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 160400.65

Электронный учебно-методический комплекс
по дисциплине в LMS Moodle

Работа выполнена по мероприятию блока 1 «Совершенствование
образовательной деятельности» Программы развития СГАУ
на 2009 – 2018 годы по проекту «Установка, настройка и использование в учебном процессе факультета
летательных аппаратов системы дистанционного обучения (СДО) Moodle совместно с блоком
«Электронный деканат»
Соглашение № 1/16 от 03.06.2013 г.

САМАРА
2013

УДК 681.3.06
И 741

Автор-составитель: **Стенгач Михаил Сергеевич**

Информатика 160400.65 [Электронный ресурс] : научно-образоват. модуль в системе дистанц. обучения Moodle / М-во образования и науки РФ, Самар. гос. аэрокосм. ун-т им. С. П. Королева (нац. исслед. ун-т); авт.-сост. М. С. Стенгач. - Электрон. текстовые и граф. дан. - Самара, 2013. – 1 эл. опт. диск (CD-ROM).

В состав учебно-методического комплекса входят:

1. Курс лекций.
2. Лабораторный практикум по информатике.
3. Тесты для контроля знаний.
4. Рабочая программа курса.

Научно-образовательный модуль «Информатика160400.65» предназначен для студентов факультета летательных аппаратов, обучающихся по специальности 160400.65 "Проектирование, производство и эксплуатация ракет и ракетно-космических комплексов" в 1 семестре.

Научно-образовательный модуль разработан на кафедре общей информатики.

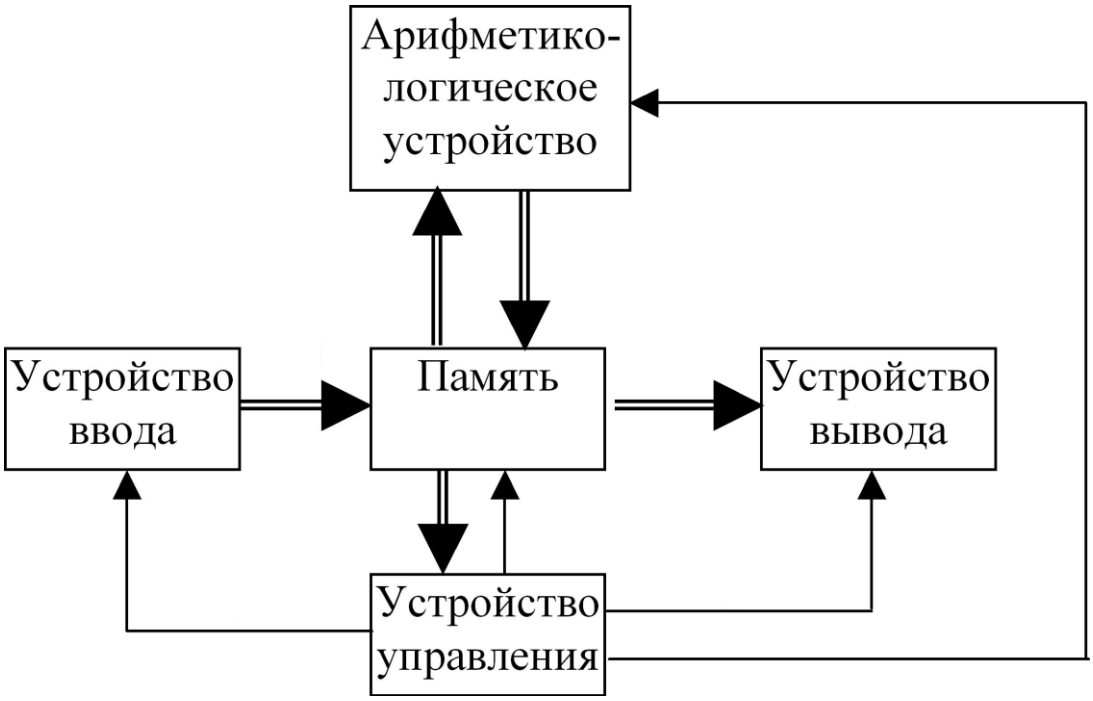
Курс лекций по информатике

Архитектура ЭВМ

Структура компьютера и принципы его функционирования.

В 1946 - 1948 годах в Принстонском университете (США) коллективом исследователей под руководством Джона фон Неймана был разработан проект ЭВМ, который никогда не был реализован, но идеи которого используются и по сей день. Этот проект получил название машины фон Неймана или Принстонской машины.

Структура Принстонской машины:



Несмотря на огромное разнообразие вычислительной техники и ее необычайно быстрое совершенствование, фундаментальные принципы устройства машин во многом остаются неизменными. В частности, начиная с самых первых поколений, любая ЭВМ состоит из следующих основных устройств: **процессор, память** (внутренняя и внешняя) и **устройства ввода и вывода** информации. Рассмотрим более подробно назначение каждого из них.

Процессор является главным устройством компьютера, в котором собственно и происходит обработка всех видов информации. Другой важной функцией процессора является обеспечение согласованного действия всех узлов, входящих в состав компьютера. Соответственно наиболее важными частями процессора являются **арифметико-логическое устройство АЛУ** и **устройство управления УУ**.

Каждый процессор способен выполнять вполне определенный набор универсальных инструкций, называемых чаще всего **машинными командами**. **Работа ЭВМ** состоит в выполнении последовательности таких команд, подготовленных в виде программы. Процессор способен организовать считывание очередной команды, ее анализ и выполнение, а также при необходимости принять данные или отправить результаты их обработки на требуемое устройство.

Хотя внутри процессора всегда имеются специальные ячейки (регистры) для оперативного хранения обрабатываемых данных и некоторой служебной информации, в нем сознательно не предусмотрено место для хранения программы. Для этой важной цели в компьютере служит другое устройство - **память.**

Память в целом **предназначена для** хранения, как данных, так и программ их обработки: согласно фундаментальному принципу фон Неймана, для обоих типов информации используется единое устройство.

Начиная с самых первых ЭВМ, память сразу стали делить на **внутреннюю** и **внешнюю.** Исторически это действительно было связано с размещением внутри или вне процессорного шкафа. Однако с уменьшением размеров машин внутрь основного процессорного корпуса удавалось поместить все большее количество устройств, и первоначальный непосредственный смысл данного деления постепенно утратился. Тем не менее, терминология сохранилась.

