждая цифра может быть использована только один раз. Если во входных данных встречаются все цифры от 0 до 9, то следует вывести "-1". Например, пусть на вход подаются следующие символы:

Например, пусть в одной из строк исходного файла содержатся следующие символы:

173439

В этом случае в результирующем файле должно быть:

86520

3. В исходном текстовом файле записаны строки, содержащие строчные и прописные английские буквы. Требуется написать, которая для каждой строки исходного файла будет составлять и выводить в результирующий файл слово из тех букв английского алфавита, которые не встречаются во входных данных ни как строчные, ни как прописные, причем буквы должны идти в алфавитном порядке. Каждая буква должна быть распечатана один раз. Буквы построенного слова должны быть прописными. Если во входных данных встречаются все буквы английского алфавита, то следует вывести строчными буквами слово "no".

Например, пусть в одной из строк исходного файла содержатся следующие символы:

absCDKLMNOPvwXYabcprst.

В этом случае в результирующем файле должно быть:

EFGHIJQUZ

4. В исходном текстовом файле записаны строки, содержащие последовательность символов, среди которых встречаются и цифры. Требуется написать, которая для каждой строки исходного файла будет составлять и выводить в результирующий файл из тех цифр, которые встречаются во входных данных, максимальное число. При составлении итогового числа каждая цифра может быть использована только один раз. Если во входных данных цифры не встречаются, то следует вывести "-1".

Например, пусть в одной из строк исходного файла содержатся следующие символы:

14ф73п439

лапд

В этом случае в результирующем файле должно быть:

97431

-1

5. В исходном текстовом файле записаны строки, содержащие текст на английском языке. Требуется написать программу на языке Паскаль или Бейсик, которая для каждой строки исходного файла будет определять и выводить в результирующий файл английскую букву, встречающуюся в этой строке реже всего (но не нулевое количество), и количество там таких букв. Строчные и прописные буквы при этом считаются не различимыми. Если искомым букв несколько, то программа должна выводить на экран первую из них по алфавиту.

Например, пусть в одной из строк исходного файла содержатся следующие символы:

It is a task for you. Yes!

В этом случае в результирующем файле должно быть:

F 1

6. В исходном текстовом файле записаны строки, содержащие цифры в произвольном порядке, возможно разделенные другими символами. Требуется написать программу, которая для каждой строки исходного файла будет выводить в результирующий файл последовательность цифр ('0','1'..'9') из входной последовательности и, через пробел, частот их повторения. Печать должна происходить в порядке возрастания.

Например, пусть в одной из строк исходного файла содержатся следующие символы:

54533526.

В этом случае в результирующем файле должно быть:

2 1, 3 2, 4 1, 5 3, 6 1

7. В исходном текстовом файле записаны строки, содержащие последовательность символов, среди которых встречаются и цифры. Требуется написать, которая для каждой строки исходного файла будет составлять и выводить в результирующий файл из тех цифр, которые не встречаются во входных данных, минимальное число. При составлении итогового числа каждая цифра может быть использована только один раз (первый 0 не выводить). Если во входных данных встречаются все цифры от 0 до 9, то следует вывести "-1". Например, пусть на вход подаются следующие символы:

Например, пусть в одной из строк исходного файла содержатся следующие символы:

173439.

В этом случае в результирующем файле должно быть:

2568

8. В исходном текстовом файле записаны строки, содержащие строчные и прописные английские буквы. Требуется написать, которая для каждой строки исходного файла будет составлять и выводить в результирующий файл слово из тех букв английского алфавита, которые встречаются во входных данных либо как строчные, либо как прописные, причем буквы должны идти в алфавитном порядке. Каждая буква должна быть распечатана один раз. Буквы построенного слова должны быть прописными. Если во входных данных встречаются все буквы английского алфавита, то следует вывести строчными буквами слово "no".

Например, пусть в одной из строк исходного файла содержатся следующие символы:

absCDKLMNOPvwXYabcprst.

В этом случае в результирующем файле должно быть:

ABCDKLMNOPRSTVWXY

9. В исходном текстовом файле записаны строки, содержащие последовательность символов, среди которых встречаются и цифры. Требуется написать, которая для каждой строки исходного файла будет составлять и выводить в результирующий файл из тех цифр, которые встречаются во входных данных, максимальное число. При составлении итогового числа каждая цифра может быть использована только один раз. Если во входных данных цифры не встречаются, то следует вывести "-1".

Например, пусть в одной из строк исходного файла содержатся следующие символы:

14ф73п439

аеро

В этом случае в результирующем файле должно быть:

13479

-1

10. В исходном текстовом файле записаны строки, содержащие текст на английском языке. Требуется написать программу, которая для каждой строки исходного файла будет определять и выводить в результирующий файл английскую букву, встречающуюся в этой строке чаще всего, и количество там таких букв. Строчные и прописные буквы при этом считаются не различимыми. Если искомым букв несколько, то программа должна выводить на экран первую из них по алфавиту. Например, пусть в одной из строк исходного файла содержатся следующие символы:

It is not a simple task. Yes!

В этом случае в результирующем файле должно быть:

I 3

11. В исходном текстовом файле записаны строки, содержащие строчные и прописные английские буквы. Требуется написать, которая для каждой строки исходного файла будет печатать в результирующий файл в алфавитном порядке только те буквы, которые встретились во входной последовательности ровно 3 раза. Каждая буква при этом должна быть распечатана один раз. Буквы построенного слова должны быть прописными. Например, пусть в одной из строк исходного файла содержатся следующие символы:

btfgbbffjrtatbama

в результирующем файле должно быть:

aft

12. В исходном текстовом файле записаны строки, содержащие текст на английском языке. Требуется написать программу, которая будет проводить частотный анализ текста и последовательно выводить в результирующий файл только букву и через пробел символ * в количестве, равном количеству повторений этой буквы в тексте (в каждой строке результирующего файла информация об одной букве, другие символы не учитываются). Сведения о буквах, которые в тексте отсутствуют, на экран выводиться не должны. Сами буквы должны выводиться в алфавитном порядке.

Например, для текста:

It is science

в результирующем файле должно быть:

S **

E **

I ***

N *

S**

T *

13. В исходном текстовом файле записаны строки, содержащие цифры в произвольном порядке, возможно разделенные другими символами. Требуется написать программу, которая для каждой строки исходного файла будет выводить в результирующий файл последовательность цифр ('0','1'..'9') из входной последовательности в порядке увеличения частоты их встречаемости. Каждая цифра при этом должна быть распечатана один раз. Если какие-то цифры встречаются одинаковое число раз, то они выводятся по возрастанию.

Например, пусть в одной из строк исходного файла содержатся следующие символы:

123**24#32

в результирующем файле должно быть:

1432

14. В исходном текстовом файле записаны строки, содержащие текст на английском языке. Требуется написать программу, которая для каждой строки исходного файла будет определять и выводить в результирующий файл буквы, встречающиеся в этой строке в порядке уменьшения частоты их встречаемости. Строчные и прописные буквы при этом считаются не различимыми. Каждая буква, которая встречается в тексте, должна быть выведена ровно один раз.

Если какие-то буквы встречаются одинаковое количество раз, то они выводятся в алфавитном порядке.

Например, пусть в одной из строк исходного файла содержатся следующие символы:

zzzbbaattt

в результирующем файле должно быть:

tzab

15. В исходном текстовом файле записаны строки, содержащие текст на английском языке. Необходимо определить количество букв в самом длинном слове файла, обозначив полученное число **K** (словом называется непрерывная последовательность английских букв, слова друг от друга отделяются любыми другими символами, длина слова не превышает 20 символов). Затем необходимо переписать строки исходного файла в результирующий файл, заменив каждую английскую букву в строке на букву, стоящую в алфавите на **K** букв раньше (алфавит считается циклическим, то есть перед буквой **A** стоит буква **Z**), оставив другие символы неизменными. Строчные буквы при этом остаются строчными, а прописные - прописными.

Например, пусть в одной из строк исходного файла содержатся следующие символы:

Ce Ud Fd,Gde Ud

в результирующем файле должно быть:

Zb Ra Ca,Dab Ra

16. В исходном текстовом файле записаны строки, содержащие текст на английском языке. Необходимо определить количество букв в самом коротком слове файла, обозначив полученное число K (словом называется непрерывная последовательность английских букв, слова друг от друга отделяются любыми другими символами, длина слова не превышает 20 символов). Затем необходимо переписать строки исходного файла в результирующий файл, заменив каждую английскую букву в строке на букву, стоящую в алфавите на K букв позже (алфавит считается циклическим, то есть перед буквой A стоит буква Z), оставив другие символы неизменными. Строчные буквы при этом остаются строчными, а прописные - прописными. Например, пусть в одной из строк исходного файла содержатся следующие символы:

Zb Ra Ca Dab Ra

в результирующем файле должно быть:

Bd Tc Ec Fcd Tc

17. В исходном текстовом файле записаны строки, содержащие текст на английском языке. Требуется написать программу, которая для каждой строки исходного файла будет определять и выводить в результирующий файл буквы, встречающиеся в этой строке в порядке уменьшения частоты их встречаемости. Строчные и прописные буквы при этом считаются неразличимыми. Каждая буква, которая встречается в тексте, должна быть выведена ровно один раз.

Если какие-то буквы встречаются одинаковое количество раз, то они выводятся в алфавитном порядке.

Например, пусть в одной из строк исходного файла содержатся следующие символы:

baobaba.

в результирующем файле должно быть:

oab

18. В исходном текстовом файле записаны строки, содержащие строчные и прописные английские буквы. Требуется написать, которая для каждой строки исходного файла будет печатать в результирующий файл в алфавитном порядке только те буквы, которые встретились во входной последовательности ровно 3 раза подряд (друг за другом). Каждая буква при этом должна быть распечатана один раз. Буквы построенного слова должны быть прописными.

Например, пусть в одной из строк исходного файла содержатся следующие символы:

bbbtffbgbbffrtatbaffma.

в результирующем файле должно быть:

bf

19. В исходном текстовом файле записаны строки, содержащие текст на английском языке. Требуется написать программу, которая для каждой строки будет печатать в результирующий файл слова, являющиеся палиндромами (читающимися в прямом и обратном порядке). Слово, состоящее из одной буквы, также считается.

20. В исходном текстовом файле записаны строки, содержащие текст на английском языке. Требуется написать программу, которая для каждой строки будет печатать в результирующий файл слова, начинающиеся и заканчивающиеся одной и той же буквой. Слово, состоящее из одной буквы, также считается.

21. В исходном текстовом файле хранится информация о реках в виде: название, протяженность в километрах. Создать новый файл, в который поместить информацию о реках в порядке уменьшения их протяженности.

22. В исходном текстовом файле хранится информация в виде символьных строк произвольной длины. Подсчитать в каждой строке количество слов. В новый файл в каждую строку записать это количество и, через пробел, самое длинное слово из каждой строки исходного файла, заключенное в скобки.

23. Из двух заданных файлов, содержащих строки произвольной длины, сформировать новый файл, содержащий строки по N символов. Каждая строка результирующего файла содержит N1 символов из 1-го файла, N2 пробелов и остальные символы (N-N1-N2) из 2-го файл.

24. В исходном текстовом файле хранится информация в виде символьных строк произвольной длины. Провести частотный анализ текста: посчитать количество слов, начинающихся на различные буквы русского и английского алфавита. В новый файл в каждую строку записать букву и количество слов, начинающихся на эти буквы. В начало файла поместить информацию о русском алфавите, затем об английском.

25. В текстовом файле хранится информация о реках в виде: название, протяженность в километрах. Найти максимальную протяженность реки и дописать это число в конец файла. В другой текстовый файл переписать информацию о реках, в названии которых встречается фрагмент, указанный пользователем.

26. В каждой строке текстового файла хранится информация следующего вида: слово и число его повторений. В новый файл записать слова, повторив их столько раз, сколько указано.

Лабораторная работа 9 – Файлы записей

В лабораторной работе разработать меню, позволяющее:

- создавать новый файл, открывать, просматривать, редактировать и удалять существующий файл;
- добавлять, редактировать, удалять записи из файла (не используя массив записей);
- ввод имен файлов осуществляется с клавиатуры, необходимо производить проверку на существование файла;

ВНИМАНИЕ: почти в каждом задании 3 (!) файла: исходный (ФЗ), результирующий 1й (ФЗ), результирующий 2й (текстовый файл). Для результирующего текстового файла предусмотреть только процедуру просмотра (естественно, с вводом имени с клавиатуры и проверкой на существование).

Задания к лабораторной работе:

1. В файле хранится информация о книгах: шифр, фамилия автора, название книги, издательство, год издания. В новый файл переписать информацию о книгах, выпущенных заданным издательством. Если таковые имеются, то проверить, есть ли среди них книги 19nn года издания, если таких книг нет, то в текстовый файл вывести список авторов, фамилия которых начинается на заданную букву.
2. В файле хранится информация о книгах: шифр, фамилия автора, название книги, год издания. В новый файл переписать информацию о книгах, в названии которых встречается заданное слово. В текстовый файл выдать информацию о названиях книг исходного файла, год издания которых меньше заданного.
3. В файле хранится информация об учениках школы: имя, фамилия, дата рождения (число, месяц, год) и название класса (год обучения и буква). В новый файл переписать информацию об учениках 10 и 11-х классов, поместив вначале сведения о десятых (10А, 10Б, 10В и т. д.), а затем об одиннадцатых классах. Вывести в текстовый файл информацию, в каких классах более 20 учеников.
4. В файле хранится информация о пассажирах: фамилия пассажира, багаж пассажира: список из 3 вещей и вес каждой вещи. В новый файл переписать информацию о пассажирах, багаж которых по весу превышает заданный вес. Вывести в текстовый файл фамилию пассажира, имеющего багаж максимального веса, и общий вес его

багажа, а также фамилию пассажира, имеющего багаж минимального веса, и общий вес его багажа

5. В файле хранится информация о кредитных историях: фамилия заемщика, кредитная история: список не более чем из 3 кредитов, с указанием названия банка, суммы кредита, ежемесячного платежа и отметки об имеющейся просрочке. В новый файл переписать информацию о заемщиках, взявших кредит в сумме, превышающей размер среднего кредита (вычисляется по файлу). Вывести в текстовый файл фамилии заемщиков, у которых нет ни одной просрочки ни по одному платежу.
6. В файле содержатся сведения об автомобилях: марка, номер, фамилия владельца, величина пробега и даты последнего техосмотра (число, месяц, год). В новый файл переписать информацию о владельцах автомобилей, которые прошли техосмотр вовремя (техосмотр проводится один раз в году). В текстовый файл вывести марку, номер, фамилию владельца автомобилей, величина пробега которых превышает заданную величину.
7. В файле хранится информация о студентах: фамилия, имя, номер группы и отметки, полученные в последнюю сессию. Создать три новых файла, в которые поместить информацию о студентах, которые:
 - будут получать стипендию (получены хорошие и отличные оценки);
 - будут отчислены по результатам сессии (получены три и больше неудовлетворительные оценки);
 - кому будет назначен «исправительный срок» (одна или две неудовлетворительные оценки).
8. В файле хранится информация о студентах: фамилия, имя, номер группы, количество пропусков занятий по неуважительной причине. Создать три новых файла, в которые поместить информацию о студентах:
 - вызвать в деканат (количество пропусков от K_1 до K_2);
 - объявить выговор (количество пропусков от K_2 до K_3);
 - отчислить (количество пропусков больше K_3).
9. В файле содержатся сведения о кубиках: размер кубика (длина ребра), цвет (красный, желтый, зеленый или синий), материал (деревянный, металлический, картонный). Выдать в текстовый файл:
 - количество кубиков каждого цвета и их суммарный объем;
 - количество деревянных кубиков с ребром 3 см;
 - количество металлических кубиков с ребром больше 5 см.

10. В файле содержатся сведения о веществах: название вещества, удельный вес, проводимость (проводник, полупроводник, диэлектрик). Создать новый файл, содержащий информацию о всех веществах, удельный вес которых не превышает заданного. Выдать в текстовый файл удельные веса и названия всех полупроводников.
11. В файле содержатся сведения о промышленных товарах: номер отдела, название товара, стоимость, размер (если это необходимо). В новый файл переписать информацию о товарах из указанного отдела; выдать в текстовый файл список товаров, стоимость которых превышает заданную цену.
12. В файле содержатся сведения о веществах: название вещества, удельный вес и проводимость (проводник, полупроводник, диэлектрик). В новый файл переписать данные, упорядочив их по убыванию удельных весов. Выдать в текстовый файл названия всех диэлектриков, удельные веса которых превышают заданный.
13. В файле содержатся сведения об автомобилях: марка, номер, фамилия владельца, величина пробега и даты последнего техосмотра (число, месяц, год). В новый файл переписать информацию о владельцах автомобилей, величина пробега которых не превышает заданную величину. В текстовый файл вывести номера и фамилии владельцев автомобилей заданной марки.
14. В файле хранится информация о книгах: шифр, фамилия автора, название книги, издательство, год издания. В новый файл переписать информацию о книгах, написанных заданным автором. Если таковые имеются, то проверить, есть ли среди них книги n11 года издания, если таких книг нет, то в текстовый файл вывести список книг, в названии которых есть заданное слово.
15. В файле содержатся сведения о производителях товаров: ИНН, название организации, телефон, фамилия директора. В новый файл переписать информацию о производителях из указанного региона (код региона – 2 первых цифры ИНН); выдать в текстовый файл список производителей и фамилии директоров организаций, телефоны которых оканчиваются на 2 заданные цифры.
16. В файле хранится информация об учениках школы: имя, фамилия, класс (номер и буква), рост. В новый файл переписать информацию о 3х самых высоких учениках в каждой параллели. Вывести в текстовый файл фамилии и рост учеников, чей рост больше среднего по школе.
17. В файле хранится информация о пассажирах: фамилия пассажира, багаж пассажира: список из 3 вещей и вес каждой вещи. В новый файл переписать информацию о пассажирах, багаж которых по весу

превышает средний вес багажа по всему файлу. Вывести в текстовый файл фамилии пассажиров, фамилии которых начинаются на заданную букву.