Под **внутренней памятью** современного компьютера принято понимать быстродействующую электронную память, расположенную на его системной плате. Наиболее существенная часть внутренней памяти называется **ОЗУ - оперативное запоминающее устройство.** Его главное назначение состоит в том, чтобы хранить данные и программы для решаемых в текущий момент задач. При выключении питания содержимое ОЗУ полностью теряется. В состав внутренней памяти современного компьютера помимо ОЗУ также входят и некоторые другие разновидности памяти, которые при первом знакомстве можно пропустить. Здесь упомянем только о **постоянном запоминающем устройстве (ПЗУ),** в котором в частности хранится информация, необходимая для первоначальной загрузки компьютера в момент включения питания. Как очевидно из названия, информация в ПЗУ не зависит от состояния компьютера (для лучшего понимания можно указать на некоторую аналогию между информацией в ПЗУ и "врожденными" безусловными рефлексами у живых существ). Раньше содержимое ПЗУ раз и навсегда формировалось на заводе, теперь же современные технологии позволяют, в случае необходимости, обновлять его, даже не извлекая из компьютерной платы.

Внешняя память реализуется в виде довольно разнообразных устройств хранения информации и обычно конструктивно оформляется в виде самостоятельных блоков. Сюда, прежде всего, следует отнести накопители на магнитных дисках (винчестерами), а также оптические дисководы (устройства для работы с CD ROM). В конструкции устройств внешней памяти имеются механически движущиеся части, поэтому скорость их работы существенно ниже, чем у полностью электронной внутренней памяти. Тем не менее, внешняя память позволяет сохранить огромные объемы информации с целью последующего использования.

Если процессор дополнить памятью, то такая система уже может быть работоспособной. Ее существенным недостатком является невозможность узнать что-либо о происходящем внутри такой системы. Для получения информации о результатах, необходимо дополнить компьютер **устройствами вывода,** которые позволяют представить их в доступной человеческому

восприятию форме. Наиболее распространенным устройством вывода является дисплей, способный быстро и оперативно отображать на своем экране как текстовую, так и графическую информацию. Для того чтобы получить копию результатов на бумаге, используют печатающее устройство, или принтер.

Наконец, поскольку пользователю часто требуется вводить в компьютерную систему новую информацию, необходимы еще и **устройства ввода.** Простейшим устройством ввода является клавиатура. Другие устройства ввода: манипулятор мышь, сканнер.

Основные положения теории информации. Понятие информации.

Термин "информация" происходит от латинского слова "informatio", что означает сведения, разъяснения, изложение. Несмотря на широкое распространение этого термина, понятие информации является одним из самых дискуссионных в науке. В настоящее время наука пытается найти общие свойства и закономерности, присущие многогранному понятию информация, но пока это понятие во многом остается интуитивным и получает различные смысловые наполнения в различных отраслях человеческой деятельности:

в обиходе информацией называют любые данные или сведения, которые кого-либо интересуют. Например, сообщение о каких-либо событиях, о чьей-либо деятельности и т.п. "Информировать" в этом смысле означает "сообщить нечто, неизвестное раньше";

в технике под информацией понимают сообщения, передаваемые в форме знаков или сигналов;

в кибернетике под информацией понимает ту часть знаний, которая используется для ориентирования, активного действия, управления, т.е. в целях сохранения, совершенствования, развития системы.

Клод Шеннон, американский учёный, заложивший основы теории информации — науки, изучающей процессы, связанные с передачей, приёмом, преобразованием и хранением информации, — рассматривает информацию как снятую неопределенность наших знаний о чем-то.

Современное научное представление об информации очень точно сформулировал Норберт Винер, "отец" кибернетики. А именно:

Информация — это обозначение содержания, полученного из внешнего мира в процессе нашего приспособления к нему и приспособления к нему наших чувств.

Люди обмениваются информацией в форме сообщений. Сообщение — это форма представления информации в виде речи, текстов, жестов, взглядов, изображений, цифровых данных, графиков, таблиц и т.п.

Одно и то же информационное сообщение (статья в газете, объявление, письмо, телеграмма, справка, рассказ, чертёж, радиопередача и т.п.) может содержать разное количество информации для разных людей — в зависимости от их предшествующих знаний, от уровня понимания этого сообщения и интереса к нему.

Так, сообщение, составленное на японском языке, не несёт никакой новой информации человеку, не знающему этого языка, но может быть высокоинформативным для человека, владеющего японским. Никакой новой информации не содержит и сообщение, изложенное на знакомом языке, если его содержание непонятно или уже известно.

Информация есть характеристика не сообщения, а соотношения между сообщением и его потребителем. Без наличия потребителя, хотя бы потенциального, говорить об информации бессмысленно.

В случаях, когда говорят об автоматизированной работе с информацией посредством каких-либо технических устройств, обычно в первую очередь интересуются не содержанием сообщения, а тем, сколько символов это сообщение содержит.

Применительно к компьютерной обработке данных под информацией понимают некоторую последовательность символических обозначений (букв, цифр, закодированных графических образов и звуков и т.п.), несущую смысловую нагрузку и представленную в понятном компьютеру виде. Каждый новый символ в такой последовательности символов увеличивает информационный объём сообщения.

Вид информации

Информация может существовать в виде:

- текстов, рисунков, чертежей, фотографий;
- световых или звуковых сигналов;
- радиоволн;
- электрических и нервных импульсов;
- магнитных записей;

Предметы, процессы, явления материального или нематериального свойства, рассматриваемые с точки зрения их информационных свойств, называются информационными объектами.

Передача информации

Информация передаётся в форме сообщений от некоторого источника информации к её приёмнику посредством канала связи между ними. Источник посылает передаваемое сообщение, которое кодируется в передаваемый сигнал. Этот сигнал посылается по каналу связи. В результате в приёмнике появляется принимаемый сигнал, который декодируется и становится принимаемым сообщением.

Пример:

Сообщение, содержащее информацию о прогнозе погоды, передаётся приёмнику (телезрителю) от источника — специалиста-метеоролога посредством канала связи — телевизионной передающей аппаратуры и телевизора.

Измерение информации

Какое количество информации содержится, к примеру, в тексте романа "Война и мир", во фресках Рафаэля или в генетическом коде человека? Ответа на эти вопросы наука не даёт и, по всей вероятности, даст не скоро. А возможно ли объективно измерить количество информации? Важнейшим результатом теории информации является следующий вывод:

«В определенных, весьма широких условиях можно пренебречь качественными особенностями информации, выразить её количество числом, а также сравнить количество информации, содержащейся в различных группах данных».

В настоящее время получили распространение подходы к определению понятия "количество информации", основанные на том, что информацию, содержащуюся в сообщении, можно нестрого трактовать в смысле её новизны или, иначе, уменьшения неопределённости наших знаний об объекте. Эти подходы используют математические понятия вероятности и логарифма.

В качестве единицы информации Клод Шеннон предложил принять один ***бит*** (англ. bit — binary digit — двоичная цифра).

Бит в теории информации — количество информации, необходимое для различения двух равновероятных сообщений (типа "орел"—"решка", "чет"—"нечет" и т.п.).

В вычислительной технике битом называют наименьшую "порцию" памяти компьютера, необходимую для хранения одного из двух знаков "0" и "1", используемых для внутримашинного представления данных и команд.

Бит — слишком мелкая единица измерения. На практике чаще применяется более крупная единица — байт, равная восьми битам. Именно восемь битов требуется для того, чтобы закодировать любой из 256 символов алфавита клавиатуры компьютера (256=28).

Широко используются также ещё более крупные производные единицы информации:

1 Килобайт (Кбайт) = 1024 байт = 2^{10} байт,

1 Мегабайт (Мбайт) = 1024 Кбайт = 2^{20} байт,

1 Гигабайт (Гбайт) = 1024 Мбайт = 2^{30} байт.

В последнее время в связи с увеличением объёмов обрабатываемой информации входят в употребление такие производные единицы, как:

1 Терабайт (Тбайт) = 1024 Гбайт = 2^{40} байт,

1 Петабайт (Пбайт) = 1024 Тбайт = 2^{50} байт.

Обработка информации

Обработка информации — получение одних информационных объектов из других информационных объектов путем выполнения некоторых алгоритмов.

Обработка является одной из основных операций, выполняемых над информацией, и главным средством увеличения объёма и разнообразия информации.

Средства обработки информации — это всевозможные устройства и системы, созданные человечеством, и в первую очередь, компьютер — универсальная машина для обработки информации.

Компьютеры обрабатывают информацию путем выполнения некоторых алгоритмов.

Алгоритмы

Алгоритм - четкое описание последовательности действий, которые необходимо выполнить при решении задачи. Можно сказать, что алгоритм описывает процесс преобразования исходных данных в результаты, т.к. для решения любой задачи необходимо:

1. Ввести исходные данные.
2. Преобразовать исходные данные в результаты (выходные данные).
3. Вывести результаты.

Разработка алгоритма решения задачи - это разбиение задачи на последовательно выполняемые этапы, причем результаты выполнения предыдущих этапов могут использоваться при выполнении последующих. При этом должны быть четко указаны как содержание каждого этапа, так и порядок выполнения этапов. Отдельный этап алгоритма представляет собой либо другую, более простую задачу, алгоритм решения которой известен (разработан заранее), либо должен быть достаточно простым и понятным без пояснений. Разработанный алгоритм можно записать несколькими способами:

- на естественном языке;
- в виде блок-схемы;

Рассмотрим пример алгоритма на естественном языке:

1. Ввести в компьютер числовые значения переменных **a, b и c**.
2. Вычислить d по формуле $\mathbf{d = b^2 - 4ac}$.
3. Если $\mathbf{d < 0}$, то напечатать сообщение "Корней нет" и перейти к п.4. Иначе вычислить и напечатать значения $\mathbf{x_1}$ **и** $\mathbf{x_2}$.
4. Прекратить вычисления.

Изображение алгоритма в виде блок-схемы

Блок-схемой называется наглядное графическое изображение алгоритма, когда отдельные его этапы изображаются при помощи различных геометрических фигур - блоков, а связи между этапами (последовательность выполнения этапов) указываются при помощи стрелок, соединяющих эти фигуры. Блоки сопровождаются надписями. Типичные действия алгоритма изображаются следующими геометрическими фигурами:

Блок начала-конца алгоритма. Надпись на блоке: "начало" ("конец").

Блок ввода-вывода данных. Надпись на блоке: слово "ввод" ("вывод" или "печать") и список вводимых (выводимых) переменных.

Блок начала-конца алгоритма

Блок ввода-вывода данных

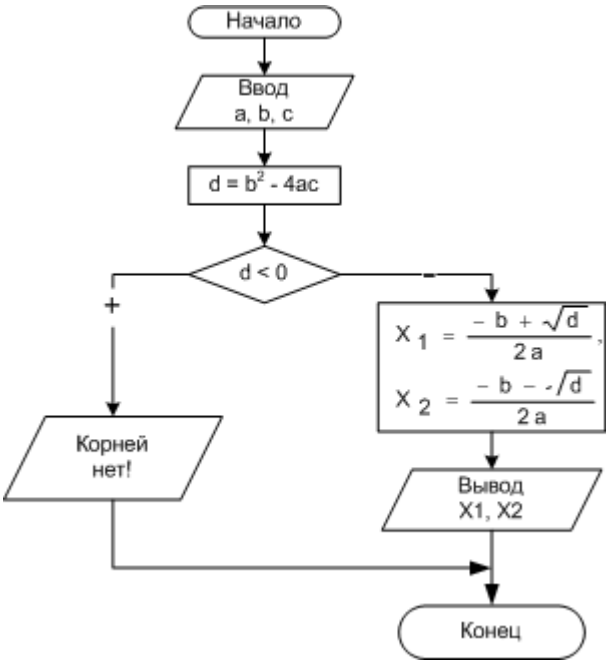
Блок решения или **арифметический .** Надпись на блоке: операция или группа операций.

Условный блок.Надпись на блоке: условие. В результате проверки условия осуществляется выбор одного из возможных путей (ветвей) вычислительного процесса. Если условие выполняется, то следующим выполняется этап по ветви "+", если условие не выполняется, то выполняется этап по ветви "–".

Арифметический блок

Условный блок

В качестве примера рассмотрим блок-схему алгоритма решения квадратного уравнения



Системы счисления

Система счисления — это совокупность приемов и правил, по которым числа записываются и читаются.

Существуют позиционные и непозиционные системы счисления.

В непозиционных системах счисления вес цифры (т. е. тот вклад, который она вносит в значение числа) **не зависит от ее позиции** в записи числа. Так, в римской системе счисления в числе XXXII (тридцать два) вес цифры X в любой позиции равен просто десяти.

В позиционных системах счисления вес каждой цифры изменяется в зависимости от ее положения (позиции) в последовательности цифр, изображающих число. Например, в числе 757,7 первая семерка означает 7 сотен, вторая — 7 единиц, а третья — 7 десятых долей единицы.

Сама же запись числа 757,7 означает сокращенную запись выражения

$$700 + 50 + 7 + 0{,}7 = 7 \cdot 10^2 + 5 \cdot 10^1 + 7 \cdot 10^0 + 7 \cdot 10^{-1} = 757{,}7.$$

Любая позиционная система счисления характеризуется своим **основанием.**

Основание позиционной системы счисления — количество различных цифр, используемых для изображения чисел в данной системе счисления.

За основание системы можно принять любое натуральное число — два, три, четыре и т.д.