18. В файле хранится информация о книгах: шифр, фамилия автора, название книги, издательство, год издания. В новый файл переписать информацию о книгах, написанных заданным автором. Если таковые имеются, то проверить, есть ли среди них книги 20nn года издания, если таких книг нет, то в текстовый файл вывести список авторов, фамилия которых начинается на заданную букву.
19. В файле хранится информация о пассажирах: фамилия пассажира, багаж пассажира: список из 3 вещей и вес каждой вещи. В новый файл переписать информацию о пассажирах, багаж которых по весу не превышает средний вес багажа по всему файлу. Вывести в текстовый файл фамилии пассажиров, фамилии которых начинаются на заданную букву.
20. В файле хранится информация о кредитных историях: фамилия, имя, отчество заемщика, кредитная история: список не более чем из 3 кредитов, с указанием названия банка, суммы кредита, ежемесячного платежа и отметки об имеющейся просрочке. В новый файл переписать информацию о заемщиках, взявших кредит в заданном банке. Вывести в текстовый файл фамилии заемщиков, у которых имеется просрочка хотя бы по одному платежу.
21. В файле хранится информация об учениках школы: имя, фамилия, класс (номер и буква) и список олимпиад (не более 5), в которых участвовал каждый ученик: предмет, количество баллов, место, занятое на олимпиаде. В новый файл переписать информацию об учениках, участвовавших в олимпиаде по заданному предмету. Вывести в текстовый файл фамилии учеников, чей средний балл по олимпиадам больше среднего по школе.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Самарский государственный аэрокосмический университет
имени академика С.П. Королёва (национальный исследовательский
университет)» (СГАУ)

Факультет информатики
Кафедра программных систем

Сопченко Е.В.

Задания на практические работы
«ИНФОРМАТИКА»

для студентов, обучающихся:

по направлению 010300.62 «Фундаментальные
информатика и информационные технологии»

по направлению 230100.62 «Информатика
и вычислительная техника»

по специальности 090303.65 «Информационная
безопасность автоматизированных систем»

Самара 2012

СОДЕРЖАНИЕ

Занятие 1 – Системы счисления, используемые в ПК	3
Занятие 2 - Вычисление количества информации.....	4
Занятие 3 – Преобразование логических выражений	7
Занятие 4 - Построение алгоритмических структур	8
Занятие 5 - Одномерные и двумерные массивы. Сортировка.....	9
Занятие 6 - Символьный и строковый тип. Обработка.....	12
Занятие 7 - Динамические структуры данных. Стек. Очередь. Односвязные списки.....	13
Занятие 8 - Динамические структуры данных. Стек. Очередь. Двусвязные списки	16
Занятие 9 - Рекурсия. Деревья.....	19

Занятие 2 - Вычисление количества информации

Выберите правильный ответ

- 1** Шифровальщик использует смешанную кодировку: азбуку Морзе (точки и тире) и +/*, это позволяет ему кодировать символы, задавая различные последовательности комбинаций указанных символов [. - + *]. Сколько различных символов (цифр, букв, знаков пунктуации и т.д.) можно закодировать, используя данный код длиной четыре символа?
1) 120 2) 512 3) 256 4) 128
- 2** Сколько килобайт информации содержит сообщение объемом 2^{20} байт?
1) 1000 2) 2 3) 1 4) 1024
- 3** В игральной колоде 72 карты 4 мастей. Какое количество бит (минимально возможное) потребуется для кодирования всех карт одной масти?
1) 6 2) 5 3) 4 4) 3
- 4** Сколько существует различных последовательностей, которые кодируются азбукой Морзе (символы «точка», «тире»), длиной от шести до семи символов?
1) 64 2) 128 3) 192 4) 224
- 5** Костюм танцора состоит из четырех предметов. Каждый предмет может быть одного из трех цветов. Сколько всего участников в абсолютно разных костюмах могут принять участие в конкурсе?
1) 12 2) 16 3) 81 4) 7
- 6** Сколько различных последовательностей символов можно закодировать, используя символы «+», «-», «*», «/» длиной не менее двух и не более трех символов?
1) 24 2) 80 3) 128 4) 160
- 7** Метрологическая станция ведет наблюдение за сейсмической активностью в регионе, которая измеряется целыми числами от 0 до 12 баллов, записываемых минимально возможным количеством бит. Станция фиксирует активность каждый час. Оцените в байтах информационный объем хранимой информации за год (в году 365 дней)
1) 8760 2) 2190 3) 365 4) 4380
- 8** Для передачи секретного сообщения используются буквы латинского алфавита «A» до «Z». При этом все буквы кодируются одним и тем же минимально возможным количеством бит. Определите информационный объем сообщения (включая точку): It is not a simple task.
1) 15 бит 2) 24 бита 3) 15 байт 4) 24 байта
- 9** Поле для игры в реверси представляет собой доску 12x12 клеток. Сколько бит потребуется для кодирования координат одного поля?
1) 24 2) 7 3) 8 4) 9
- 10** Наибольшее натуральное число, кодируемое 8 битами?
1) 127 2) 255 3) 256 4) 512
- 11** Число байт, необходимое для записи числа 394?
1) 2 2) 1 3) 394 4) 197
- 12** Число байт, необходимое для записи числа 207?
1) 109 2) 1 3) 2 4) 3
- 13** Число байт, необходимое для записи числа 912?
1) 6 2) 3 3) 2 4) 1
- 14** Наибольшее натуральное число, кодируемое 16 битами?
1) 256 2) 255 3) 65535 4) 32767
- 15** Наибольшее целое число, кодируемое 16 битами?
1) 256 2) 255 3) 65535 4) 32767

Вычислите количество информации

1. Какой объем информации содержит сообщение, уменьшающее неопределенность знаний в 4 раза.
2. Группа школьников пришла в бассейн, в котором 4 дорожки для плавания. Тренер сообщил, что группа будет плавать на дорожке № 3. Сколько информации получили школьники из этого сообщения?
3. При угадывании целого числа в диапазоне от 1 до N было получено 8 бит информации. Чему равно N ?
4. Какое количество информации несет сообщение, что встреча назначена на 15 число?
5. В корзине лежат 8 шаров. Все шары одного цвета. Сколько информации несет сообщение о том, что из корзины достали красный шар?
6. Сообщение о том, что ваш друг живет на 10 этаже несет 4 бита информации. Сколько этажей в доме?
7. В коробке лежат 7 разноцветных карандашей. Какое количество информации содержит сообщение, что из коробки достали красный карандаш?
8. «Вы выходите на следующей остановке?» - спросили человека в автобусе. «Нет», - ответил он. Сколько информации содержит ответ?
9. Сообщение о том, что Петя живет во втором подъезде, несет 3 бита информации. Сколько подъездов в доме?
10. Какое количество информации несет сообщение: «Встреча назначена на сентябрь».
11. Для хранения области экрана монитора размером 512 x 256 точек выделено 64 Кбайт оперативной памяти. Вычислить максимально возможное количество цветов в данном случае.
12. Область экрана имеет размеры 512 x 128 точек. Каждая точка может иметь 1 из 256 оттенков. Найти минимальный объем памяти (в килобайтах) для хранения этой области экрана.
13. Сколько секунд потребуется модему, передающему сообщения со скоростью 28800 бит/с, чтобы передать цветное растровое изображение размером 640x480 пикселей, при условии, что цвет каждого пикселя кодируется тремя байтами?
14. Для хранения растрового изображения размером 128x128 пикселей отвели 4 килобайта памяти. Каково максимально возможное число цветов в палитре изображения?
15. Растровый графический файл содержит изображение с 16 градациями цвета, размером 100 x 100 точек. Найти объем этого файла в байтах.

16. Видео формируется за счет быстрой смены кадров – частота 25 Гц (25 кадров в секунду). Рассчитать объем (в Мбайт) 32-минутного видеоролика при разрешении 800 x 640, глубине цвета – 16 бит.
17. Для хранения области экрана монитора размером 256x128 точек выделено 32 Кбайт оперативной памяти. Сколько цветов максимально допустимо использовать для раскраски точек?
18. Для хранения области экрана монитора размером 512x256 точек выделено 64 Кбайта оперативной памяти. Какова глубина цвета этого изображения?
19. Емкость одного условного печатного листа равна приблизительно 32Кбайт (1 символ занимает 8 бит), скорость печати - 64 символа в секунду. Сколько минут потребуется без учета смены бумаги для распечатки текста одной газеты (2 усл. п.л.) на принтере (ответ округлить до целого числа)?
20. Емкость одного условного печатного листа равна приблизительно 96 Кбайтам, при этом символы закодированы в ASCII-коде. Сколько потребуется минут для распечатки текста одной газеты (4 усл. п.л.) на лазерном принтере (скорость печати - 512 символов в секунду) без учета смены бумаги (ответ округлите до целого числа)?
21. Объем информации о клиентах, хранимой в одном отделении банка занимает 45 мегабайт. Каждая карта клиента имеет объем 12 страниц (48 строк по 64 символа в каждой, 1 символ занимает 8 бит). Сколько карт клиентов хранится в отделении банка?
22. Вариант теста по информатике в среднем имеет объем 20 килобайт (на каждой странице теста 40 строк по 64 символа в каждой, 1 символ занимает 8 бит), Сколько страниц в тесте?
23. Сведения о сотруднике хранятся в виде строки из 2048 символов, кодируемых в Unicode. Сколько памяти (в Мбайт) требуется для хранения сведений обо всех 8192 сотрудниках учреждения?
24. Какое максимальное количество страниц книги (32 строки по 64 символа, 1 символ занимает 8 бит), которое поместится в файле объемом 640 Кбайт?
25. Определить количество уровней звукового сигнала при использовании устаревших 8-битных звуковых карт.
26. Известно, что видеопамять компьютера имеет объем 512 Кбайт. Разрешающая способность экрана 640 на 200. Сколько страниц экрана одновременно разместится в видеопамети при палитре 256 цветов?
27. Какой информационный объем имеет моноаудиофайл, длительность звучания которого 1 секунда, при среднем качестве звука (16 бит, 24 кГц)?
28. Определить объем стереоаудиофайла длительностью 20 секунд при 20-битном кодировании и частоте дискретизации 44.1 кГц.

Занятие 3 – Преобразование логических выражений

1. Определите с помощью преобразований, какие из следующих формул являются тождественно истинными или тождественно ложными. Проверьте с помощью таблиц истинности:

1. 1. $\overline{a \cdot a \vee b \cdot (a \cdot b \vee b)}$ (В ответе укажите: истина / ложь / не является и приведите таблицу истинности)

1. 2. $\left((a \vee \bar{b}) \rightarrow b \right) \cdot (\bar{a} \vee b)$ (В ответе укажите: истина / ложь / не является и приведите таблицу истинности)

2. Упростите формулы, используя законы алгебры логики

2. 1. $a \cdot b \cdot c \vee \bar{a} \cdot b \cdot c$

2. 2. $a \cdot b \cdot c \vee a \cdot \bar{b} \cdot c$

2. 3. $(\bar{a} \vee \bar{b} \vee \bar{c}) \cdot (a \vee b \vee c)$

2. 4. $a \cdot (a \vee b) \cdot (a \vee c)$

2. 5. $a \cdot b \cdot (a \cdot c \vee a \cdot b)$

2. 6. $a \cdot \bar{c} \vee c \cdot (b \vee \bar{c}) \vee (a \vee \bar{b}) \cdot c$

2. 7. $\overline{a \cdot (b \vee \bar{c}) \vee a \cdot b}$

2. 8. $(\bar{a} \vee c) \cdot \bar{a} \cdot c \cdot (b \vee \bar{c}) \cdot \bar{b} \cdot c$

2. 9. $\overline{x + y} \cdot (x + \bar{y})$

2. 10. $\bar{x} \cdot y + \overline{x + y} + x$

2. 11. $(x + y) \cdot (\bar{x} + y) \cdot (\bar{x} + \bar{y})$

2. 12. $x \cdot \bar{y} + \bar{x} \cdot y \cdot z + x \cdot z$

2. 13. $\overline{x \cdot y + z}$

2. 14. $x \cdot y + x \cdot y \cdot z + x \cdot z \cdot p$

2. 15. $\overline{x + y \cdot z + \overline{x + y + z}}$

2. 16. $x \cdot \bar{y} + x \cdot y \cdot z + x \cdot \bar{y} \cdot z + x \cdot y \cdot \bar{z}$

2. 17. $(x \cdot \bar{y} + z) \cdot (\bar{x} + y) + \bar{z}$

2. 18. $x \cdot y \cdot (\overline{x \cdot z + x \cdot y \cdot z + z \cdot t})$

2. 19. $A \vee \neg(A \& B)$

2. 20. $A \& (A \vee B) \& (C \vee \neg B)$

2. 21. $A \& \neg B \vee \neg B \& C \vee \neg A \& \neg B$

2. 22. $(A \vee \neg A) \& B$

Занятие 4 - Построение алгоритмических структур

Построить алгоритм для следующих задач:

1. Найти минимум и максимум двух данных чисел без использования массивов и циклов.
2. Найти минимум и максимум трех данных чисел без использования массивов и циклов.
3. Найти минимум и максимум четырех данных чисел без использования массивов и циклов.
4. Найти все корни заданного квадратного уравнения.
5. Найти наибольший общий делитель двух натуральных чисел (алгоритм Евклида).
6. Перевод натурального числа из десятичной системы счисления в позиционную систему с основанием, меньшим или равным 10. Обработка и преобразование такой записи числа.
7. Нахождение сумм, произведений элементов данной конечной числовой последовательности.
8. Поиск наименьшего простого делителя данного натурального числа.
9. Проверка числа на простоту.
10. Найти все простые числа в диапазоне от 1 до N .

Занятие 5 - Одномерные и двумерные массивы.

Сортировка

Одномерные массивы

Соседние элементы	
1.	Опишите на одном из языков программирования алгоритм подсчета количества элементов целочисленного массива из 30 элементов, которые больше своего предыдущего элемента этого массива.
2.	Опишите на одном из языков программирования алгоритм подсчета количества элементов целочисленного массива из 30 элементов, которые меньше своего предыдущего элемента этого массива.
Поиск номера элемента	
3.	Опишите на русском языке или одном из языков программирования алгоритм поиска номера первого из двух последовательных элементов в целочисленном массиве из 30 элементов, сумма которых максимальна (если таких пар несколько, то можно выбрать любую из них).
4.	Опишите на одном из языков программирования алгоритм поиска номера элемента в целочисленном массиве из 30 элементов, приращение которого при переходе к следующему элементу (т.е. разность между следующим и текущим) максимально. Если таких элементов несколько, то можно выбрать любой из них.
5.	Опишите алгоритм поиска трех последовательных элементов, сумма которых максимальна, в числовом массиве из 30 элементов.
Максимум (минимум) и его количество	
6.	Опишите на русском языке или одном из языков программирования алгоритм подсчета числа элементов, равных максимальному, в числовом массиве из 30 элементов.
7.	Опишите на русском языке или одном из языков программирования алгоритм подсчета числа элементов, равных минимальному, в числовом массиве из 30 элементов (за один проход массива).
8.	Опишите на одном из языков программирования алгоритм поиска второго по величине (т.е. следующего по величине за максимальным) элемента в целочисленном массиве из 30 элементов.
9.	Опишите на одном из языков программирования алгоритм поиска третьего по величине (т.е. следующего по величине за двумя самыми максимальными) элемента в целочисленном массиве из 30 элементов.
10.	Опишите на одном из языков программирования алгоритм вычисления максимальной разности двух последовательных значений элементов (из предыдущего вычитается последующий) в целочисленном массиве из 30 элементов.
Два массива	
11.	Опишите на русском языке или одном из языков программирования алгоритм получения из заданного целочисленного массива размером 30 элементов другого массива, который будет содержать модули значений элементов первого массива (не используя специальной функции, вычисляющей модуль числа).
12.	Дано два целочисленных массива A и B из 30 элементов каждый. Опишите на одном из языков программирования алгоритм, который записывает в каждый элемент массива B с индексом k произведение всех элементов массива A с индексами, не превосходящими k.
Вычисления	
13.	Опишите на одном из языков программирования алгоритм вычисления разности между средним арифметическим максимального и минимального значений элементов заданного целочисленного массива из 30 элементов и средним арифметическим всех элементов этого массива.
14.	Опишите на одном из языков программирования алгоритм подсчета среднего значения четных элементов в заданном целочисленном массиве из 30 элементов, проверяя, что в нем есть хотя бы один четный элемент.
15.	Опишите на одном из языков программирования алгоритм вычисления разности максимального среди элементов, имеющих четные значения, и максимальным среди элементов, имеющих нечетные значения, в заданном целочисленном массиве из 30 элементов (в предположении, что в массиве есть и четные и нечетные элементы).