Следовательно, **возможно бесчисленное множество позиционных систем:** двоичная, троичная, четверичная и т.д. Запись чисел в каждой из систем счисления с основанием q означает сокращенную запись выражения

$$a_{n-1}\,q^{n-1} + a_{n-2}\,q^{n-2} + \ldots + a_1\,q^1 + a_0\,q^0 + a_{-1}\,q^{-1} + \ldots + a_{-m}\,q^{-m},$$

где a_i — цифры системы счисления; n и m — число целых и дробных разрядов, соответственно.

Целые числа в позиционных системах счисления

В каждой системе счисления цифры упорядочены в соответствии с их значениями: 1 больше 0, 2 больше 1 и т.д.

Продвижением цифры называют замену её следующей по величине.

Продвинуть цифру 1 значит заменить её на 2, продвинуть цифру 2 значит заменить её на 3 и т.д.

Продвижение старшей цифры (например, цифры 9 в десятичной системе) **означает замену её на 0**. В двоичной системе, использующей только две цифры — 0 и 1, продвижение 0 означает замену его на 1, а продвижение 1 — замену её на 0.

Целые числа в любой системе счисления порождаются с помощью Правила счета:

Для образования целого числа, следующего за любым данным целым числом, нужно продвинуть самую правую цифру числа; если какая-либо цифра после продвижения стала нулем, то нужно продвинуть цифру, стоящую слева от неё.

Применяя это правило, запишем первые десять целых чисел

- в двоичной системе: 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001;
- в троичной системе: 0, 1, 2, 10, 11, 12, 20, 21, 22, 100;
- в пятеричной системе: 0, 1, 2, 3, 4, 10, 11, 12, 13, 14;
- в восьмеричной системе: 0, 1, 2, 3, 4, 5, 6, 7, 10, 11.

Системы счисления, используемые в компьютерах

Кроме десятичной широко используются системы с основанием, являющимся целой степенью числа 2, а именно:

- **двоичная** (используются цифры 0, 1);
- **восьмеричная** (используются цифры 0, 1, ..., 7);
- **шестнадцатеричная** (для первых целых чисел от нуля до девяти используются цифры 0, 1, ..., 9, а для следующих чисел — от десяти до пятнадцати — в качестве цифр используются символы A, B, C, D, E, F).

Из всех систем счисления особенно проста и поэтому интересна для технической реализации в компьютерах **двоичная система счисления.**

Люди предпочитают десятичную систему, вероятно, потому, что с древних времен считали по пальцам, а пальцев у людей по десять на руках и ногах. Не всегда и не везде люди пользуются десятичной системой счисления. В Китае, например, долгое время пользовались пятеричной системой счисления.

А компьютеры используют двоичную систему потому, что она имеет ряд преимуществ перед другими системами:

- для ее реализации нужны **технические устройства с двумя устойчивыми состояниями** (есть ток — нет тока, намагничен — не намагничен и т.п.), а не, например, с десятью, — как в десятичной;
- представление информации посредством только двух состояний **надежно** и **помехоустойчиво;**
- возможно **применение аппарата булевой алгебры** для выполнения логических преобразований информации;
- двоичная арифметика намного проще десятичной.

Двоичная система, удобная для компьютеров, для человека неудобна из-за ее громоздкости и непривычной записи.

Перевод чисел из десятичной системы в двоичную и наоборот выполняет машина. Однако, чтобы профессионально использовать компьютер, следует научиться понимать слово машины. Для этого и разработаны восьмеричная и шестнадцатеричная системы.

Числа в этих системах читаются почти так же легко, как десятичные, требуют соответственно в три (восьмеричная) и в четыре (шестнадцатеричная) раза меньше разрядов, чем в двоичной системе (ведь числа 8 и 16 — соответственно, третья и четвертая степени числа 2).

Перевод восьмеричных и шестнадцатеричных чисел в двоичную систему очень прост: достаточно каждую цифру заменить эквивалентной ей двоичной триадой (тройкой цифр) или тетрадой (четверкой цифр).

Чтобы перевести число из двоичной системы в восьмеричную или шестнадцатеричную, его нужно разбить влево и вправо от запятой на триады (для восьмеричной) или тетрады (для шестнадцатеричной) и каждую такую группу заменить соответствующей восьмеричной (шестнадцатеричной) цифрой.

Для перевода целого десятичного числа N в систему счисления с основанием q необходимо N разделить с остатком ("нацело") на q , записанное в той же десятичной системе. Затем неполное частное, полученное от такого деления, нужно снова разделить с остатком на q , и т.д., пока последнее полученное неполное частное не станет равным нулю. Представлением числа N в новой системе счисления будет последовательность остатков деления, изображенных одной q-ичной цифрой и записанных в порядке, обратном порядку их получения.

Перевод в десятичную систему числа x, записанного в q-ичной системе счисления (q = 2, 8 или 16) в виде $x_q = (a_n a_{n-1} \ \dots \ a_0 \ , \ a_{-1} \ a_{-2} \ \dots \ a_{-m})_q$ сводится к вычислению значения многочлена

$$x_{10} = a_n \ q^n + \ a_{n-1} \ q^{n-1} \ + \ \dots \ + \ a_0 \ q^0 \ + \ a_{-1} \ q^{-1} \ + \ a_{-2} \ q^{-2} \ + \ \dots \ + \ a_{-m} \ q^{-m}$$

средствами десятичной арифметики.

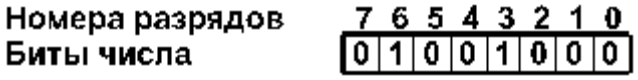
Представление в компьютере целых чисел

Целые числа могут представляться в компьютере со знаком или без знака.

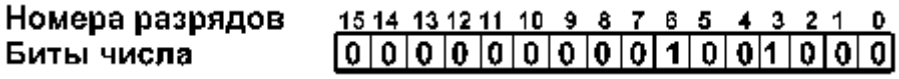
Целые числа без знака обычно занимают в памяти компьютера один или два байта. В однобайтовом формате принимают значения от 00000000_2 до 11111111_2. В двубайтовом формате — от $00000000\ 00000000_2$ до $11111111\ 11111111_2$.

Примеры:

а) число $72_{10} = 1001000_2$ в **однобайтовом** формате:



б) это же число в **двубайтовом** формате:



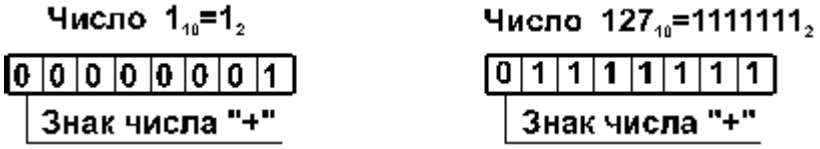
Целые числа со знаком обычно занимают в памяти компьютера один, два или четыре байта, при этом самый левый (старший) разряд содержит информацию о знаке числа.