Последовательности элементов	
16.	Опишите на одном из языков программирования алгоритм подсчета максимальной длины последовательности непрерывно возрастающих значений элементов в целочисленном массиве из 30 элементов.
17.	Опишите на одном из языков программирования алгоритм подсчета максимального количества подряд идущих совпадающих элементов в целочисленном массиве длины 30.
18.	Опишите на одном из языков программирования алгоритм подсчета максимальной длины последовательности непрерывно убывающих значений элементов в целочисленном массиве из 30 элементов.
19.	Опишите на одном из языков программирования алгоритм подсчета максимального количества подряд идущих положительных элементов в заданном целочисленном массиве из 30 элементов.
20.	Опишите на одном из языков программирования алгоритм подсчета суммы элементов, расположенных между первым и последним отрицательными элементами

Двумерные массивы

1.	Дан целочисленный квадратный массив 10 x 10. Опишите на одном из языков программирования алгоритм вычисления суммы максимальных элементов из каждой строки. Напечатать значение этой суммы. Предполагается, что в каждой строке такой элемент единственный.
2.	Дан целочисленный квадратный массив 10 x 10. Опишите на одном из языков программирования алгоритм поиска строки с наименьшей суммой элементов. Вывести на печать номер строки и сумму ее элементов. Предполагается, что такая строка единственная.
3.	Дан целочисленный квадратный массив 10 x 20. Опишите на одном из языков программирования алгоритм поиска строки с наименьшей суммой элементов. Вывести на печать номер строки и сумму ее элементов. Предполагается, что такая строка единственная.
4.	Дан целочисленный квадратный массив 10 x 10. Опишите на одном из языков программирования алгоритм вычисления суммы элементов строки, в которой расположен элемент с максимальным значением. Вывести на печать значение суммы строки. Предполагается, что такой элемент единственный.
5.	Дан целочисленный квадратный массив 6 x 10. Опишите на одном из языков программирования алгоритм вычисления суммы минимальных элементов из каждой строки. Напечатать значение этой суммы. Предполагается, что в каждой строке такой элемент единственный.
6.	Опишите на одном из языков программирования алгоритм, отбирающий из заданного целочисленного массива размером 30 элементов только положительные значения и печатающий их на стандартном устройстве вывода, а также подсчитывающий количество таких элементов.
7.	Опишите на одном из языков программирования алгоритм, отбирающий из заданного целочисленного массива размером 30 элементов только ненулевые значения и печатающий их на стандартном устройстве вывода, а также подсчитывающий количество таких элементов.
8.	Опишите на русском языке или одном из языков программирования алгоритм поиска номера первого из двух последовательных элементов в целочисленном массиве из 30 элементов, сумма которых минимальна (если таких пар несколько, то можно выбрать любую из них).
9.	Опишите на русском языке или одном из языков программирования алгоритм подсчета количества элементов, равных минимальному, в числовом массиве из 30 элементов.

Сортировка

1.	Дан целочисленный массив длиной не более 30 элементов. Отсортировать массив по возрастанию с использованием метода прямого обмена.
2.	Дан целочисленный массив длиной не более 30 элементов. Отсортировать массив по убыванию с использованием метода прямого обмена.
3.	Дан целочисленный массив длиной не более 30 элементов. Отсортировать массив по возрастанию с использованием метода выбора.
4.	Дан целочисленный массив длиной не более 30 элементов. Отсортировать массив по убыванию с использованием метода выбора.
5.	Дан целочисленный массив длиной не более 30 элементов. Отсортировать массив по возрастанию вставкой.
6.	Дан целочисленный массив длиной не более 30 элементов. Отсортировать массив по убыванию вставки.
7.	Дан целочисленный массив длиной не более 30 элементов. Отсортировать массив по возрастанию с использованием метода Шелла.
8.	Дан целочисленный массив длиной не более 30 элементов. Отсортировать массив по убыванию с использованием метода Шелла.
9.	Дан целочисленный массив длиной не более 30 элементов. Отсортировать массив по возрастанию с использованием быстрой сортировки.
10.	Дан целочисленный массив длиной не более 30 элементов. Отсортировать массив по убыванию с использованием быстрой сортировки.

Занятие 6 - Символьный и строковый тип. Обработка

1. Дана строка, заканчивающаяся точкой. Подсчитать, сколько в ней слов.
2. Дана строка, содержащая английский текст. Найти количество слов, начинающихся с буквы z.
3. Дана строка. Подсчитать в ней количество вхождений букв a, s, q.
4. Дана строка. Определить, сколько в ней символов знаков препинания.
5. Дана строка. Преобразовать ее, удалив каждый символ ! и повторив дважды каждый символ, отличный от !.
6. Дана строка. Определить количество вхождений фрагмента, заданного пользователем.
7. Составьте программу, выясняющую, есть ли в строке S буква 'A'.
8. Подсчитать количество букв 'a' и 'A' в строке S.
9. Составьте программу, удваивающую каждую букву слова S.
10. Составьте программу обращения строки S с удалением незначащих пробелов.
11. Дана строка. Указать те слова, которые содержат хотя бы одну букву w.
12. Дана строка. Найти в ней те слова, которые начинаются и оканчиваются одной и той же буквой.
13. В строке заменить все двоеточия (:) тире (-). Подсчитать количество замен.
14. В строке удалить символ двоеточие (:) и подсчитать количество удаленных символов.
15. В строке между словами вставить вместо пробела запятую и пробел.
16. Определить, сколько раз в строке встречается заданное слово.
17. Подсчитать количество символов до * и после нее.
18. Дана строка из p символов. Преобразовать ее, заменив все двоеточия (:), встречающиеся среди первых p/2 символов, и заменив точками все восклицательные знаки, встречающиеся среди символов, стоящих после p/2 символов.

Занятие 7 - Динамические структуры данных. Стек. Очередь. Односвязные списки¹

1. Исключите из линейного однонаправленного списка элементы, начиная с номера N до номера K, если оба элемента с такими номерами присутствуют в списке.
2. Исключите из линейного однонаправленного нециклического списка все элементы между двумя элементами с заданными значениями информационных полей, если оба таких элемента присутствуют в списке (повторяющихся информационных полей в списке нет).
3. Элементы линейного однонаправленного нециклического списка хранят слова, состоящие из 10 символов, включающих только английские прописные и строчные буквы. Исключите из списка все элементы, хранящие слова, начинающиеся с заданной комбинации символов (длина комбинации соответственно не более 10 символов).
4. На основе исходного линейного однонаправленного нециклического списка получите два новых списка путем копирования информационных полей его элементов: в первый список необходимо скопировать элементы, у которых значение информационного поля больше заданного значения; во второй список – все остальные элементы.
5. Даны два однонаправленных нециклических линейных списка. Постройте третий список путем копирования информационных полей элементов двух заданных списков в следующем порядке: 1-й элемент первого списка, 1-й элемент второго списка; 2-й элемент первого списка, 2-й элемент второго списка и т.д. Списки могут быть разной длины.
6. Даны два однонаправленных нециклических линейных списка. Исключите из первого списка элементы, значения информационных полей которых совпадают со значениями информационных полей элементов второго списка. Списки могут быть разной длины. Повторяющихся элементов в каждом из списков нет. Например,
1-й список: 10, 5, 20, 100, 7, 1;
2-й список: 10, 11, 7, 28;
1-й список после выполнения программы: 5, 20, 100, 1.
7. Дан однонаправленный нециклический линейный список. Постройте новый список путем копирования в него информационного поля каждого K-го по счету элемента первого списка, если его значение не превышает N.
8. Даны два однонаправленных нециклических линейных списка. Постройте новый список, элементы которого содержат информационные поля, значения которых входят одновременно в оба заданных списка. Списки могут быть разной длины. Повторяющихся элементов в каждом из списков нет. Например,
1-й список: 10, 5, 20, 100, 7, 1;
2-й список: 10, 11, 7, 1, 28;
3-й список: 10, 7, 1.

¹ Симонова Е.В. Курс лекций по динамическим структурам данных [Текст]. Самара: 2000, СГАУ, 112 с.

9. Дан линейный однонаправленный нециклический список. Информационные поля элементов могут повторяться. Вставьте в список новые элементы за каждым элементом с заданным значением информационного поля.
10. Даны два линейных однонаправленных нециклических списка. Исключите из первого списка элементы, значения информационных полей которых не совпадают со значениями информационных полей элементов второго списка. Списки могут быть разной длины. Повторяющихся элементов в каждом из списков нет. Например,
 1-й список: 10, 5, 20, 100, 7, 1;
 2-й список: 10, 11, 7, 28;
 1-й список после выполнения программы: 10, 7.
11. Дан линейный однонаправленный нециклический список. Информационные поля элементов могут повторяться. Вставьте в список новые элементы перед каждым элементом с заданным значением информационного поля.
12. Дан линейный однонаправленный нециклический список. Постройте новый список путем копирования в него информационного поля каждого элемента первого списка, если его значение превышает N.
13. Создайте линейный однонаправленный нециклический список. Из каждой группы следующих подряд элементов с одинаковыми значениями информационных полей сохраните только один – первый встретившийся, остальные элементы исключите. Например,
 исходный список: 1 1 1 2 2 3 1 1 1 1 4 5 6 6 6
 результирующий список: 1 2 3 1 4 5 6.
14. Создайте линейный однонаправленный нециклический список. Исключите из этого списка все элементы с совпадающими значениями информационных полей, кроме первого встретившегося элемента. Например,
 исходный список: 1 2 1 2 2 3 2 1 1 4 3
 результирующий список: 1 2 3 4.
15. Даны два однонаправленных нециклических линейных списка. Постройте новый список, элементы которого содержат информационные поля, значения которых входят в первый список, но не входят во второй. Списки могут быть разной длины. Повторяющихся элементов в каждом из списков нет. Например,
 1-й список: 10, 5, 20, 100, 7, 1;
 2-й список: 10, 11, 7, 1, 28;
 3-й список: 5, 20, 100.
16. Создайте линейный однонаправленный нециклический список, узлы которого могут содержать положительные и отрицательные числа. Исключите из этого списка все элементы с отрицательными значениями информационных полей и постройте из них новый список.
17. Однонаправленный нециклический список упорядочен по неубыванию значений информационных полей элементов, например, 1, 1, 2, 5, 5, 5, 10, 10, 20, 30, 30. Первоначально список пуст. Определите следующие действия:
- включить новый элемент в список так, чтобы не нарушилось упорядочение элементов,

- исключить из списка элемент (элементы) по заданному значению информационного поля так, чтобы не нарушилось упорядочение элементов.

18. Создайте два линейных однонаправленных нециклических списка. В первом списке за первым встретившимся элементом с заданным значением информационного поля включите все элементы второго списка, не меняя их расположение в памяти. Например,

до выполнения программы

1-й список: 2 1 2 3 2 2

2-й список: 5 6;

после выполнения программы

1-й список: 2 5 6 1 2 3 2 2

2-й список: пуст.

19. Постройте однонаправленный циклический линейный список, вставляя положительные числа непосредственно в начало списка, а отрицательные – в хвост списка. Постройте два однонаправленных нециклических линейных списка, копируя в один из них положительные значения информационных полей элементов циклического списка, а в другой – отрицательные значения информационных полей элементов циклического списка.

20. Очередь – тип линейного списка, в котором включение/исключение элементов выполняется в соответствии с правилом «первым пришел – первым вышел». Длина очереди ограничена. Определите следующие действия:

- включить новый элемент в очередь,
- исключить элемент из очереди,
- проверить очередь на отсутствие элементов,
- проверить очередь на переполнение.

Реализуйте очередь с использованием линейного однонаправленного нециклического списка.

21. Постройте копию линейного однонаправленного нециклического списка, изменив порядок составляющих его элементов на обратный.

22.* Постройте однонаправленный нециклический линейный список, вставляя положительные числа непосредственно в начало списка, а отрицательные – в хвост списка. Разделите полученный список на два списка, содержащие положительные и отрицательные элементы соответственно, не меняя расположения элементов в памяти.

23.* Исключите из линейного однонаправленного нециклического списка все элементы между двумя элементами с заданными номерами, если оба таких элемента присутствуют в списке (повторяющихся информационных полей в списке нет) и построьте из них новый список, не изменяя размещения элементов в памяти.

Занятие 8 - Динамические структуры данных. Стек. Очередь. Двусвязные списки²

1. Исключите из двусвязного циклического списка элементы, начиная с номера N до номера K, если оба элемента с такими номерами присутствуют в списке.
2. Исключите из двусвязного циклического списка все элементы между двумя элементами с заданными значениями информационных полей, если оба таких элемента присутствуют в списке (повторяющихся информационных полей в списке нет).
3. Элементы двусвязного циклического списка хранят слова, состоящие из 10 символов, включающих только английские прописные и строчные буквы. Исключите из списка все элементы, хранящие слова, начинающиеся с заданной комбинации символов (длина комбинации соответственно не более 10 символов).
4. На основе исходного двусвязного циклического списка получите два новых списка путем копирования информационных полей его элементов: в первый список необходимо скопировать элементы, у которых значение информационного поля больше заданного значения; во второй список – все остальные элементы.
5. Даны два двусвязных циклических списка. Постройте третий список путем копирования информационных полей элементов двух заданных списков в следующем порядке: 1-й элемент первого списка, 1-й элемент второго списка; 2-й элемент первого списка, 2-й элемент второго списка и т.д. Списки могут быть разной длины.
6. Даны два двусвязных циклических списка. Исключите из первого списка элементы, значения информационных полей которых совпадают со значениями информационных полей элементов второго списка. Списки могут быть разной длины. Повторяющихся элементов в каждом из списков нет. Например,
1-й список: 10, 5, 20, 100, 7, 1;
2-й список: 10, 11, 7, 28;
1-й список после выполнения программы: 5, 20, 100, 1.
7. Дан двусвязный циклический список. Постройте новый список путем копирования в него информационного поля каждого K-го по счету элемента первого списка, если его значение не превышает N.
8. Даны два двусвязных циклических списка. Постройте новый список, элементы которого содержат информационные поля, значения которых входят одновременно в оба заданных списка. Списки могут быть разной длины. Повторяющихся элементов в каждом из списков нет. Например,
1-й список: 10, 5, 20, 100, 7, 1;
2-й список: 10, 11, 7, 1, 28;
3-й список: 10, 7, 1.

² Симонова Е.В. Курс лекций по динамическим структурам данных [Текст]. Самара: 2000, СГАУ, 112 с.

9. Дан двусвязный циклический список. Информационные поля элементов могут повторяться. Вставьте в список новые элементы за каждым элементом с заданным значением информационного поля.
10. Даны два двусвязных циклических списка. Исключите из первого списка элементы, значения информационных полей которых не совпадают со значениями информационных полей элементов второго списка. Списки могут быть разной длины. Повторяющихся элементов в каждом из списков нет. Например,
 1-й список: 10, 5, 20, 100, 7, 1;
 2-й список: 10, 11, 7, 28;
 1-й список после выполнения программы: 10, 7.
11. Дан двусвязный циклический список. Информационные поля элементов могут повторяться. Вставьте в список новые элементы перед каждым элементом с заданным значением информационного поля.
12. Дан двусвязный циклический список. Постройте новый список путем копирования в него информационного поля каждого элемента первого списка, если его значение превышает N.
13. Создайте двусвязный циклический список. Из каждой группы следующих подряд элементов с одинаковыми значениями информационных полей сохраните только один – первый встретившийся, остальные элементы исключите. Например,
 исходный список: 1 1 1 2 2 3 1 1 1 4 5 6 6 6
 результирующий список: 1 2 3 1 4 5 6.
14. Создайте двусвязный циклический список. Исключите из этого списка все элементы с совпадающими значениями информационных полей, кроме первого встретившегося элемента. Например,
 исходный список: 1 2 1 2 2 3 2 1 1 4 3
 результирующий список: 1 2 3 4.
15. Даны два двусвязных циклических списка. Постройте новый список, элементы которого содержат информационные поля, значения которых входят в первый список, но не входят во второй. Списки могут быть разной длины. Повторяющихся элементов в каждом из списков нет. Например,
 1-й список: 10, 5, 20, 100, 7, 1;
 2-й список: 10, 11, 7, 1, 28;
 3-й список: 5, 20, 100.
16. Создайте двусвязный циклический список, узлы которого могут содержать положительные и отрицательные числа. Исключите из этого списка все элементы с отрицательными значениями информационных полей и постройте из них новый список.
17. Двусвязный циклический список упорядочен по неубыванию значений информационных полей элементов, например, 1, 1, 2, 5, 5, 5, 10, 10, 20, 30, 30. Первоначально список пуст. Определите следующие действия:
- включить новый элемент в список так, чтобы не нарушилось упорядочение элементов,

- исключить из списка элемент (элементы) по заданному значению информационного поля так, чтобы не нарушилось упорядочение элементов.

18.Создайте два двусвязных циклических списка. В первом списке за первым встретившимся элементом с заданным значением информационного поля включите все элементы второго списка, не меняя их расположение в памяти. Например,

до выполнения программы

1-й список: 2 1 2 3 2 2

2-й список: 5 6;

после выполнения программы

1-й список: 2 5 6 1 2 3 2 2

2-й список: пуст.

19.Очередь – тип линейного списка, в котором включение/исключение элементов выполняется в соответствии с правилом «первым пришел – первым вышел». Длина очереди ограничена. Определите следующие действия:

- включить новый элемент в очередь,
- исключить элемент из очереди,
- проверить очередь на отсутствие элементов,
- проверить очередь на переполнение.

Реализуйте очередь с использованием двусвязного циклического списка.

Занятие 9 - Рекурсия. Деревья³

1. Создайте сбалансированное бинарное дерево. Вычислите среднее арифметическое значений информационных полей узлов дерева.
2. Создайте сбалансированное бинарное дерево. Подсчитайте количество узлов дерева с положительными и отрицательными значениями информационных полей.
3. Создайте сбалансированное дерево. Подсчитайте количество нетерминальных узлов с заданным значением информационного поля и скопируйте эти значения в однонаправленный линейный список.
4. Создайте дихотомическое дерево произвольного вида. Подсчитайте количество терминальных узлов дерева.
5. По заданному массиву положительных чисел (числа используются для заполнения ключевых полей узлов дерева) постройте дихотомическое дерево и вычислите количество информационных полей, равных заданному значению.
6. Создайте сбалансированное бинарное дерево. Подсчитайте количество узлов дерева с заданным значением информационного поля.
7. Создайте дихотомическое дерево. Поместите в двусвязный циклический список значения информационных полей узлов дихотомического дерева в порядке возрастания значений ключей этих узлов.
8. Создайте сбалансированное бинарное дерево. Найдите разницу между максимальным и минимальным значениями информационных полей узлов дерева.
9. Создайте сбалансированное дерево. Подсчитайте количество нетерминальных узлов с заданным значением информационного поля и скопируйте эти значения в однонаправленный линейный список.
10. Создайте сбалансированное дерево произвольного вида. Подсчитайте сумму значений информационных полей элементов дерева. После этого измените всем узлам значения информационных полей на противоположные (по знаку).
11. Создайте дихотомическое дерево. Найдите в дереве узлы с заданными значениями ключей и поместите значения информационных полей этих узлов в массив.
12. Создайте сбалансированное бинарное дерево. Поместите в массив значения информационных полей узлов этого дерева, которые находятся в заданном диапазоне значений.
13. Создайте дихотомическое дерево. Найдите в дереве узлы с заданными значениями ключей и скопируйте значения информационных полей этих узлов в однонаправленный линейный список.
14. Создайте дихотомическое дерево. Замените все отрицательные значения информационных полей узлов дерева на их абсолютные величины, посчитайте среднее арифметическое положительных элементов.

³ Симонова Е.В. Курс лекций по динамическим структурам данных [Текст]. Самара: 2000, СГАУ, 112 с.

15. По заданному массиву целых чисел (числа используются для заполнения информационных полей узлов дерева) постройте сбалансированное дерево так, чтобы порядок следования чисел при нисходящем обходе дерева совпадал с порядком следования чисел в массиве.
16. Создайте дихотомическое дерево. Поместите в односвязный линейный список значения информационных полей узлов дихотомического дерева в порядке убывания значений ключей этих узлов.
17. Создайте дихотомическое дерево произвольного вида. Определите среднее арифметическое значений информационных полей элементов дерева, находящихся в заданном диапазоне.
18. Постройте бинарное дерево произвольного вида. Подсчитайте количество узлов, находящихся на заданном уровне дерева.
19. Создайте бинарное дерево произвольного вида. Подсчитайте количество узлов дерева с положительными и отрицательными значениями информационных полей.
20. Создайте сбалансированное бинарное дерево. Замените все отрицательные значения информационных полей узлов дерева на их абсолютные величины.
21. Создайте бинарное дерево произвольного вида. Подсчитайте количество терминальных узлов дерева с заданным значением информационного поля.
22. Создайте дихотомическое дерево. Определите глубину дерева.
23. Создайте дихотомическое дерево. Постройте копию этого дерева.
24. Создайте бинарное дерево произвольного вида. Постройте копию дерева произвольного вида.
25. По заданному массиву целых чисел (числа используются для заполнения информационных полей узлов дерева) постройте сбалансированное дерево так, чтобы порядок следования чисел при нисходящем обходе дерева был обратным по отношению к порядку следования чисел в массиве.
26. Создайте бинарное дерево произвольного вида. Поменяйте местами значения максимального и минимального элементов дерева.
27. Создайте бинарное дерево произвольного вида. Скопируйте в односвязный линейный список значения информационных полей узлов дерева в заданном диапазоне.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Самарский государственный аэрокосмический университет
имени академика С.П. Королёва (национальный исследовательский
университет)» (СГАУ)

Факультет информатики
Кафедра программных систем

Сопченко Е.В.

Тесты для промежуточного контроля знаний
«ИНФОРМАТИКА»

для студентов, обучающихся:

по направлению 010300.62 «Фундаментальные
информатика и информационные технологии»

по направлению 230100.62 «Информатика
и вычислительная техника»

по специальности 090303.65 «Информационная
безопасность автоматизированных систем»

Самара 2012

СОДЕРЖАНИЕ

Тест по теме 1	3
Тест по теме 2	4
Тест по теме 3	5
Тест по теме 4	6
Тест по теме 5	7
Тест по теме 6	8
Тест по теме 7	9
Тест по теме 8	10
Тест по теме 9	11
Тест по теме 10	12
Тест по теме 11	13
Тест по теме 12	14
Тест по теме 13	15
Тест по теме 14	16
Тест по теме 15	17
Тест по теме 16	18
Тест по теме 17	19
Тест по теме 18	20

Тест по теме 1

1. Информация присутствует во всех материальных системах как органической, так и неорганической природы и является следствием свойства отражения, согласно ... подходу к определению информации

- 1) философскому
- 2) функционально-методологическому
- 3) философско-методологическому
- 4) кибернетическому

2. Информация является свойством общества, живых существ и кибернетических устройств, согласно ... подходу к определению информации.

- 1) философскому
- 2) функционально-методологическому
- 3) философско-методологическому
- 4) кибернетическому

3. Эта информационная революция связана с изобретением письменности.

- 1) первая
- 2) вторая
- 3) третья
- 4) пятая

4. Эта информационная революция обусловлена открытием электричества, благодаря которому появились телеграф, телефон, радио, позволяющие оперативно передавать информацию в любом объеме.

- 1) первая
- 2) вторая
- 3) третья
- 4) пятая

Тест по теме 2

1. Наука о знаках семиотика **НЕ** включает в себя:

- 1) синтаксис
- 2) пунктуацию
- 3) семантику
- 4) прагматику

2. Знаки, которые не имеют прямой зависимости между формой и значением, называются

- 1) символы
- 2) диокритики
- 3) дистракторы
- 4) указатели

3. Совокупность сведений, которыми располагает пользователь или система – это ...

- 1) запись
- 2) база данных
- 3) тезаурус
- 4) атрибут

4. Эта мера определяет полезность (ценность) информации для достижения пользователем поставленной цели.

- 1) синтаксическая
- 2) прагматическая
- 3) семантическая
- 4) финансовая

Тест по теме 3

1. Показатель качества информации, который отражает правильность отбора и формирования информации...
 - 1) доступность
 - 2) точность
 - 3) репрезентативность
 - 4) достоверность

2. Показатель качества информации, который отражает степень сокращения ценности информации ...
 - 1) актуальность
 - 2) точность
 - 3) устойчивость
 - 4) достоверность

3. Показатель качества информации, который отражает степень близости получаемой информации реальному состоянию объекта ...
 - 1) актуальность
 - 2) точность
 - 3) устойчивость
 - 4) достоверность

4. Показатель качества информации, который отражает способность реагировать на изменение исходных данных без нарушения необходимой точности ...
 - 1) актуальность
 - 2) точность
 - 3) устойчивость
 - 4) достоверность

Тест по теме 4

1. Кодирование информации это информационный процесс ...
 - 1) ввода-вывода информации
 - 2) хранения информации
 - 3) передачи информации
 - 4) обработки информации

2. Знак, физический процесс или явление, несущие сообщение о каком-либо событии, состоянии объекта либо передающие команды управления, оповещения и т. д. – это...
 - 1) символ
 - 2) код
 - 3) сигнал
 - 4) кодовая таблица

3. Совокупность средств, методов и персонала, используемых для приема, хранения, обработки и передачи информации, т.е. для организации информационных процессов называется ...
 - 1) информационным процессом
 - 2) информационной системой
 - 3) аппаратной частью
 - 4) информационным обеспечением

4. Обратная связь необходима для...
 - 1) ввода-вывода информации
 - 2) хранения информации
 - 3) передачи информации
 - 4) корректировки входной информации

Тест по теме 5

1. Для параллельной передачи всех разрядов числового кода операнда используется
 - 1) шина адреса
 - 2) шина данных
 - 3) шина управления
 - 4) шина питания

2. Для параллельной передачи всех разрядов числового кода адреса ячейки оперативной памяти или порта ввода-вывода внешнего устройства используется ...
 - 1) шина адреса
 - 2) шина данных
 - 3) шина управления
 - 4) шина питания

3. Для передачи управляющих сигналов во все блоки ПК используется ...
 - 1) шина данных
 - 2) шина адреса
 - 3) шина управления
 - 4) шина питания

5. Для подключения блоков ПК к системе электропитания используется ...
 - 4) шина питания
 - 2) шина адреса
 - 3) шина управления
 - 4) шина данных

Тест по теме 6

1. Как представлено число 25_{10} в двоичной системе счисления?

- 1) 1001_2
- 2) 11001_2
- 3) 10011_2
- 4) 11010_2

2. Вычислите значение суммы $10_2 + 10_8 + 10_{16}$ в двоичной системе счисления.

- 1) 10100010_2
- 2) 11110_2
- 3) 11010_2
- 4) 10100_2

3. Как представлено число 83_{10} в двоичной системе счисления?

- 1) 1001011_2
- 2) 1100101_2
- 3) 1010011_2
- 4) 101001_2

4. Вычислите сумму двоичных чисел x и y , если $x = 1010101_2$ $y = 1010011_2$

- 1) 10100010_2
- 2) 10101000_2
- 3) 10100100_2
- 4) 10111000_2

Тест по теме 7

1. Производится одноканальная (моно) звукозапись с частотой дискретизации 16 кГц и 24-битным разрешением. Запись длится 1 минуту, ее результаты записываются в файл, сжатие данных не производится. Какая из приведенных ниже величин наиболее близка к размеру полученного файла?

- 1) 0.2 Мбайт
- 2) 2 Мбайт
- 3) 3 Мбайт
- 4) 4 Мбайт

2. Для кодирования цвета фона Интернет-страницы используется атрибут `bgcolor="#XXXXXX"`, где в кавычках задаются шестнадцатеричные значения интенсивности цветовых компонент в 24-битной RGB-модели. К какому цвету будет близок цвет страницы, заданный тэгом `<body bgcolor="#747474">`?

- 1) серый
- 2) белый
- 3) фиолетовый
- 4) черный

3. Для хранения растрового изображения размером 128×128 пикселей отвели 4 килобайта памяти. Каково максимально возможное число цветов в палитре изображения?

- 1) 8
- 2) 2
- 3) 16
- 4) 4

4. Укажите минимальный объем памяти (в килобайтах), достаточный для хранения любого растрового изображения размером 64×64 пикселя, если известно, что в изображении используется палитра из 256 цветов. Саму палитру хранить не нужно.

- 1) 128
- 2) 2
- 3) 256
- 4) 4

Тест по теме 8

1. Группа школьников пришла в бассейн, в котором 4 дорожки для плавания. Тренер сообщил, что группа будет плавать на дорожке № 3. Сколько бит информации получили школьники из этого сообщения?

- 1) 1
- 2) 2
- 3) 3
- 4) 4

2. Сообщение о том, что ваш друг живет на 10 этаже несет 4 бита информации. Сколько этажей в доме?

- 1) 16
- 2) 4
- 3) 10
- 4) 40

3. В коробке лежат 7 разноцветных карандашей. Какое количество информации содержит сообщение, что из коробки достали красный карандаш?

- 1) 56 бит
- 2) 3 бита
- 3) 7 бит
- 4) 3 байта

4. Сообщение о том, что Иван живет во втором подъезде, несет 3 бита информации. Сколько подъездов в доме?

- 1) 24
- 2) 3
- 3) 8
- 4) 6

Тест по теме 9

1. Какое логическое выражение равносильно выражению $\neg (A \wedge B) \wedge \neg C$?

- 1) $\neg A \vee B \vee \neg C$
- 2) $(\neg A \vee \neg B) \wedge \neg C$
- 3) $(\neg A \vee \neg B) \wedge C$
- 4) $\neg A \wedge \neg B \wedge \neg C$

2. Для какого из указанных значений числа X истинно высказывание

$((X < 5) \rightarrow (X < 3)) \wedge ((X < 2) \rightarrow (X < 1))$

- 1) 1
- 2) 2
- 3) 3
- 4) 4

3. Укажите, какое логическое выражение равносильно выражению $A \wedge \neg (\neg B \vee C)$.

- 1) $\neg A \vee \neg B \vee \neg C$
- 2) $A \wedge \neg B \wedge \neg C$
- 3) $A \wedge B \wedge \neg C$
- 4) $A \wedge \neg B \wedge C$

4. Какое из приведенных имен удовлетворяет логическому условию

$\neg (\text{первая буква гласная} \rightarrow \text{вторая буква гласная}) \wedge \text{последняя буква гласная}$

- 1) ИРИНА
- 2) МАКСИМ
- 3) АРТЕМ
- 4) МАРИЯ

Тест по теме 10

1. Точная формулировка решения задачи на компьютере с описанием входной и выходной информации рассматриваемой предметной области – это ...
 - 1) постановка задачи
 - 2) предметная область
 - 3) модель данных
 - 4) входная информация

2. Совокупность связанных между собой функций, задач управления, с помощью которых достигается выполнение поставленных целей – это ...
 - 1) постановка задачи
 - 2) предметная область
 - 3) модель данных
 - 4) входная информация

3. Используемая знаковая система для обозначения структур данных и операций их обработки называется ...
 - 1) постановкой задачи
 - 2) предметной областью
 - 3) моделью данных
 - 4) входной информацией

4. Данные, поступающие на вход задачи и используемые для ее решения, это ...
 - 1) входная информация
 - 2) предметная область
 - 3) модель данных
 - 4) постановка задачи

Тест по теме 11

1. Применимость алгоритма для решения определенного класса задач – это ...
 - 1) дискретность
 - 2) детерминированность
 - 3) результативность
 - 4) массовость

2. Обязательное получение желаемого результата за конечное число шагов при допустимых исходных данных это ...
 - 1) дискретность
 - 2) детерминированность
 - 3) результативность
 - 4) массовость

3. Разбиение процесса обработки информации на более простые этапы (шаги), выполнение которых компьютером или человеком не вызывает затруднений, это ...
 - 1) дискретность
 - 2) детерминированность
 - 3) результативность
 - 4) массовость

4. Однозначность получаемого результата при одних и тех же исходных данных, это ...
 - 1) дискретность
 - 2) детерминированность
 - 3) результативность
 - 4) массовость

Тест по теме 12

1. Что **НЕ** является показателем качества программного продукта?

- 1) мобильность
- 2) надежность
- 3) достоверность
- 4) эффективность

2. Что не является показателем качества программного продукта?

- 1) дискретность
- 2) надежность
- 3) мобильность
- 4) эффективность

3. Что не является показателем качества программного продукта?

- 1) коммуникативность
- 2) надежность
- 3) мобильность
- 4) массовость

4. Что не является показателем качества программного продукта?

- 1) мобильность
- 2) надежность
- 3) актуальность
- 4) эффективность

Тест по теме 13

1. Совокупность программ, позволяющая организовать решение задач пользователя на компьютере – это ...

- 1) операционная система
- 2) среда программирования
- 3) программное обеспечение
- 4) антивирус

2. Программы, управляющие оперативной памятью, процессором, внешними устройствами и обеспечивающие возможность работы других программ, называют

- 1) утилитами
- 2) драйверами
- 3) операционными системами
- 4) системами программирования

3. Ниже перечислено 8 различных программных средств. Какие из них являются операционными системами? (Выберите ответ, в котором перечислены только операционные системы)

- | | | | |
|-------------------|---------------------------|----------------------|-----------------------|
| A) Acrobat Reader | B) IBM PC DOS | Д) Microsoft Office | Ж) Norton SystemWorks |
| Б) ASP Linux | Г) Macromedia Dreamweaver | Е) Microsoft Windows | З) RealOne Player |

- 1) АВЕЗ
- 2) БВЕ
- 3) ВДЕ
- 4) ЕЖ

4. Совокупность программных средств, обеспечивающая управление аппаратной частью компьютера и прикладными программами, а также их взаимодействие между собой и пользователем.

- 1) операционная система
- 2) среда программирования
- 3) программное обеспечение
- 4) антивирус

Тест по теме 14

1. Значения двумерного массива размера 7×7 задаются с помощью вложенного оператора цикла в представленном фрагменте программы

```
for n:=1 to 7 do  
  for k:=1 to 7 do  
    V[n, k]:=k-n;
```

Сколько элементов массива будут иметь положительные значения?