Рассмотрим особенности записи целых чисел со знаком на примере **однобайтового формата**, при котором для знака отводится один разряд, а для цифр абсолютной величины – семь разрядов.

В компьютерной технике применяются три формы записи (кодирования) целых чисел со знаком: **прямой код, обратный код, дополнительный код.**

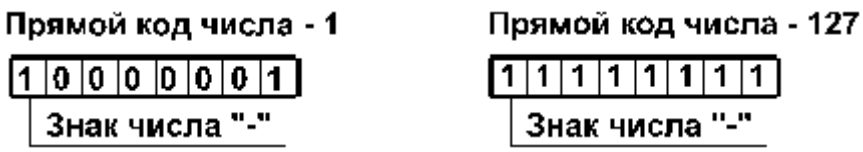
Последние две формы применяются особенно широко, так как позволяют упростить конструкцию арифметико-логического устройства компьютера путем замены разнообразных арифметических операций операцией сложения.

Положительные числа в прямом, обратном и дополнительном кодах изображаются одинаково — двоичными кодами с цифрой 0 в знаковом разряде. Например:



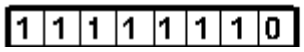
Отрицательные числа в прямом, обратном и дополнительном кодах имеют разное изображение.

1. **Прямой код.** В знаковый разряд помещается цифра 1, а в разряды цифровой части числа — двоичный код его абсолютной величины. Например:

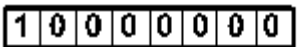


2. **Обратный код.** Получается инвертированием всех цифр двоичного кода абсолютной величины числа, включая разряд знака: нули заменяются единицами, а единицы — нулями. Например:

Число: -1
Код модуля числа: 0 0000001
Обратный код числа: 1 1111110

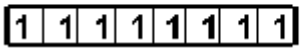


Число: -127
Код модуля числа: 0 1111111
Обратный код числа: 1 0000000

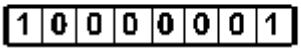


3. **Дополнительный код.** Получается образованием обратного кода с последующим прибавлением единицы к его младшему разряду. Например:

Дополнительный код числа - 1



Дополнительный код числа - 127



Обычно **отрицательные десятичные числа при вводе в машину автоматически преобразуются в обратный или дополнительный двоичный код** и в таком виде хранятся, перемещаются и участвуют в операциях. При выводе таких чисел из машины происходит **обратное преобразование в отрицательные десятичные числа.**

Компьютерная сеть

Компьютерная сеть (англ. Computer NetWork, от net — сеть и work — работа) — совокупность компьютеров, соединенных с помощью каналов связи и средств коммутации в единую систему для обмена сообщениями и доступа пользователей к программным, техническим, информационным и организационным ресурсам сети.

Компьютерную сеть представляют как совокупность **узлов** (компьютеров и сетевого оборудования) и соединяющих их **ветвей** (каналов связи). **Ветвь сети** — это путь, соединяющий два смежных узла. Различают узлы **оконечные,** расположенные в конце только одной ветви, **промежуточные,** расположенные на концах более чем одной ветви, и **смежные** — такие узлы соединены по крайней мере одним путём, не содержащим никаких других узлов. Компьютеры могут объединяться в сеть разными способами.

Логический и физический способы соединения компьютеров, кабелей и других компонентов, в целом составляющих сеть, называется ее топологией. Топология характеризует свойства сетей, не зависящие от их размеров. При этом не учитывается производительность и принцип работы этих объектов, их типы, длины каналов, хотя при проектировании эти факторы очень важны.

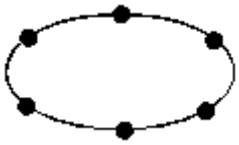
СПРАВКА. Топология как математическое понятие:

Топология (от греч. topos — место и ... логия), раздел математики, изучающий топологические свойства фигур, т. е. свойства, не изменяющиеся при любых деформациях, производимых без разрывов и склеиваний. Примерами топологических свойств фигур являются размерность, число кривых , ограничивающих данную область и т. д. Так, окружность, эллипс, контур квадрата имеют одни и те же топологические свойства, т. к. эти линии могут быть деформированы одна в другую описанным выше образом; в то же время кольцо и круг обладают различными топологическими свойствами: круг ограничен одним контуром, а кольцо —двумя.

(Советский энциклопедический словарь. "Советская энциклопедия", 1979).

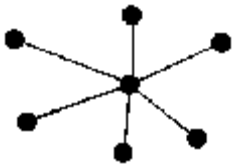
Наиболее распространенные виды топологий сетей:

Линейная сеть. Содержит только два оконечных узла, любое число промежуточных узлов и имеет только один путь между любыми двумя узлами.

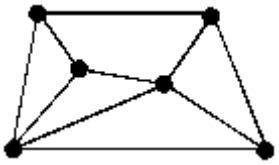


Кольцевая сеть. Сеть, в которой к каждому узлу присоединены две и только две ветви.

Древовидная сеть. Сеть, которая содержит более двух оконечных узлов и по крайней мере два промежуточных узла, и в которой между двумя узлами имеется только один путь.



Звездообразная сеть. Сеть, в которой имеется только один промежуточный узел.



Ячеистая сеть. Сеть, которая содержит по крайней мере два узла, имеющих два или более пути между ними.

Полносвязанная сеть. Сеть, в которой имеется ветвь между любыми двумя узлами. Важнейшая характеристика компьютерной сети — её архитектура.

Архитектура сети — это реализованная структура сети передачи данных, определяющая её топологию, состав устройств и правила их взаимодействия в сети. В рамках архитектуры сети рассматриваются вопросы кодирования информации, её адресации и передачи, управления потоком сообщений, контроля ошибок и анализа работы сети в аварийных ситуациях и при ухудшении характеристик.

Наиболее распространённые архитектуры:

- **Ethernet** (англ. ether — эфир) — широковещательная сеть. Это значит, что все станции сети могут принимать все сообщения. Топология — линейная или звездообразная. Скорость передачи данных 10 или 100 Мбит/сек.
- **Arcnet** (Attached Resource Computer Network — компьютерная сеть соединённых ресурсов) — широковещательная сеть. Физическая топология — дерево. Скорость передачи данных 2,5 Мбит/сек.

- **Token Ring** (эстафетная кольцевая сеть, сеть с передачей маркера) — кольцевая сеть, в которой принцип передачи данных основан на том, что каждый узел кольца ожидает прибытия некоторой короткой уникальной последовательности битов — маркера — из смежного предыдущего узла. Поступление маркера указывает на то, что можно передавать сообщение из данного узла дальше по ходу потока. Скорость передачи данных 4 или 16 Мбит/сек.
- **FDDI** (Fiber Distributed Data Interface) — сетевая архитектура высокоскоростной передачи данных по оптоволоконным линиям. Скорость передачи — 100 Мбит/сек. Топология — двойное кольцо или смешанная (с включением звездообразных или древовидных подсетей). Максимальное количество станций в сети — 1000. Очень высокая стоимость оборудования.
- **ATM** (Asynchronous Transfer Mode) — перспективная, пока ещё очень дорогая архитектура, обеспечивает передачу цифровых данных, видеоинформации и голоса по одним и тем же линиям. Скорость передачи до 2,5 Гбит/сек. Линии связи оптические.