- 1) 49
- 2) 28
- 3) 21
- 4) 7

2. Значения двух массивов $A[1..100]$ и $B[1..100]$ задаются с помощью следующего фрагмента программы:

Паскаль
for n:=1 to 100 do A[n]:=n-10; for n:=1 to 100 do B[n]:=A[n]*n

Сколько элементов массива B будут иметь положительные значения?

- 1) 10
- 2) 50
- 3) 90
- 4) 100

3. Значения двух массивов $A[1..100]$ и $B[1..100]$ задаются с помощью следующего фрагмента программы:

```
for n:=1 to 100 do  
  A[n]:= (n-80)*(n-80);  
for n:=1 to 100 do  
  B[101-n]:=A[n];
```

Какой элемент массива B будет наибольшим?

- 1) B[1]
- 2) B[21]
- 3) B[80]
- 4) B[100]

4. Дан фрагмент программы, обрабатывающей двумерный массив A размера $n \times n$.

k:=1; for i:=1 to n do begin c:=A[i,i]; A[i,i]:=A[k,i]; A[k,i]:=c end

Представим массив в виде квадратной таблицы, в которой для элемента массива $A[i,j]$ величина i является номером строки, а величина j – номером столбца, в котором расположен элемент. Тогда данный алгоритм меняет местами:

- 1) два столбца в таблице
- 2) две строки в таблице
- 3) элементы диагонали и k -ой строки таблицы
- 4) элементы диагонали и k -го столбца таблицы

Тест по теме 15

1. Маской сети называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес. Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске. По заданным IP-адресу узла и маске определите адрес сети. IP –адрес узла: 217.233.232.3 Маска: 255.255.252.0

При записи ответа выберите из приведенных в таблице чисел четыре элемента IP-адреса и запишите в нужном порядке соответствующие им буквы. Точки писать не нужно.

A	B	C	D	E	F	G	H
0	3	217	233	232	244	252	255

Пример. Пусть искомым IP-адрес 192.168.128.0, и дана таблица

A	B	C	D	E	F	G	H
128	168	255	8	127	0	17	192

В этом случае правильный ответ будет записан в виде: HBAF

- 1)HBAF 2) CDAE 3) HAFB 4) CDEA

2. Петя записал IP-адрес школьного сервера на листке бумаги и положил его в карман куртки. Петина мама случайно постирала куртку вместе с запиской. После стирки Петя обнаружил в кармане четыре обрывка с фрагментами IP-адреса. Эти фрагменты обозначены буквами А, Б, В и Г. Восстановите IP-адрес. В ответе укажите последовательность букв, обозначающих фрагменты, в порядке, соответствующем IP-адресу.

.64	3.13	3.133	20
A	B	B	Г

- 1) АБВГ 2) ГБВА 3)ГАВБ 4)ВАГБ

3. На месте преступления были обнаружены четыре обрывка бумаги. Следствие установило, что на них записаны фрагменты одного IP-адреса. Криминалисты обозначили эти фрагменты буквами А, Б, В и Г. Восстановите IP-адрес. В ответе укажите последовательность букв, обозначающих фрагменты, в порядке, соответствующем IP-адресу.

.64	2.16	16	8.132
A	B	B	Г

- 1) ВБГА
2) ГБВА
3) ГАВБ
4) ВАГБ

4. Доступ к файлу htm.net, находящемуся на сервере com.edu, осуществляется по протоколу ftp. В таблице фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети Интернет.

A	/	B	com	B	.edu	Г	://	Д	.net	E	htm	Ж	ftp
----------	---	----------	-----	----------	------	----------	-----	----------	------	----------	-----	----------	-----

- 1) БВГЕДАЖ
2) ЕДАБВГЖ
3) ЖГБВАЕД
4) БВАЖГЕД

Тест по теме 16

1. Внедряются в загрузочный сектор дискеты или в главную загрузочную запись жесткого диска ... вирусы
 - 1) Документные
 - 2) Загрузочные
 - 3) Файловые
 - 4) Сетевые

2. после завершения инфицированной программы остаются в оперативной памяти и продолжают свои деструктивные действия ... вирусы
 - 1) загрузочные
 - 2) резидентные
 - 3) нерезидентные
 - 4) мутирующие

3. Внедряются в исполняемые файлы с расширениями exe и com, в драйверы внешних устройств и библиотеки ... вирусы
 - 1) документные
 - 2) загрузочные
 - 3) файловые
 - 4) сетевые

4. Периодически изменяют свой программный код ... вирусы
 - 1) загрузочные
 - 2) паразитирующие
 - 3) троянские кони
 - 4) мутирующие

Тест по теме 17

1. К правовым методам защиты программных продуктов и баз данных не относятся:
 - 1) Патентная защита.
 - 2) Закон о защите прав потребителей.
 - 3) Лицензионные соглашения и контракты.
 - 4) Закон об авторском праве.

2. Какой тип лицензий на программный продукт дает пользователю право на использование одной копии программного продукта?
 - 1) исключительная лицензия
 - 2) простая лицензия
 - 3) правовая лицензия
 - 4) этикеточная лицензия

3. Какой тип лицензий на программный продукт дает пользователю право монопольного владения программным продуктом?
 - 1) этикеточная лицензия
 - 2) простая лицензия
 - 3) правовая лицензия
 - 4) исключительная лицензия

4. Контроль доступа не может быть обеспечен путем:
 - 1) парольной защиты программ при их запуске;
 - 2) выключения компьютера.
 - 3) использование ключевой дискеты для запуска программ;
 - 4) ограничения программ или данных, доступных пользователям

Тест по теме 18

1. Цифровая подпись не обладает следующим свойством:
 - 1) является неотъемлемой частью любого текстового документа
 - 2) удостоверяет, что подписанный текст исходит от лица, поставившего подпись
 - 3) не дает лицу, подписавшему текст, отказаться от обязательств
 - 4) гарантирует целостность подписанного текста

2. Управление ключами - информационный процесс, не включающий в себя этот элемент:
 - 1) генерацию ключей
 - 2) дублирование ключей
 - 3) накопление ключей
 - 4) распределение ключей.

3. Предмет или устройство, владение которым подтверждает подлинность пользователя, это ...
 - 1) сетчатка глаза
 - 2) токен
 - 3) пароль
 - 4) координаты

4. Символы шифруемого текста переставляются по некоторому правилу в пределах заданного блока передаваемого текста при использовании этого вида преобразований.
 - 1) перестановка
 - 2) подстановка
 - 3) аналитическое преобразование
 - 4) алгоритмическое преобразование

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Самарский государственный аэрокосмический университет
имени академика С.П. Королёва (национальный исследовательский
университет)» (СГАУ)

Факультет информатики
Кафедра программных систем

Сопченко Е.В.

Тесты для итогового контроля знаний
«ИНФОРМАТИКА»

для студентов, обучающихся:

по направлению 010300.62 «Фундаментальные
информатика и информационные технологии»

по направлению 230100.62 «Информатика
и вычислительная техника»

по специальности 090303.65 «Информационная
безопасность автоматизированных систем»

Самара 2012

СОДЕРЖАНИЕ

Тестирование по информатике - Вариант 1	3
Тестирование по информатике - Вариант 2	6
Тестирование по информатике - Вариант 3	9
Тестирование по информатике - Вариант 4	12

Тестирование по информатике - Вариант 1

1. Информация присутствует во всех материальных системах как органической, так и неорганической природы и является следствием свойства отражения, согласно ... подходу к определению информации
 - 1) философскому
 - 2) функционально-методологическому
 - 3) философско-методологическому
 - 4) кибернетическому

2. Наука о знаках семиотика **НЕ** включает в себя:
 - 1) синтаксис
 - 2) пунктуацию
 - 3) семантику
 - 4) прагматику

3. Показатель качества информации, который отражает правильность отбора и формирования информации...
 - 1) доступность
 - 2) точность
 - 3) репрезентативность
 - 4) достоверность

4. Кодирование информации это информационный процесс ...
 - 1) ввода-вывода информации
 - 2) хранения информации
 - 3) передачи информации
 - 4) обработки информации

5. Для параллельной передачи всех разрядов числового кода операнда используется
 - 1) шина адреса
 - 2) шина данных
 - 3) шина управления
 - 4) шина питания

6. Как представлено число 25_{10} в двоичной системе счисления?
 - 1) 1001_2
 - 2) 11001_2
 - 3) 10011_2
 - 4) 11010_2

7. Производится одноканальная (моно) звукозапись с частотой дискретизации 16 кГц и 24-битным разрешением. Запись длится 1 минуту, ее результаты записываются в файл, сжатие данных не производится. Какая из приведенных ниже величин наиболее близка к размеру полученного файла?
 - 1) 0.2 Мбайт
 - 2) 2 Мбайт
 - 3) 3 Мбайт
 - 4) 4 Мбайт

8. Группа школьников пришла в бассейн, в котором 4 дорожки для плавания. Тренер сообщил, что группа будет плавать на дорожке № 3. Сколько бит информации получили школьники из этого сообщения?
 - 1) 1
 - 2) 2
 - 3) 3
 - 4) 4

9. Какое логическое выражение равносильно выражению $\neg (A \wedge B) \wedge \neg C$?

- 1) $\neg A \vee B \vee \neg C$
- 2) $(\neg A \vee \neg B) \wedge \neg C$
- 3) $(\neg A \vee \neg B) \wedge C$
- 4) $\neg A \wedge \neg B \wedge \neg C$

10. Точная формулировка решения задачи на компьютере с описанием входной и выходной информации рассматриваемой предметной области – это ...

- 1) постановка задачи
- 2) предметная область
- 3) модель данных
- 4) входная информация

11. Применимость алгоритма для решения определенного класса задач – это ...

- 1) дискретность
- 2) детерминированность
- 3) результативность
- 4) массовость

12. Что **НЕ** является показателем качества программного продукта?

- 1) мобильность
- 2) надежность
- 3) достоверность
- 4) эффективность

13. Совокупность программ, позволяющая организовать решение задач пользователя на компьютере – это ...

- 1) операционная система
- 2) среда программирования
- 3) программное обеспечение
- 4) антивирус

14. Значения двумерного массива размера 7×7 задаются с помощью вложенного оператора цикла в представленном фрагменте программы

```
for n:=1 to 7 do
for k:=1 to 7 do
  B[n, k]:=k-n;
```

Сколько элементов массива будут иметь положительные значения?
1) 49 2) 28 3) 21 4) 7

15. Маской сети называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес. Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске. По заданным IP-адресу узла и маске определите адрес сети.

IP –адрес узла: 217.233.232.3 Маска: 255.255.252.0

При записи ответа выберите из приведенных в таблице чисел четыре элемента IP-адреса и запишите в нужном порядке соответствующие им буквы. Точки писать не нужно.

A	B	C	D	E	F	G	H
0	3	217	233	232	244	252	255

Пример. Пусть искомым IP-адрес 192.168.128.0, и дана таблица

A	B	C	D	E	F	G	H
128	168	255	8	127	0	17	192

В этом случае правильный ответ будет записан в виде: HBAF

- 1) HBAF 2) CDAE 3) HAFB 4) CDEA

16. Внедряются в загрузочный сектор дискеты или в главную загрузочную запись жесткого диска

... вирусы

- 1) Документные
- 2) Загрузочные
- 3) Файловые
- 4) Сетевые

17. К правовым методам защиты программных продуктов и баз данных не относятся:

- 1) Патентная защита.
- 2) Закон о защите прав потребителей.
- 3) Лицензионные соглашения и контракты.
- 4) Закон об авторском праве.

18. Цифровая подпись не обладает следующим свойством:

- 1) является неотъемлемой частью любого текстового документа
- 2) удостоверяет, что подписанный текст исходит от лица, поставившего подпись
- 3) не дает лицу, подписавшему текст, отказаться от обязательств
- 4) гарантирует целостность подписанного текста

Тестирование по информатике - Вариант 2

- Информация является свойством общества, живых существ и кибернетических устройств, согласно ... подходу к определению информации.
 - 1) философскому
 - 2) функционально-методологическому
 - 3) философско-методологическому
 - 4) кибернетическому
- Знаки, которые не имеют прямой зависимости между формой и значением, называются
 - 1) символы
 - 2) диокритики
 - 3) дистракторы
 - 4) указатели
- Показатель качества информации, который отражает степень сокращения ценности информации ...
 - 1) актуальность
 - 2) точность
 - 3) устойчивость
 - 4) достоверность
- Знак, физический процесс или явление, несущие сообщение о каком-либо событии, состоянии объекта либо передающие команды управления, оповещения и т. д. – это ...
 - 1) символ
 - 2) код
 - 3) сигнал
 - 4) кодовая таблица
- Для параллельной передачи всех разрядов числового кода адреса ячейки оперативной памяти или порта ввода-вывода внешнего устройства используется ...
 - 1) шина адреса
 - 2) шина данных
 - 3) шина управления
 - 4) шина питания
- Вычислите значение суммы $10_2 + 10_8 + 10_{16}$ в двоичной системе счисления.
 - 1) 10100010_2
 - 2) 11110_2
 - 3) 11010_2
 - 4) 10100_2
- Для кодирования цвета фона Интернет-страницы используется атрибут `bgcolor="#XXXXXX"`, где в кавычках задаются шестнадцатеричные значения интенсивности цветовых компонент в 24-битной RGB-модели. К какому цвету будет близок цвет страницы, заданный тэгом `<body bgcolor="#747474">`?
 - 1) серый
 - 2) белый
 - 3) фиолетовый
 - 4) черный
- Сообщение о том, что ваш друг живет на 10 этаже несет 4 бита информации. Сколько этажей в доме?
 - 1) 16
 - 2) 4
 - 3) 10
 - 4) 40

9. Для какого из указанных значений числа X истинно высказывание

$((X < 5) \rightarrow (X < 3)) \wedge ((X < 2) \rightarrow (X < 1))$

- 1) 1
- 2) 2
- 3) 3
- 4) 4

10. Совокупность связанных между собой функций, задач управления, с помощью которых достигается выполнение поставленных целей – это ...

- 1) постановка задачи
- 2) предметная область
- 3) модель данных
- 4) входная информация

11. Обязательное получение желаемого результата за конечное число шагов при допустимых исходных данных это ...

- 1) дискретность
- 2) детерминированность
- 3) результативность
- 4) массовость

12. Что не является показателем качества программного продукта?

- 1) дискретность
- 2) надежность
- 3) мобильность
- 4) эффективность

13. Программы, управляющие оперативной памятью, процессором, внешними устройствами и обеспечивающие возможность работы других программ, называют

- 1) утилитами
- 2) драйверами
- 3) операционными системами
- 4) системами программирования

14. Значения двух массивов A[1..100] и B[1..100] задаются с помощью следующего фрагмента программы:

Паскаль
for n:=1 to 100 do A[n]:=n-10;
for n:=1 to 100 do B[n]:=A[n]*n

Сколько элементов массива B будут иметь положительные значения?

- 1) 10
- 2) 50
- 3) 90
- 4) 100

15. Петя записал IP-адрес школьного сервера на листке бумаги и положил его в карман куртки. Петина мама случайно постирала куртку вместе с запиской. После стирки Петя обнаружил в кармане четыре обрывка с фрагментами IP-адреса. Эти фрагменты обозначены буквами А, Б, В и Г. Восстановите IP-адрес. В ответе укажите последовательность букв, обозначающих фрагменты, в порядке, соответствующем IP-адресу.

.64	3.13	3.133	20
А	Б	В	Г

- 1) АБВГ
- 2) ГБВА
- 3) ГАВБ
- 4) ВАГБ

16. после завершения инфицированной программы остаются в оперативной памяти и продолжают свои деструктивные действия ... вирусы

- 1) загрузочные
- 2) резидентные
- 3) нерезидентные
- 4) мутирующие

17. Какой тип лицензий на программный продукт дает пользователю право на использование одной копии программного продукта?

- 1) исключительная лицензия
- 2) простая лицензия
- 3) правовая лицензия
- 4) этикеточная лицензия

18. Управление ключами - информационный процесс, не включающий в себя этот элемент:

- 1) генерацию ключей
- 2) дублирование ключей
- 3) накопление ключей
- 4) распределение ключей.

Тестирование по информатике - Вариант 3

1. Эта информационная революция связана с изобретением письменности.
 - 1) первая
 - 2) вторая
 - 3) третья
 - 4) пятая
2. Совокупность сведений, которыми располагает пользователь или система – это ...
 - 1) запись
 - 2) база данных
 - 3) тезаурус
 - 4) атрибут
3. Показатель качества информации, который отражает степень близости получаемой информации реальному состоянию объекта ...
 - 1) актуальность
 - 2) точность
 - 3) устойчивость
 - 4) достоверность
4. Совокупность средств, методов и персонала, используемых для приема, хранения, обработки и передачи информации, т.е. для организации информационных процессов называется ...
 - 1) информационным процессом
 - 2) информационной системой
 - 3) аппаратной частью
 - 4) информационным обеспечением
5. Для передачи управляющих сигналов во все блоки ПК используется ...
 - 1) шина данных
 - 2) шина адреса
 - 3) шина управления
 - 4) шина питания
6. Как представлено число 83_{10} в двоичной системе счисления?
 - 1) 1001011_2
 - 2) 1100101_2
 - 3) 1010011_2
 - 4) 101001_2
7. Для хранения растрового изображения размером 128×128 пикселей отвели 4 килобайта памяти. Каково максимально возможное число цветов в палитре изображения?
 - 1) 8
 - 2) 2
 - 3) 16
 - 4) 4
8. В коробке лежат 7 разноцветных карандашей. Какое количество информации содержит сообщение, что из коробки достали красный карандаш?
 - 1) 56 бит
 - 2) 3 бита
 - 3) 7 бит
 - 4) 3 байта
9. Укажите, какое логическое выражение равносильно выражению $A \wedge \neg (\neg B \vee C)$.
 - 1) $\neg A \vee \neg B \vee \neg C$
 - 2) $A \wedge \neg B \wedge \neg C$
 - 3) $A \wedge B \wedge \neg C$
 - 4) $A \wedge \neg B \wedge C$

10. Используемая знаковая система для обозначения структур данных и операций их обработки называется ...

- 1) постановкой задачи
- 2) предметной областью
- 3) моделью данных
- 4) входной информацией

11. Разбиение процесса обработки информации на более простые этапы (шаги), выполнение которых компьютером или человеком не вызывает затруднений, это ...

- 1) дискретность
- 2) детерминированность
- 3) результативность
- 4) массовость

12. Что не является показателем качества программного продукта?

- 1) коммуникативность
- 2) надежность
- 3) мобильность
- 4) массовость

13. Ниже перечислено 8 различных программных средств. Какие из них являются операционными системами? (Выберите ответ, в котором перечислены только операционные системы)

- A) Acrobat Reader B) IBM PC DOS Д) Microsoft Office Ж) Norton SystemWorks
Б) ASP Linux Г) Macromedia Dreamweaver Е) Microsoft Windows З) RealOne Player

- 1) АВЕЗ
- 2) БВЕ
- 3) ВДЕ
- 4) ЕЖ

14. Значения двух массивов A[1..100] и B[1..100] задаются с помощью следующего фрагмента программы:

```
for n:=1 to 100 do  
  A[n]:= (n-80)*(n-80);  
for n:=1 to 100 do  
  B[101-n]:=A[n];
```

Какой элемент массива B будет наибольшим?
1) B[1] 2) B[21] 3) B[80] 4) B[100]

15. На месте преступления были обнаружены четыре обрывка бумаги. Следствие установило, что на них записаны фрагменты одного IP-адреса. Криминалисты обозначили эти фрагменты буквами А, Б, В и Г. Восстановите IP-адрес. В ответе укажите последовательность букв, обозначающих фрагменты, в порядке, соответствующем IP-адресу.

.64	2.16	16	8.132
А	Б	В	Г

- 1) ВБГА
- 2) ГБВА
- 3) ГАВБ
- 4) ВАГБ

16. Внедряются в исполняемые файлы с расширениями exe и com, в драйверы внешних устройств и библиотеки ... вирусы

- 1) документные
- 2) загрузочные
- 3) файловые
- 4) сетевые

17. Какой тип лицензий на программный продукт дает пользователю право монопольного владения программным продуктом?

- 1) этикеточная лицензия
- 2) простая лицензия
- 3) правовая лицензия
- 4) исключительная лицензия

18. Предмет или устройство, владение которым подтверждает подлинность пользователя, это ...

- 1) сетчатка глаза
- 2) токен
- 3) пароль
- 4) координаты

Тестирование по информатике - Вариант 4

1. Эта информационная революция обусловлена открытием электричества, благодаря которому появились телеграф, телефон, радио, позволяющие оперативно передавать информацию в любом объеме.
 - 1) первая
 - 2) вторая
 - 3) третья
 - 4) пятая
2. Эта мера определяет полезность (ценность) информации для достижения пользователем поставленной цели.
 - 1) синтаксическая
 - 2) прагматическая
 - 3) семантическая
 - 4) финансовая
3. Показатель качества информации, который отражает способность реагировать на изменение исходных данных без нарушения необходимой точности ...
 - 1) актуальность
 - 2) точность
 - 3) устойчивость
 - 4) достоверность
4. Обратная связь необходима для...
 - 1) ввода-вывода информации
 - 2) хранения информации
 - 3) передачи информации
 - 4) корректировки входной информации
5. Для подключения блоков ПК к системе электропитания используется ...
 - 1) шина питания
 - 2) шина адреса
 - 3) шина управления
 - 4) шина данных
6. Вычислите сумму двоичных чисел x и y , если $x = 1010101_2$ $y = 1010011_2$
 - 1) 10100010_2
 - 2) 10101000_2
 - 3) 10100100_2
 - 4) 10111000_2
7. Укажите минимальный объем памяти (в килобайтах), достаточный для хранения любого растрового изображения размером 64×64 пикселя, если известно, что в изображении используется палитра из 256 цветов. Саму палитру хранить не нужно.
 - 1) 128
 - 2) 2
 - 3) 256
 - 4) 4
8. Сообщение о том, что Иван живет во втором подъезде, несет 3 бита информации. Сколько подъездов в доме?
 - 1) 24
 - 2) 3
 - 3) 8
 - 4) 6

9. Какое из приведенных имен удовлетворяет логическому условию \neg (первая буква гласная \rightarrow вторая буква гласная) \wedge последняя буква гласная

- 1) ИРИНА
- 2) МАКСИМ
- 3) АРТЕМ
- 4) МАРИЯ

10. Данные, поступающие на вход задачи и используемые для ее решения, это ...

- 1) входная информация
- 2) предметная область
- 3) модель данных
- 4) постановка задачи

11. Однозначность получаемого результата при одних и тех же исходных данных, это ...

- 1) дискретность
- 2) детерминированность
- 3) результативность
- 4) массовость

12. Что не является показателем качества программного продукта?

- 1) мобильность
- 2) надежность
- 3) актуальность
- 4) эффективность

13. совокупность программных средств, обеспечивающая управление аппаратной частью компьютера и прикладными программами, а также их взаимодействие между собой и пользователем.

- 1) операционная система
- 2) среда программирования
- 3) программное обеспечение
- 4) антивирус

14. Дан фрагмент программы, обрабатывающей двумерный массив A размера $n \times n$.

```
k:=1;
for i:=1 to n do
begin
  c:=A[i,i];
  A[i,i]:=A[k,i];
  A[k,i]:=c
end
```

Представим массив в виде квадратной таблицы, в которой для элемента массива $A[i,j]$ величина i является номером строки, а величина j – номер столбца, в котором расположен элемент. Тогда данный алгоритм меняет местами:

- 1) два столбца в таблице
- 2) две строки в таблице
- 3) элементы диагонали и k -ой строки таблицы
- 4) элементы диагонали и k -го столбца таблицы

15. Доступ к файлу htm.net, находящемуся на сервере com.edu, осуществляется по протоколу ftp. В таблице фрагменты адреса файла закодированы буквами от А до Ж. Запишите последовательность этих букв, кодирующую адрес указанного файла в сети Интернет.

А	/	Б	com	В	.edu	Г	://	Д	.net	Е	htm	Ж	ftp
---	---	---	-----	---	------	---	-----	---	------	---	-----	---	-----

- 1) БВГЕДАЖ
- 2) ЕДАБВГЖ
- 3) ЖГБВАЕД
- 4) БВАЖГЕД

16. Периодически изменяют свой программный код ... вирусы

- 1) загрузочные
- 2) паразитирующие
- 3) троянские кони
- 4) мутирующие

17. Контроль доступа не может быть обеспечен путем:

- 1) парольной защиты программ при их запуске;
- 2) выключения компьютера.
- 3) использование ключевой дискеты для запуска программ;
- 4) ограничения программ или данных, доступных пользователям

18. Символы шифруемого текста переставляются по некоторому правилу в пределах заданного блока передаваемого текста при использовании этого вида преобразований.

- 1) перестановка
- 2) подстановка
- 3) аналитическое преобразование
- 4) алгоритмическое преобразование

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Самарский государственный аэрокосмический университет
имени академика С.П. Королёва (национальный исследовательский
университет)» (СГАУ)

Факультет информатики
Кафедра программных систем

Сопченко Е.В.

Вопросы к экзамену
«ИНФОРМАТИКА»

для студентов, обучающихся:

по направлению 010300.62 «Фундаментальные
информатика и информационные технологии»

по направлению 230100.62 «Информатика и
вычислительная техника»

по специальности 090303.65 «Информационная
безопасность автоматизированных систем»

Самара 2012

Вопросы к экзамену

1. Информационные революции.
2. Основные подходы к определению информации.
3. Знаковые системы.
4. Семиотические аспекты информатики.
5. Основные отличия искусственных языков от естественных языков.
6. Информация: виды и свойства.
7. Информация и данные. Адекватность информации.
8. Меры информации. Синтаксическая мера.
9. Семантическая и прагматическая меры информации.
10. Показатели качества информации.
11. Классификация информации.
12. Информационные системы и процессы.
13. Компьютер как универсальная информационная система. Структурная схема ПК.
14. Системный блок: основные устройства и их назначение.
15. Устройство обработки информации. Функции микропроцессора.
16. Устройства хранения информации. Понятия записи и чтения информации. Характеристики памяти.
17. Внутренняя память.
18. Внешняя память.
19. Развитие средств хранения информации.
20. Устройства ввода информации.
21. Устройства вывода информации.
22. Представление информации в компьютере. Системы кодирования.
23. Системы счисления и формы представления чисел.
24. Основные системы счисления, используемые для представления информации в ЭВМ.
25. Варианты представления информации в ПК. Естественная форма, нормальная форма представления чисел.
26. Алгебра логики. Логические функции. Таблицы истинности.
27. Методы работы с процедурной информацией. Процесс решения задач на компьютере.
28. Постановка задачи. Основные характеристики задач.
29. Построение алгоритма решения задачи.
30. Графическая форма представления алгоритмов. Примеры.
31. Сигнал: кодирование, декодирование.
32. Основные подходы к определению количества информации. Понятие энтропии.
33. Формула Шеннона, формула Хартли.
34. Стадии разработки программного продукта.
35. Характеристики программного продукта.
36. Показатели качества программного продукта.
37. Программное обеспечение компьютера. Состав и назначение.
38. Вычислительные сети. Локальные и глобальные вычислительные сети. Топологии сети. Адресация в сети.
39. Компьютерные вирусы и методы защиты от них.
40. Понятие несанкционированного доступа. Методы защиты информации.
41. Криптографические методы защиты информации.

Список использованных источников

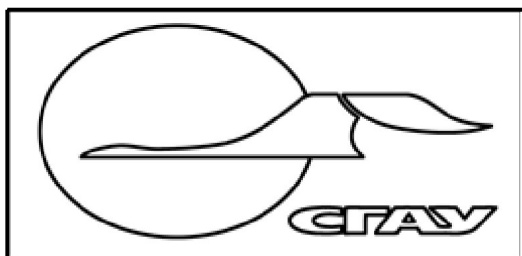
Основная и дополнительная литература

- 1 Баландин, Александр Васильевич. Курс лекций по информатике [Текст]: для студентов заоч. формы обучения / А. В. Баландин, Л. С. Зеленко, О. П. Солдатова; Самар. гос. аэрокосм. ун-т им. С. П. Королева. - Самара : [б. и.], 1999. - 57 с.
- 2 Вирт Н. Алгоритмы и структуры данных [Текст].- М.: Мир, 1989.
- 3 Иванова, Галина Сергеевна. Основы программирования [Текст]: [учеб. по направлению "Информатика и вычисл. техника"] / Г. С. Иванова. - 4-е изд., стер. - М. : Изд-во МГТУ, 2007. - 415 с. - (Информатика в техническом университете).
- 4 Информатика [Текст] : [учеб. пособие] / А. В. Могилев, Н. И. Пак, Е. К. Хеннер; под ред. Е. К. Хеннера. - 3-е изд., перераб. и доп. - М. : Academia, 2004. - 841 с. - (Высшее профессиональное образование. Педагогические специальности).
- 5 Информатика. Практикум по технологии работы на компьютере [Текст]: Под ред. Макаровой Н.В. 2-е изд., перераб. - М.: Финансы и статистика, 2005. — 256 с.
- 6 Информатика: Учебник для вузов [Текст]: Под ред. Макаровой Н.В. - 3-е изд., перераб. - М.: Финансы и статистика, 2009.
- 7 Калентьев, Анатолий Алексеевич. Основы информатики [Текст]: Учеб. пособие / А. А. Калентьев; Самар. гос. аэрокосм. ун-т им. С.П.Королева. - М.: [б.и.], 2003.
- 8 Кнут Д.Э. Искусство программирования [Текст]. Том 3. Сортировка и поиск. М.: Вильямс, 2004.
- 9 Королев, Лев Николаевич. Информатика. Введение в компьютерные науки: [учеб. для вузов] / Л. Н. Королев, А. И. Миков. - М. : Высш. шк., 2003. - 341 с.
- 10 Мальцев А.И. Алгоритмы и рекурсивные функции [Текст].- М.: Наука, 1965.
- 11 Окулов, Станислав Михайлович. Основы программирования [Текст]: [учебник] / С. Окулов. - М. : Бином. Лаб. знаний, 2004. - 424 с.
- 12 Олифер Н.А., Олифер В.Г.: Основы компьютерных сетей [Текст]. [учебное пособие]. – СПб.: Питер, 2009.
- 13 Симонова Е.В. Курс лекций по динамическим структурам данных [Текст]. Самара: 2000, СГАУ, 112 с.

Электронные источники и интернет ресурсы

- 14 TeachPro Информатика [Электронный ресурс]: Мультимедийный учебник.-М.: Copyright, 2001.-1 электрон. опт. диск (CD-ROM).
- 15 Информатика [Электронный ресурс]: учебное электронное издание / УГАТУ.- Мультимедиа ресурсы (10 директорий; 100 файлов; 740МВ).-Уфа: УГАТУ, 2005.-1 электрон. опт. диск (CD-ROM).
- 16 Научная электронная библиотека eLibrary (www.elibrary.ru)
- 17 Сайт www.kpolyakov.narod.ru.
- 18 Сайт www.infojournal.ru.
- 19 Сайт www.slovari.yandex.ru
- 20 Симановский, Евгений Аркадьевич. Введение в информатику [Электронный ресурс] : [учеб. пособие] / Е. А. Симановский ; М-во образования и науки РФ, Самар. гос. аэрокосм. ун-т им. С. П. Королева (Нац. исслед. ун-т). - Электрон. дан. (1 файл : 988 Кбайт). - Самара : Изд-во СГАУ, 2010. – Полный доступ – электронный ресурс.
- 21 Сопченко, Елена Вильевна. Разработка гипертекстовой среды обучения по курсу "Информатика" [Электронный ресурс] / Е. В. Сопченко, О. А. Дегтярева ; Федер. агентство по образованию, Самар. гос. аэрокосм. ун-т им. С. П. Королева, Нац. исслед. ун-т. - Электрон. текстовые дан. - Самара : [б. и.], 2009. - 1 эл. опт. диск (CD-RW). - (Программа развития государственного образовательного учреждения высшего профессионального образования "Самарский государственный аэрокосмический университет имени академика С. П. Королева" на 2009-2018 годы. Полный доступ – электронный ресурс.
- 22 Электронный каталог НТБ СГАУ (lib.ssau.ru)

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Самарский государственный аэрокосмический
университет имени академика С.П. Королёва
(национальный исследовательский университет)»



СОГЛАСОВАНО

УТВЕРЖДАЮ

Управление образовательных программ

Проректор по учебной работе

_____ / А.В. Дорошин /

_____ / Ф.В. Гречников /

" ____ " _____ 20__ г.

" ____ " _____ 20__ г.

РАБОЧАЯ ПРОГРАММА

Наименование модуля (дисциплины)

Информатика

Цикл, в рамках которого происходит освоение модуля (дисциплины)

Б3. Профессиональный цикл

Часть цикла

Базовая

Код учебного плана

010300.2.62-2011-О-П-4г00м

Факультет

6

Кафедра

Программных систем

Курс

1

Семестр

1

Лекции (СЛ)

36

Семинарские и практические занятия (СП)

18

Лабораторные занятия (СЛР)

18

Экзамен 1

Контроль самостоятельной работы (КСР)

0

Зачет

Самостоятельная работа (СРС)

108

Всего

180

Наименование стандарта, на основании которого составлена рабочая программа:

010300 Фундаментальная информатика и информационные технологии
(квалификация (степень) «бакалавр»)

Соответствие содержания рабочей программы, условий ее реализации, материально-технической и учебно-методической обеспеченности учебного процесса по дисциплине всем требованиям государственных стандартов подтверждаем.

Составители:

Сопченко Елена Вильевна, к.т.н., доцент

_____ /
(подпись)

Заведующий кафедрой:

Коварцев Александр Николаевич, д.т.н.,
профессор

_____ /
(подпись)

Рабочая программа обсуждена на заседании кафедры

Программных систем

Протокол № ___ от " ___ " _____ 20__ г.

Наличие основной литературы в фондах научно-технической библиотеки (НТБ)
подтверждаем:

Директор НТБ

_____ /
(подпись)

_____ /
(расшифровка подписи)

Согласовано:

Декан

_____ /
(подпись)

_____ /
(расшифровка подписи)

1 Цели и задачи модуля (дисциплины), требования к уровню освоения содержания

1.1 Перечень развиваемых компетенций

ОК-1, ОК-4, ОК-5, ОК-6, ОК-11, ОК-12, ОК-15, – общекультурные компетенции
ПК-1, ПК-2 - профессиональная компетенция

1.2 Цели и задачи изучения модуля (дисциплины)

1. Ознакомление студентов с основами информатики;
2. Обучение студентов методам представления информации, построения алгоритмов и реализации алгоритмов на языках программирования;
3. Изучение фундаментальных основ информатики, часто применяемых в практической деятельности на вычислительных машинах.