Для соединения между собой устройств сети используется специальное оборудование:

- **Сетевые кабели** (**коаксиальные**, состоящие из двух изолированных между собой концентрических проводников, из которых внешний имеет вид трубки; **оптоволоконные**; кабели на **витых парах**, образованные двумя переплетёнными друг с другом проводами, и др.).
- **Коннекторы** (соединители) для подключения кабелей к компьютеру; **разъёмы** для соединения отрезков кабеля.
- **Сетевые интерфейсные адаптеры** для приёма и передачи данных. В соответствии с определённым протоколом управляют доступом к среде передачи данных. Размещаются в системных блоках компьютеров, подключенных к сети. К разъёмам адаптеров подключается сетевой кабель.
- **Трансиверы** повышают уровень качества передачи данных по кабелю, отвечают за приём сигналов из сети и обнаружение конфликтов.
- **Хабы** (концентраторы) и **коммутирующие хабы** (коммутаторы) расширяют топологические, функциональные и скоростные возможности компьютерных сетей. Хаб с набором разнотипных портов позволяет **объединять сегменты сетей с различными кабельными системами.** К порту хаба можно подключать как отдельный узел сети, так и другой хаб или сегмент кабеля.
- **Повторители** (репитеры) усиливают сигналы, передаваемые по кабелю при его большой длине.

Классификация компьютерных сетей:

Локальная сеть (ЛВС или LAN — Local Area NetWork) — сеть, связывающая ряд компьютеров в зоне, ограниченной пределами одной комнаты, здания или предприятия.

Глобальная сеть (ГВС или WAN — World Area NetWork) — сеть, соединяющая компьютеры, удалённые географически на большие расстояния друг от друга. Отличается от локальной сети более протяженными коммуникациями (спутниковыми, кабельными и др.). Глобальная сеть объединяет локальные сети.

Городская сеть (MAN — Metropolitan Area NetWork) — сеть, которая обслуживает информационные потребности большого города.

Для соединения локальных сетей используются следующие устройства, которые различаются между собой по назначению и возможностям:

- **Мост** (англ. Bridge) — связывает две локальные сети. **Передаёт данные между сетями в пакетном виде, не производя в них никаких изменений.** Ниже на рисунке показаны три локальные сети, соединённые двумя мостами.

Здесь мосты создали **расширенную сеть,** которая обеспечивает своим пользователям **доступ к прежде недоступным ресурсам.** Кроме этого, мосты могут **фильтровать пакеты**, охраняя всю сеть от локальных потоков данных и пропуская наружу только те данные, которые предназначены для других сегментов сети.

- **Маршрутизатор** (англ. Router) объединяет сети с общим протоколом более эффективно, чем мост. Он позволяет, например, расщеплять большие сообщения на более мелкие куски, обеспечивая тем самым взаимодействие локальных сетей с разным размером пакета. Маршрутизатор может пересылать пакеты на конкретный адрес (мосты только отфильтровывают ненужные пакеты), выбирать лучший путь для прохождения пакета и многое другое. Чем сложней и больше сеть, тем больше выгода от использования маршрутизаторов.
- **Мостовой маршрутизатор** (англ. Brouter) — это гибрид моста и маршрутизатора, который сначала пытается выполнить маршрутизацию, где это только возможно, а затем, в случае неудачи, переходит в режим моста.
- **Шлюз** (англ. GateWay), в отличие от моста, применяется в случаях, когда соединяемые сети имеют **различные сетевые протоколы.** Поступившее в шлюз сообщение от одной сети преобразуется в другое сообщение, соответствующее требованиям следующей сети. Таким образом, шлюзы не просто соединяют сети, а позволяют им работать как единая сеть. С помощью шлюзов также локальные сети подсоединяются к **мэйнфреймам** — универсальным мощным компьютерам.

Беспроводные сети используются там, где прокладка кабелей затруднена, нецелесообразна или просто невозможна. Например, в исторических зданиях, промышленных помещениях с металлическим или железобетонным полом, в офисах, полученных в краткосрочную аренду, на складах, выставках, конференциях и т.п.

В этих случаях сеть реализуется при помощи **сетевых радио-адаптеров,** снабжённых **всенаправленными антеннами** и использующих в качестве среды передачи информации

радиоволны. Такая сеть реализуется топологией "**Все-Со-Всеми**" и работоспособна при дальности 50-200 м.

Для связи между беспроводной и кабельной частями сети используется специальное устройство, называемое **точкой входа** (или **радиомостом**). Можно использовать и обычный компьютер, в котором установлены два сетевых адаптера — **беспроводной** и **кабельный.**

Другой важной областью применения беспроводных сетей является **организация связи между удалёнными сегментами локальных сетей при отсутствии инфраструктуры передачи данных** (кабельных сетей общего доступа, высококачественных телефонных линий и др.), что типично для нашей страны. В этом случае для наведения беспроводных мостов между двумя удалёнными сегментами используются **радиомосты с антенной направленного типа.**

Если в сеть нужно объединить **несколько сегментов**, то используется топология типа "**звезда**". При этом в **центральном** узле устанавливается **всенаправленная антенна, а удалённых** узлах — **направленные.** Сети звездообразной топологии могут образовывать сети разнообразной конфигурации.

Сетевая магистраль с беспроводным доступом позволяет отказаться от использования медленных модемов.

Интернет

Интернет — гигантская всемирная компьютерная сеть, объединяющая десятки тысяч сетей всего мира. Её назначение — обеспечить любому желающему постоянный доступ к любой информации. Интернет предлагает практически неограниченные информационные ресурсы, полезные сведения, учёбу, развлечения, возможность общения с компетентными людьми, услуги удалённого доступа, передачи файлов, электронной почты и многое другое. Интернет обеспечивает принципиально новый способ общения людей, не имеющий аналогов в мире.

Благодаря сети стал доступен (бесплатно или за умеренную плату) огромный объём информации. Так, пользователь в любой стране может связаться с людьми, разделяющими его интересы, или получить ценные сведения в электронных библиотеках, даже если они находятся на другом конце света. Нужная информация окажется в его компьютере за считанные секунды, пройдя путь по длинной цепочке промежуточных компьютеров, по кабелям и по радио, через горы и моря, по дну океана и через спутник.