1.3 Требования к уровню подготовки студента, завершившего изучение данного модуля (дисциплины)

Студенты, завершившие изучение данной дисциплины, должны знать:

1. Семантические аспекты информатики, основные методы расчета количества информации;
2. Структуру основных элементов компьютера;
3. Основы построения алгоритмических структур и их реализации на языках программирования высокого уровня;
4. Алгоритмы обработки базовых структур данных.

1.4 Связь с предшествующими модулями (дисциплинами)

Данная дисциплина является базовой, изучается в 1 семестре, поэтому не требует знания студентами дополнительных курсов.

1.5 Связь с последующими модулями (дисциплинами)

Курс «Информатика» относится к разделу общепрофессиональных дисциплин федерального компонента, материалы которого могут быть использованы в следующих курсах:

- Вычислительная математика;
- Логическое программирование;
- Моделирование информационных процессов и систем;
- Методы оптимизации и исследование операций.

2 Содержание рабочей программы (модуля)

Семестр 1		
СЛ 0,2 36 часов 1 ЗЕТ	Активные 1	Лекция 1. Информатика и ее связь с другими науками. Основные подходы к определению информации. Информация и данные.

		Лекция 2. Знаковые системы. Семиотические аспекты информатики. Адекватность информации. Меры информации.
		Лекция 3. Показатели качества информации. Классификация информации.
		Лекция 4. Сигнал: кодирование, декодирование. Информационные процессы и системы.
		Лекция 5. Структурная схема персонального компьютера. Основные устройства ПК и их назначение. Память. Понятия чтения-записи.
		Лекция 6. Представление информации в компьютере. Системы счисления.
		Лекция 7. Варианты представления информации в ПК. Кодирование текстовой, графической и звуковой информации.
		Лекция 8. Основные подходы к определению количества информации. Понятие энтропии. Формула Шеннона, формула Хартли.
		Лекция 9. Элементы алгебры логики. Основные логические функции. Законы преобразования логических выражений.
		Лекция 10. Методы работы с процедурной информацией. Постановка задачи. Моделирование информации.
		Лекция 11. Построение алгоритма решения задачи. Графическая форма представления алгоритмов. Пример реализации различных форм алгоритма.
		Лекция 12. Стадии разработки программного продукта. Характеристики программного продукта. Показатели качества программного продукта.
		Лекция 13. Программное обеспечение компьютера. Состав и назначение.
		Лекция 14. Базовые типы данных. Основные алгоритмы обработки.
		Лекция 15. Вычислительные сети. Локальные и глобальные вычислительные сети. Топологии сети. Адресация в сети.
		Лекция 16. Компьютерные вирусы и методы защиты от них.

		Лекция 17. Понятие несанкционированного доступа. Методы защиты информации. Криптографические методы защиты информации.
	Интерактивные 0	
	Традиционные 0	
СП 0,1 18 часов 0,5 ЗЕТ	Активные 0	
	Интерактивные 1	Занятие 1. Системы счисления, используемые в ПК.
		Занятие 2. Вычисление количества информации.
		Занятие 3. Преобразование логических выражений.
		Занятие 4. Построение алгоритмических структур.
		Занятие 5. Одномерные и двумерные массивы. Сортировка.
		Занятие 6. Символьный и строковый тип. Обработка.
		Занятие 7. Динамические структуры данных. Стек. Очередь. Реализация односвязных списков.
		Занятие 8. Динамические структуры данных. Стек. Очередь. Реализация двусвязных списков.
		Занятие 9. Рекурсия. Деревья.
	Традиционные 0	
СЛР 0,1 18 часов 0,5 ЗЕТ	Активные 0	Лабораторная работа № 1. Структура программы на Паскале. Основные типы данных.
		Лабораторная работа № 2. Обработка одномерных массивов.
		Лабораторная работа № 3. Обработка двумерных массивов.
		Лабораторная работа № 4. Использование процедур и функций.
		Лабораторная работа № 5. Обработка строковых и символьных переменных.
		Лабораторная работа № 6. Обработка числовых рядов.

		Лабораторная работа № 7. Сортировка массивов.
		Лабораторная работа № 8. Текстовые файлы.
		Лабораторная работа № 9. Файлы записей.
	Интерактивные 1	Лабораторная работа № 1. Структура программы на Паскале. Основные типы данных.
		Лабораторная работа № 2. Обработка одномерных массивов.
		Лабораторная работа № 3. Обработка двумерных массивов.
		Лабораторная работа № 4. Использование процедур и функций.
		Лабораторная работа № 5. Обработка строковых и символьных переменных.
		Лабораторная работа № 6. Обработка числовых рядов.
		Лабораторная работа № 7. Сортировка массивов.
		Лабораторная работа № 8. Текстовые файлы.
		Лабораторная работа № 9. Файлы записей.
	Традиционные 0	
КСР 0 0 часов 0 ЗЕТ	Активные 0	
	Интерактивные 0	
	Традиционные 0	
СРС 0,6 108 часов 3 ЗЕТ	Активные 0	
	Интерактивные 1	Реферат на тему: «Алгоритмы быстрой сортировки»
		Реферат на тему: «Динамические структуры»
		Реферат на тему: «Развитие современных устройств обработки информации»
	Традиционные 0	

3 Инновационные методы обучения

Для автоматизации учебного процесса преподавания курса «Информатика» на кафедре программных систем используется система дистанционного обучения (СДО), построенная на основе стандарт Moodle, где использовались широкие возможности системы для организации

тестирования промежуточных знаний студентов.

В рамках учебного курса с помощью системы СДО «Кафедра ПС» разработан комплект тестов, который позволяет преподавателю легко проверить как текущие, так и остаточные знания студентов. Автоматизированное тестирование студентов можно проводить во время занятий для того, чтобы оценить уровень их подготовки по заданному разделу. В качестве тестовых заданий можно задавать задачи небольшой трудоемкости, не требующие больших затрат на их решение.

4 Технические средства и материальное обеспечение учебного процесса

Компьютерный класс с IBM-совместимыми компьютерами, объединенными в локальную сеть с выходом в сеть Интернет.

5 Учебно-методическое обеспечение

5.1 Основная литература

1. Иванова, Галина Сергеевна. Основы программирования [Текст]: [учеб. по направлению "Информатика и вычисл. техника"] / Г. С. Иванова. - 4-е изд., стер. - М. : Изд-во МГТУ, 2007. - 415 с. - (Информатика в техническом университете). (10 экз.)
2. Калентьев, Анатолий Алексеевич. Основы информатики [Текст] : Учеб. пособие / А. А. Калентьев; Самар. гос. аэрокосм. ун-т им. С.П.Королева. - М.: [б.и.], 2003. (20 экз.)

5.2 Дополнительная литература

1. Королев, Лев Николаевич. Информатика. Введение в компьютерные науки : [учеб. для вузов] / Л. Н. Королев, А. И. Миков. - М. : Высш. шк., 2003. - 341 с. (2 экз.)
2. Информатика [Текст] : [учеб. пособие] / А. В. Могилев, Н. И. Пак, Е. К. Хеннер ; под ред. Е. К. Хеннера. - 3-е изд., перераб. и доп. - М. : Academia, 2004. - 841 с. - (Высшее профессиональное образование. Педагогические специальности). (3 экз.)
3. Окулов, Станислав Михайлович. Основы программирования : учебник / С. Окулов. - М. : Бином. Лаб. знаний , 2004. - 424 с. (3 экз.)
4. Баландин, Александр Васильевич. Курс лекций по информатике [Текст] : для студентов заоч. формы обучения / А. В. Баландин, Л. С. Зеленко, О. П. Солдатова ; Самар. гос. аэрокосм. ун-т им. С. П. Королева. - Самара : [б. и.], 1999. - 57 с. (175 экз.)

5.3 Электронные источники и интернет ресурсы

1. Электронный каталог НТБ СГАУ (lib.ssau.ru)
2. Научная электронная библиотека eLibrary (www.elibrary.ru)
3. Сопченко, Елена Вильевна. Разработка гипертекстовой среды обучения по курсу "Информатика" [Электронный ресурс] / Е. В. Сопченко, О. А. Дегтярева ; Федер. агентство по образованию, Самар. гос. аэрокосм. ун-т им. С. П. Королева, Нац. исслед. ун-т. - Элек.
4. Симановский, Евгений Аркадьевич. Введение в информатику [Электронный ресурс] : [учеб. пособие] / Е. А. Симановский ; М-во образования и науки РФ, Самар. гос. аэрокосм. ун-т им. С. П. Королева (Нац. исслед. ун-т). - Электрон. дан.

(1 файл : 988 Кбайт).

5.4 Методические указания и рекомендации

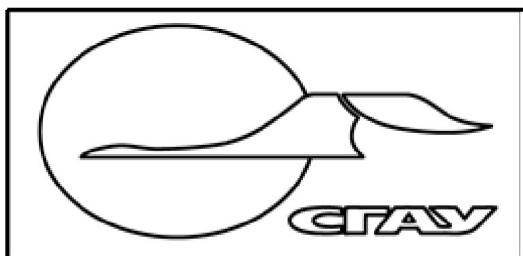
В процессе преподавания курса лекций основное внимание необходимо уделять разъяснению основных понятий курса.

Для оценки качества знаний необходимо в течение семестра проводить как промежуточное, так и окончательное тестирование.

При изложении теоретического материала желательно пользоваться иллюстративными материалами в виде слайдов или презентаций, чтобы повысить наглядность подачи материала и степень его запоминания.

Текущий контроль знаний студентов проводится на каждом практическом занятии в виде теоретического опроса студентов. По мере освоения тематических блоков дисциплины реализуется промежуточное тестирование уровня знаний студентов.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Самарский государственный аэрокосмический
университет имени академика С.П. Королёва
(национальный исследовательский университет)»



СОГЛАСОВАНО

УТВЕРЖДАЮ

Управление образовательных программ

Проректор по учебной работе

_____ / А.В. Дорошин /

_____ / Ф.В. Гречников /

" ____ " _____ 20__ г.

" ____ " _____ 20__ г.

РАБОЧАЯ ПРОГРАММА

Наименование модуля (дисциплины)

Информатика

Цикл, в рамках которого происходит освоение модуля (дисциплины)

Б2. Математический и естественнонаучный цикл

Часть цикла

Базовая

Код учебного плана

230100.2.62-2011-О-П-4г00м

Факультет

6

Кафедра

Программных систем

Курс

1

Семестр

1

Лекции (СЛ)

36

Семинарские и практические занятия (СП)

0

Лабораторные занятия (СЛР)

18

Экзамен 1

Контроль самостоятельной работы (КСР)

18

Зачет

Самостоятельная работа (СРС)

72

Всего

144

Наименование стандарта, на основании которого составлена рабочая программа:

230100 Информатика и вычислительная техника (квалификация (степень) «бакалавр»)

Соответствие содержания рабочей программы, условий ее реализации, материально-технической и учебно-методической обеспеченности учебного процесса по дисциплине всем требованиям государственных стандартов подтверждаем.

Составители:

Сопченко Елена Вильевна, к.т.н., доцент

_____ /
(подпись)

Заведующий кафедрой:

Коварцев Александр Николаевич, д.т.н.,
профессор

_____ /
(подпись)

Рабочая программа обсуждена на заседании кафедры

Программных систем

Протокол № ___ от " ___ " _____ 20__ г.

Наличие основной литературы в фондах научно-технической библиотеки (НТБ) подтверждаем:

Директор НТБ

_____ /
(подпись)

_____ /
(расшифровка подписи)

Согласовано:

Декан

_____ /
(подпись)

_____ /
(расшифровка подписи)

1 Цели и задачи модуля (дисциплины), требования к уровню освоения содержания

1.1 Перечень развиваемых компетенций

ОК-1, ОК-4, ОК-5, ОК-6, ОК-11, ОК-12, ОК-15, – общекультурные компетенции
ПК-1, ПК-2 - профессиональная компетенция

1.2 Цели и задачи изучения модуля (дисциплины)

1. Ознакомление студентов с основами информатики;
2. Обучение студентов методам представления информации, построения алгоритмов и реализации алгоритмов на языках программирования;
3. Изучение фундаментальных основ информатики, часто применяемых в практической деятельности на вычислительных машинах.

1.3 Требования к уровню подготовки студента, завершившего изучение данного модуля (дисциплины)

Студенты, завершившие изучение данной дисциплины, должны знать:

1. Семантические аспекты информатики, основные методы расчета количества информации;
2. Структуру основных элементов компьютера;
3. Основы построения алгоритмических структур и их реализации на языках программирования высокого уровня;
4. Алгоритмы обработки базовых структур данных.

1.4 Связь с предшествующими модулями (дисциплинами)

Данная дисциплина является базовой, изучается в 1 семестре, поэтому не требует знания студентами дополнительных курсов.

1.5 Связь с последующими модулями (дисциплинами)

Курс «Информатика» относится к разделу общепрофессиональных дисциплин федерального компонента, материалы которого могут быть использованы в следующих курсах:

- Вычислительная математика;
- Логическое программирование;
- Моделирование информационных процессов и систем;
- Методы оптимизации и исследование операций.

2 Содержание рабочей программы (модуля)

Семестр 1		
СЛ 0,25 36 часов 1 ЗЕТ	Активные 1	Лекция 1. Информатика и ее связь с другими науками. Основные подходы к определению информации. Информация и данные.

		Лекция 2. Знаковые системы. Семиотические аспекты информатики. Адекватность информации. Меры информации.
		Лекция 3. Показатели качества информации. Классификация информации.
		Лекция 4. Сигнал: кодирование, декодирование. Информационные процессы и системы.
		Лекция 5. Структурная схема персонального компьютера. Основные устройства ПК и их назначение. Память. Понятия чтения-записи.
		Лекция 6. Представление информации в компьютере. Системы счисления.
		Лекция 7. Варианты представления информации в ПК. Кодирование текстовой, графической и звуковой информации.
		Лекция 8. Основные подходы к определению количества информации. Понятие энтропии. Формула Шеннона, формула Хартли.
		Лекция 9. Элементы алгебры логики. Основные логические функции. Законы преобразования логических выражений.
		Лекция 10. Методы работы с процедурной информацией. Постановка задачи. Моделирование информации.
		Лекция 11. Построение алгоритма решения задачи. Графическая форма представления алгоритмов. Пример реализации различных форм алгоритма.
		Лекция 12. Стадии разработки программного продукта. Характеристики программного продукта. Показатели качества программного продукта.
		Лекция 13. Программное обеспечение компьютера. Состав и назначение.
		Лекция 14. Базовые типы данных. Основные алгоритмы обработки.
		Лекция 15. Вычислительные сети. Локальные и глобальные вычислительные сети. Топологии сети. Адресация в сети.
		Лекция 16. Компьютерные вирусы и методы защиты от них.

		Лекция 17. Понятие несанкционированного доступа. Методы защиты информации. Криптографические методы защиты информации.
	Интерактивные 0	
	Традиционные 0	
СП 0 0 часов 0 ЗЕТ	Активные 0	
	Интерактивные 1	Занятие 1. Системы счисления, используемые в ПК.
		Занятие 2. Вычисление количества информации.
		Занятие 3. Преобразование логических выражений.
		Занятие 4. Построение алгоритмических структур.
		Занятие 5. Одномерные и двумерные массивы. Сортировка.
		Занятие 6. Символьный и строковый тип. Обработка.
		Занятие 7. Динамические структуры данных. Стек. Очередь. Реализация односвязных списков.
		Занятие 8. Динамические структуры данных. Стек. Очередь. Реализация двусвязных списков.
		Занятие 9. Рекурсия. Деревья.
	Традиционные 0	
СЛР 0,125 18 часов 0,5 ЗЕТ	Активные 0	Лабораторная работа № 1. Структура программы на Паскале. Основные типы данных.
		Лабораторная работа № 2. Обработка одномерных массивов.
		Лабораторная работа № 3. Обработка двумерных массивов.
		Лабораторная работа № 4. Использование процедур и функций.
		Лабораторная работа № 5. Обработка строковых и символьных переменных.
		Лабораторная работа № 6. Обработка числовых рядов.

		Лабораторная работа № 7. Сортировка массивов.
		Лабораторная работа № 8. Текстовые файлы.
		Лабораторная работа № 9. Файлы записей.
	Интерактивные 1	Лабораторная работа № 1. Структура программы на Паскале. Основные типы данных.
		Лабораторная работа № 2. Обработка одномерных массивов.
		Лабораторная работа № 3. Обработка двумерных массивов.
		Лабораторная работа № 4. Использование процедур и функций.
		Лабораторная работа № 5. Обработка строковых и символьных переменных.
		Лабораторная работа № 6. Обработка числовых рядов.
		Лабораторная работа № 7. Сортировка массивов.
		Лабораторная работа № 8. Текстовые файлы.
		Лабораторная работа № 9. Файлы записей.
	Традиционные 0	
КСР 0,125 18 часов 0,5 ЗЕТ	Активные 0	
	Интерактивные 0	
	Традиционные 0	
СРС 0,5 72 часов 2 ЗЕТ	Активные 0	
	Интерактивные 1	Реферат на тему: «Алгоритмы быстрой сортировки»
		Реферат на тему: «Динамические структуры»
		Реферат на тему: «Развитие современных устройств обработки информации»
	Традиционные 0	

3 Инновационные методы обучения

Для автоматизации учебного процесса преподавания курса «Информатика» на кафедре программных систем используется система дистанционного обучения (СДО), построенная на основе стандарт Moodle, где использовались широкие возможности системы для организации

тестирования промежуточных знаний студентов.