Интернет финансируется правительствами, научными и образовательными учреждениями, коммерческими структурами и миллионами частных лиц во всех частях света, но никто конкретно не является её владельцем. Управляет сетью **"Совет по архитектуре Интернет",** формируемый из приглашённых добровольцев.

Сеть была создана в 1984 году, и сейчас ею пользуются примерно сорок миллионов человек. Интернет всё время изменяется, поскольку имеет много квалифицированных пользователей, которые пишут программы для себя, а затем распространяют их среди желающих. Постоянно

появляются новые серверы, а существующие обновляют свой "репертуар". Стремительно растут информационные потоки.

Отдельные участки Интернет представляют собой сети различной архитектуры, которые связываются между собой с помощью **маршрутизаторов.** Передаваемые данные разбиваются на небольшие порции, называемые **пакетами**. Каждый пакет перемещается по сети независимо от других пакетов. Сети в Интернет **неограниченно коммутируются (т.е. связываются) друг с другом**, потому что все компьютеры, участвующие в передаче данных, используют единый протокол коммуникации **TCP/IP** (читается "ти-си-пи / ай-пи"). На самом деле протокол TCP/IP — это два разных протокола, определяющих различные аспекты передачи данных в сети:

- **протокол TCP** (Transmission Control Protocol) — протокол управления передачей данных, использующий автоматическую повторную передачу пакетов, содержащих ошибки; этот протокол отвечает за разбиение передаваемой информации на пакеты и правильное восстановление информации из пакетов получателя;
- **протокол IP** (Internet Protocol) — протокол межсетевого взаимодействия, отвечающий за адресацию и позволяющий пакету на пути к конечному пункту назначения проходить по нескольким сетям.

Схема передачи информации по протоколу TCP/IP такова: протокол TCP разбивает информацию на пакеты и нумерует все пакеты; далее с помощью протокола IP все пакеты передаются получателю, где с помощью протокола TCP проверяется, все ли пакеты получены; после получения всех пакетов протокол TCP располагает их в нужном порядке и собирает в единое целое.

Каждый компьютер, подключенный к сети Интернет имеет два равноценных уникальных адреса: **цифровой IP-адрес и символический доменный адрес.** Присваивание адресов происходит по следующей схеме: международная организация Сетевой информационный центр выдает группы адресов владельцам локальных сетей, а последние распределяют конкретные адреса по своему усмотрению.

IP-адрес компьютера имеет длину 4 байта. Обычно первый и второй байты определяют **адрес сети**, третий байт определяет **адрес подсети**, а четвертый — **адрес компьютера в подсети**. Для удобства IP-адрес записывают в виде четырех чисел со значениями от 0 до 255, разделенных точками, например: 145.37.5.150. Адрес сети — 145.37; адрес подсети — 5; адрес компьютера в подсети — 150.

Доменный адрес (англ. domain — область), в отличие от цифрового, является символическим и легче запоминается человеком. Пример доменного адреса: **barsuk.les.nora.ru**. Здесь домен barsuk — имя реального компьютера, обладающего IP-адресом, домен les — имя группы, присвоившей имя этому компьютеру, домен nora — имя более крупной группы, присвоившей имя домену les , и т.д. В процессе передачи данных доменный адрес преобразуются в IP-адрес.

World Wide Web — главный информационный сервис интернета.

World Wide Web (WWW, "Всемирная паутина") — гипертекстовая, а точнее, гипермедийная информационная система поиска ресурсов Интернет и доступа к ним.

Гипертекст — информационная структура, позволяющая устанавливать смысловые связи между элементами текста на экране компьютера таким образом, чтобы можно было легко осуществлять переходы от одного элемента к другому. На практике в гипертексте некоторые слова выделяют путем подчёркивания или **окрашивания в другой цвет.** Выделение слова говорит о наличии связи этого слова с некоторым документом, в котором тема, связанная с выделенным словом, рассматривается более подробно.

Гипермедиа — это то, что получится, если в определении гипертекста заменить слово "текст" на "любые виды информации": звук, графику, видео. Такие гипермедийные ссылки возможны, поскольку наряду с текстовой информацией можно связывать и любую другую двоичную информацию, например, закодированный звук или графику, Так, если программа отображает карту мира и если пользователь выбирает на этой карте с помощью мыши какой-либо континент, программа может тут же дать о нём графическую, звуковую и текстовую информацию.

Система WWW построена на специальном протоколе передачи данных, который называется **протоколом передачи гипертекста HTTP** (HyperText Transfer Protocol). Всё содержимое системы WWW состоит из **WWW-страниц.**

WWW-страницы — гипермедийные документы системы World Wide Web. Создаются с помощью языка разметки гипертекста HTML (Hypertext markup language).

Язык HTML позволяет добавлять к текстовым документам специальные командные фрагменты — **тэги** (англ. **tag** — "этикетка, ярлык") таким образом, что становится возможным связывать с этими документами другие тексты, графику, звук и видео, задавать заголовки различных уровней, разделять текст на абзацы, строить таблицы и т.д.

Одну WWW-страницу на самом деле обычно составляет **набор гипермедийных документов, расположенных на одном сервере, переплетённых взаимными ссылками и связанных по смыслу** (например, содержащих информацию об одном учебном заведении или об одном музее). Каждый документ страницы, в свою очередь, может содержать несколько экранных страниц текста и иллюстраций. Каждая WWW-страница имеет свой "титульный лист" (англ. "homepage") — гипермедийный документ, содержащий ссылки на главные составные части страницы. Адреса "титульных листов" распространяются в Интернет в качестве адресов страниц.

При работе с системой WWW пользователи имеют дело с **программами-клиентами системы, называемыми браузерами.**

Браузеры (англ. browse — листать, просматривать) — программы, с помощью которых пользователь организует диалог с системой WWW: просматривает WWW страницы, взаимодействует с WWW-серверами и другими ресурсами в Интернет.

Существуют сотни программ-браузеров. Самые популярные браузеры: Netscape Navigator и Microsoft Internet Explorer. Браузеры WWW умеют взаимодействовать с любыми типами серверов,

используя при этом их собственные протоколы. Информацию, полученную от любого сервера, браузер WWW выводит на экран в стандартной, удобной для восприятия форме. При этом переключения с одного протокола на другой для пользователя часто остаются незаме

Алгебра логики

Алгебра логики — это раздел математики, изучающий высказывания, рассматриваемые со стороны их логических значений (истинности или ложности) и логических операций над ними. Алгебра логики возникла в середине XIX века в трудах английского математика **Джорджа Буля**. Ее создание представляло собой попытку решать традиционные логические задачи алгебраическими методами.

Логическое высказывание — это любое повествовательное предложение, в отношении которого можно однозначно сказать, истинно оно или ложно.