В рамках учебного курса с помощью системы СДО «Кафедра ПС» разработан комплект тестов, который позволяет преподавателю легко проверить как текущие, так и остаточные знания студентов. Автоматизированное тестирование студентов можно проводить во время занятий для того, чтобы оценить уровень их подготовки по заданному разделу. В качестве тестовых заданий можно задавать задачи небольшой трудоемкости, не требующие больших затрат на их решение.

4 Технические средства и материальное обеспечение учебного процесса

Компьютерный класс с IBM-совместимыми компьютерами, объединенными в локальную сеть с выходом в сеть Интернет.

5 Учебно-методическое обеспечение

5.1 Основная литература

1. Иванова, Галина Сергеевна. Основы программирования [Текст]: [учеб. по направлению "Информатика и вычисл. техника"] / Г. С. Иванова. - 4-е изд., стер. - М. : Изд-во МГТУ, 2007. - 415 с. - (Информатика в техническом университете). (10 экз.)
2. Калентьев, Анатолий Алексеевич. Основы информатики [Текст] : Учеб. пособие / А. А. Калентьев; Самар. гос. аэрокосм. ун-т им. С.П.Королева. - М.: [б.и.], 2003. (20 экз.)

5.2 Дополнительная литература

1. Королев, Лев Николаевич. Информатика. Введение в компьютерные науки : [учеб. для вузов] / Л. Н. Королев, А. И. Миков. - М. : Высш. шк., 2003. - 341 с. (2 экз.)
2. Информатика [Текст] : [учеб. пособие] / А. В. Могилев, Н. И. Пак, Е. К. Хеннер ; под ред. Е. К. Хеннера. - 3-е изд., перераб. и доп. - М. : Academia, 2004. - 841 с. - (Высшее профессиональное образование. Педагогические специальности). (3 экз.)
3. Окулов, Станислав Михайлович. Основы программирования : учебник / С. Окулов. - М. : Бином. Лаб. знаний , 2004. - 424 с. (3 экз.)
4. Баландин, Александр Васильевич. Курс лекций по информатике [Текст] : для студентов заоч. формы обучения / А. В. Баландин, Л. С. Зеленко, О. П. Солдатова ; Самар. гос. аэрокосм. ун-т им. С. П. Королева. - Самара : [б. и.], 1999. - 57 с. (175 экз.)

5.3 Электронные источники и интернет ресурсы

1. Электронный каталог НТБ СГАУ (lib.ssau.ru)
2. Научная электронная библиотека eLibrary (www.elibrary.ru)
3. Сопченко, Елена Вильевна. Разработка гипертекстовой среды обучения по курсу "Информатика" [Электронный ресурс] / Е. В. Сопченко, О. А. Дегтярева ; Федер. агентство по образованию, Самар. гос. аэрокосм. ун-т им. С. П. Королева, Нац. исслед. ун-т. - Электрон. текстовые дан. - Самара : [б. и.], 2009. - 1 эл. опт. диск (CD-RW). - (Программа развития государственного образовательного учреждения высшего профессионального образования "Самарский государственный аэрокосмический

университет имени академика С. П. Королева" на 2009-2018 годы. Полный доступ – электронный ресурс.

4. Симановский, Евгений Аркадьевич. Введение в информатику [Электронный ресурс] : [учеб. пособие] / Е. А. Симановский ; М-во образования и науки РФ, Самар. гос. аэрокосм. ун-т им. С. П. Королева (Нац. исслед. ун-т). - Электрон. дан. (1 файл : 988 Кбайт)
) . - Самара : Изд-во СГАУ, 2010. – Полный доступ – электронный ресурс.

5.4 Методические указания и рекомендации

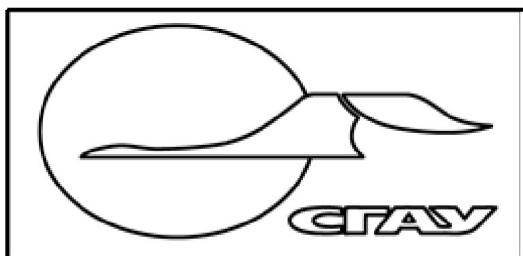
В процессе преподавания курса лекций основное внимание необходимо уделять разъяснению основных понятий курса.

Для оценки качества знаний необходимо в течение семестра проводить как промежуточное, так и окончательное тестирование.

При изложении теоретического материала желательно пользоваться иллюстративными материалами в виде слайдов или презентаций, чтобы повысить наглядность подачи материала и степень его запоминания.

Текущий контроль знаний студентов проводится на каждом практическом занятии в виде теоретического опроса студентов. По мере освоения тематических блоков дисциплины реализуется промежуточное тестирование уровня знаний студентов.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Самарский государственный аэрокосмический
университет имени академика С.П. Королёва
(национальный исследовательский университет)»



СОГЛАСОВАНО

УТВЕРЖДАЮ

Управление образовательных программ

Проректор по учебной работе

_____ / А.В. Дорошин /

_____ / Ф.В. Гречников /

" ____ " _____ 20__ г.

" ____ " _____ 20__ г.

РАБОЧАЯ ПРОГРАММА

Наименование модуля (дисциплины)

Информатика

Цикл, в рамках которого происходит освоение модуля (дисциплины)

С2. Математический и естественнонаучный цикл

Часть цикла

Базовая

Код учебного плана

090303.2.65-2011-О-П-5г00м

Факультет

6

Кафедра

Программных систем

Курс

1

Семестр

1

Лекции (СЛ)

36

Семинарские и практические занятия (СП)

18

Лабораторные занятия (СЛР)

18

Экзамен 1

Контроль самостоятельной работы (КСР)

0

Зачет

Самостоятельная работа (СРС)

54

Всего

126

Наименование стандарта, на основании которого составлена рабочая программа:
090303 Информационная безопасность автоматизированных систем (квалификация
(степень) «специалист»)

Соответствие содержания рабочей программы, условий ее реализации, материально-технической и учебно-методической обеспеченности учебного процесса по дисциплине всем требованиям государственных стандартов подтверждаем.

Составители:

Сопченко Елена Вильевна, к.т.н., доцент _____
(подпись)

Заведующий кафедрой:

Коварцев Александр Николаевич, д.т.н., _____
профессор (подпись)

Рабочая программа обсуждена на заседании кафедры
Программных систем

Протокол № ___ от " ___ " _____ 20___ г.

Наличие основной литературы в фондах научно-технической библиотеки (НТБ)
подтверждаем:

Директор НТБ _____ / _____ /
(подпись) (расшифровка подписи)

Согласовано:

Декан _____ / _____ /
(подпись) (расшифровка подписи)

1 Цели и задачи модуля (дисциплины), требования к уровню освоения содержания

1.1 Перечень развиваемых компетенций

ОК-1, ОК-4, ОК-5, ОК-6, ОК-11, ОК-12, ОК-15, – общекультурные компетенции
ПК-1, ПК-2 - профессиональная компетенция

1.2 Цели и задачи изучения модуля (дисциплины)

1. Ознакомление студентов с основами информатики;
2. Обучение студентов методам представления информации, построения алгоритмов и реализации алгоритмов на языках программирования;
3. Изучение фундаментальных основ информатики, часто применяемых в практической деятельности на вычислительных машинах.

1.3 Требования к уровню подготовки студента, завершившего изучение данного модуля (дисциплины)

Студенты, завершившие изучение данной дисциплины, должны знать:

1. Семантические аспекты информатики, основные методы расчета количества информации;
2. Структуру основных элементов компьютера;
3. Основы построения алгоритмических структур и их реализации на языках программирования высокого уровня;
4. Алгоритмы обработки базовых структур данных.

1.4 Связь с предшествующими модулями (дисциплинами)

Данная дисциплина является базовой, изучается в 1 семестре, поэтому не требует знания студентами дополнительных курсов.

1.5 Связь с последующими модулями (дисциплинами)

Курс «Информатика» относится к разделу общепрофессиональных дисциплин федерального компонента, материалы которого могут быть использованы в следующих курсах:

- Вычислительная математика;
- Логическое программирование;
- Моделирование информационных процессов и систем;
- Методы оптимизации и исследование операций.

2 Содержание рабочей программы (модуля)

Семестр 1		
СЛ 0,2857 36 часов 0,9999 ЗЕТ	Активные 1	Лекция 1. Информатика и ее связь с другими науками. Основные подходы к определению информации. Информация и данные.

		Лекция 2. Знаковые системы. Семиотические аспекты информатики. Адекватность информации. Меры информации.
		Лекция 3. Показатели качества информации. Классификация информации.
		Лекция 4. Сигнал: кодирование, декодирование. Информационные процессы и системы.
		Лекция 5. Структурная схема персонального компьютера. Основные устройства ПК и их назначение. Память. Понятия чтения-записи.
		Лекция 6. Представление информации в компьютере. Системы счисления.
		Лекция 7. Варианты представления информации в ПК. Кодирование текстовой, графической и звуковой информации.
		Лекция 8. Основные подходы к определению количества информации. Понятие энтропии. Формула Шеннона, формула Хартли.
		Лекция 9. Элементы алгебры логики. Основные логические функции. Законы преобразования логических выражений.
		Лекция 10. Методы работы с процедурной информацией. Постановка задачи. Моделирование информации.
		Лекция 11. Построение алгоритма решения задачи. Графическая форма представления алгоритмов. Пример реализации различных форм алгоритма.
		Лекция 12. Стадии разработки программного продукта. Характеристики программного продукта. Показатели качества программного продукта.
		Лекция 13. Программное обеспечение компьютера. Состав и назначение.
		Лекция 14. Базовые типы данных. Основные алгоритмы обработки.
		Лекция 15. Вычислительные сети. Локальные и глобальные вычислительные сети. Топологии сети. Адресация в сети.
		Лекция 16. Компьютерные вирусы и методы защиты от них.

		Лекция 17. Понятие несанкционированного доступа. Методы защиты информации. Криптографические методы защиты информации.
	Интерактивные 0	
	Традиционные 0	
СП 0,1429 18 часов 0,5001 ЗЕТ	Активные 0	
	Интерактивные 1	Занятие 1. Системы счисления, используемые в ПК.
		Занятие 2. Вычисление количества информации.
		Занятие 3. Преобразование логических выражений.
		Занятие 4. Построение алгоритмических структур.
		Занятие 5. Одномерные и двумерные массивы. Сортировка.
		Занятие 6. Символьный и строковый тип. Обработка.
		Занятие 7. Динамические структуры данных. Стек. Очередь. Реализация односвязных списков.
		Занятие 8. Динамические структуры данных. Стек. Очередь. Реализация двусвязных списков.
		Занятие 9. Рекурсия. Деревья.
	Традиционные 0	
СЛР 0,1429 18 часов 0,5001 ЗЕТ	Активные 0	Лабораторная работа № 1. Структура программы на Паскале. Основные типы данных.
		Лабораторная работа № 2. Обработка одномерных массивов.
		Лабораторная работа № 3. Обработка двумерных массивов.
		Лабораторная работа № 4. Использование процедур и функций.
		Лабораторная работа № 5. Обработка строковых и символьных переменных.
		Лабораторная работа № 6. Обработка числовых рядов.

		Лабораторная работа № 7. Сортировка массивов.
		Лабораторная работа № 8. Текстовые файлы.
		Лабораторная работа № 9. Файлы записей.
	Интерактивные 1	Лабораторная работа № 1. Структура программы на Паскале. Основные типы данных.
		Лабораторная работа № 2. Обработка одномерных массивов.
		Лабораторная работа № 3. Обработка двумерных массивов.
		Лабораторная работа № 4. Использование процедур и функций.
		Лабораторная работа № 5. Обработка строковых и символьных переменных.
		Лабораторная работа № 6. Обработка числовых рядов.
		Лабораторная работа № 7. Сортировка массивов.
		Лабораторная работа № 8. Текстовые файлы.
		Лабораторная работа № 9. Файлы записей.
	Традиционные 0	
КСР 0 0 часов 0 ЗЕТ	Активные 0	
	Интерактивные 0	
	Традиционные 0	
СРС 0,4286 54 часов 1,5001 ЗЕТ	Активные 0	
	Интерактивные 1	Реферат на тему: «Алгоритмы быстрой сортировки»
		Реферат на тему: «Динамические структуры»
		Реферат на тему: «Развитие современных устройств обработки информации»
	Традиционные 0	

3 Инновационные методы обучения

Для автоматизации учебного процесса преподавания курса «Информатика» на кафедре программных систем используется система дистанционного обучения (СДО), построенная на основе стандарт Moodle, где использовались широкие возможности системы для организации

тестирования промежуточных знаний студентов.

В рамках учебного курса с помощью системы СДО «Кафедра ПС» разработан комплект тестов, который позволяет преподавателю легко проверить как текущие, так и остаточные знания студентов. Автоматизированное тестирование студентов можно проводить во время занятий для того, чтобы оценить уровень их подготовки по заданному разделу. В качестве тестовых заданий можно задавать задачи небольшой трудоемкости, не требующие больших затрат на их решение.

4 Технические средства и материальное обеспечение учебного процесса

Компьютерный класс с IBM-совместимыми компьютерами, объединенными в локальную сеть с выходом в сеть Интернет.

5 Учебно-методическое обеспечение

5.1 Основная литература

1. Иванова, Галина Сергеевна. Основы программирования [Текст]: [учеб. по направлению "Информатика и вычисл. техника"] / Г. С. Иванова. - 4-е изд., стер. - М. : Изд-во МГТУ, 2007. - 415 с. - (Информатика в техническом университете). (10 экз.)
2. Калентьев, Анатолий Алексеевич. Основы информатики [Текст] : Учеб. пособие / А. А. Калентьев; Самар. гос. аэрокосм. ун-т им. С.П.Королева. - М.: [б.и.], 2003. (20 экз.)

5.2 Дополнительная литература

1. Королев, Лев Николаевич. Информатика. Введение в компьютерные науки : [учеб. для вузов] / Л. Н. Королев, А. И. Миков. - М. : Высш. шк., 2003. - 341 с. (2 экз.)
2. Информатика [Текст] : [учеб. пособие] / А. В. Могилев, Н. И. Пак, Е. К. Хеннер ; под ред. Е. К. Хеннера. - 3-е изд., перераб. и доп. - М. : Academia, 2004. - 841 с. - (Высшее профессиональное образование. Педагогические специальности). (3 экз.)
3. Окулов, Станислав Михайлович. Основы программирования : учебник / С. Окулов. - М. : Бином. Лаб. знаний , 2004. - 424 с. (3 экз.)
4. Баландин, Александр Васильевич. Курс лекций по информатике [Текст] : для студентов заоч. формы обучения / А. В. Баландин, Л. С. Зеленко, О. П. Солдатова ; Самар. гос. аэрокосм. ун-т им. С. П. Королева. - Самара : [б. и.], 1999. - 57 с. (175 экз.)

5.3 Электронные источники и интернет ресурсы

1. Электронный каталог НТБ СГАУ (lib.ssau.ru)
2. Научная электронная библиотека eLibrary (www.elibrary.ru)
3. Сопченко, Елена Вильевна. Разработка гипертекстовой среды обучения по курсу "Информатика" [Электронный ресурс] / Е. В. Сопченко, О. А. Дегтярева ; Федер. агентство по образованию, Самар. гос. аэрокосм. ун-т им. С. П. Королева, Нац. исслед. ун-т. - Электрон. текстовые дан. - Самара : [б. и.], 2009. - 1 эл. опт. диск (CD-RW). - (Программа развития государственного образовательного учреждения высшего профессионального образования "Самарский государственный аэрокосмический университет имени академика С. П. Королева" на 2009-2018

годы. Полный доступ – электронный ресурс.

4. Симановский, Евгений Аркадьевич. Введение в информатику [Электронный ресурс] : [учеб. пособие] / Е. А. Симановский ; М-во образования и науки РФ, Самар. гос. аэрокосм. ун-т им. С. П. Королева (Нац. исслед. ун-т). - Электрон. дан. (1 файл : 988 Кбайт). - Самара : Изд-во СГАУ, 2010. – Полный доступ – электронный ресурс.

5.4 Методические указания и рекомендации

В процессе преподавания курса лекций основное внимание необходимо уделять разъяснению основных понятий курса.

Для оценки качества знаний необходимо в течение семестра проводить как промежуточное, так и окончательное тестирование.

При изложении теоретического материала желательно пользоваться иллюстративными материалами в виде слайдов или презентаций, чтобы повысить наглядность подачи материала и степень его запоминания.

Текущий контроль знаний студентов проводится на каждом практическом занятии в виде теоретического опроса студентов. По мере освоения тематических блоков дисциплины реализуется промежуточное тестирование уровня знаний студентов.