Так, например, предложение "*6 — четное число*" следует считать высказыванием, так как оно истинное. Предложение "*Рим — столица Франции*" тоже высказывание, так как оно ложное.

Разумеется, **не всякое предложение является логическим высказыванием**. Высказываниями не являются, например, предложения "*ученик десятого класса*" и "*информатика — интересный предмет*". Первое предложение ничего не утверждает об ученике, а второе использует слишком неопределённое понятие "*интересный предмет*". Вопросительные и восклицательные предложения также не являются высказываниями, поскольку говорить об их истинности или ложности не имеет смысла.

Предложения типа "*в городе **A** более миллиона жителей*"**,** "*у него голубые глаза*" не являются высказываниями, так как для выяснения их истинности или ложности нужны дополнительные сведения: о каком конкретно городе или человеке идет речь. Такие предложения называются *высказывательными формами***.**

Высказывательная форма — это повествовательное предложение, которое прямо или косвенно содержит хотя бы одну переменную и становится высказыванием, когда все переменные замещаются своими значениями.

Алгебра логики рассматривает любое высказывание только с одной точки зрения — является ли оно истинным или ложным. Заметим, что **зачастую трудно установить истинность высказывания**. Так, например, высказывание "*площадь поверхности Индийского океана равна 75 млн кв. км*" в одной ситуации можно посчитать ложным, а в другой — истинным. Ложным — так как указанное значение неточное и вообще не является постоянным. Истинным — если рассматривать его как некоторое приближение, приемлемое на практике.

Употребляемые в обычной речи слова и словосочетания **"не", "и", "или", "если... , то", "тогда и только тогда"** и другие позволяют из уже заданных высказываний строить новые высказывания. Такие слова и словосочетания называются **логическими связками.**

Высказывания, образованные из других высказываний с помощью логических связок, называются **составными.** Высказывания, не являющиеся составными, называются **элементарными.**

Так, например, из элементарных высказываний "*Петров — врач*", "*Петров — шахматист*" при помощи связки "*и*" можно получить составное высказывание "*Петров — врач и шахматист*", понимаемое как "*Петров — врач, хорошо играющий в шахматы*".

При помощи связки "*или*" из этих же высказываний можно получить составное высказывание "*Петров — врач или шахматист*", понимаемое в алгебре логики как "*Петров или врач, или шахматист, или и врач и шахматист одновременно*".

Истинность или ложность получаемых таким образом составных высказываний зависит от истинности или ложности элементарных высказываний.

Чтобы обращаться к логическим высказываниям, им назначают имена. Пусть через **A** обозначено высказывание *"Тимур поедет летом на море",* а через **B** — высказывание *"Тимур летом отправится в горы"*. Тогда составное высказывание *"Тимур летом побывает и на море, и в горах"* можно кратко записать как **A и B**. Здесь **"и"** — логическая связка, **A, B** — логические переменные, которые могут принимать только два значения — "истина" или "ложь", обозначаемые, соответственно, "1" и "0".

Каждая логическая связка рассматривается как операция над логическими высказываниями и имеет свое название и обозначение:

НЕ Операция, выражаемая словом *"не",* называется **отрицанием** и обозначается чертой над высказыванием (или знаком ¬). Высказывание $\overline{A}$ истинно, когда A ложно, и ложно, когда A истинно. Пример. "*Луна — спутник Земли*" (A); "*Луна — не спутник Земли*" ($\overline{A}$).

И Операция, выражаемая связкой *"и",* называется **конъюнкцией** (лат. conjunctio — соединение) или логическим умножением и обозначается точкой "$\cdot$" (может также обозначаться знаками $\wedge$ или **&**). Высказывание **A** $\cdot$ **B** истинно тогда и только тогда, когда оба высказывания **A** и **B** истинны. Например, высказывание *"10 делится на 2 и 5 больше 3"* истинно, а высказывания *"10 делится на 2 и 5 не больше 3", "10 не делится на 2 и 5 больше 3", "10 не делится на 2 и 5 не больше 3"* — ложны.

ИЛИ Операция, выражаемая связкой *"или"* (в неисключающем смысле этого слова), называется **дизъюнкцией** (лат. disjunctio — разделение) или логическим сложением и обозначается знаком **v** (или плюсом). Высказывание **A v B** ложно тогда и только тогда, когда оба высказывания A и B ложны. Например, высказывание *"10 не делится на 2 или 5 не больше 3"* ложно, а высказывания *"10 делится на 2 или 5 больше 3", "10 делится на 2 или 5 не больше 3", "10 не делится на 2 или 5 больше 3"* — истинны.

ЕСЛИ-ТО Операция, выражаемая связками *"если ..., то", "из ... следует", "... влечет ...",* называется **импликацией** (лат. *implico* — тесно связаны) и обозначается знаком $\rightarrow$. Высказывание $A \rightarrow B$ ложно тогда и только тогда, когда **A** истинно, а **B** ложно.

Каким же образом импликация связывает два элементарных высказывания? Покажем это на примере высказываний: *"данный четырёхугольник — квадрат"* (**A**) и *"около данного четырёхугольника можно описать окружность"* (**B**). Рассмотрим составное высказывание $A \rightarrow B$

, понимаемое как *"если данный четырёхугольник квадрат, то около него можно описать окружность"*. Есть **три варианта,** когда высказывание $\mathbf{A \to B}$ истинно:

1. **A** истинно и **B** истинно, то есть данный четырёхугольник квадрат, и около него можно описать окружность;
2. **A** ложно и **B** истинно, то есть данный четырёхугольник не является квадратом, но около него можно описать окружность (разумеется, это справедливо не для всякого четырёхугольника);
3. **A** ложно и **B** ложно, то есть данный четырёхугольник не является квадратом, и около него нельзя описать окружность.

Ложен только один вариант, когда A истинно, а B ложно, то есть данный четырёхугольник является квадратом, но около него нельзя описать окружность.

В обычной речи связка *"если ..., то"* описывает причинно-следственную связь между высказываниями. Но в логических операциях смысл высказываний не учитывается. Рассматривается только их истинность или ложность. Поэтому не надо смущаться "бессмысленностью" импликаций, образованных высказываниями, совершенно не связанными по содержанию. Например, такими: *"если президент США — демократ, то в Африке водятся жирафы", "если арбуз — ягода, то в бензоколонке есть бензин"*.

РАВНОСИЛЬНО Операция, выражаемая связками "*тогда и только тогда*"**,** "*необходимо и достаточно*"**,** "*... равносильно ...*", называется **эквиваленцией** или двойной импликацией и обозначается знаком $\Leftrightarrow$ или **~.** Высказывание $\mathbf{A \leftrightarrow B}$ истинно тогда и только тогда, когда значения **A** и **B** совпадают. Например, высказывания *"24 делится на 6 тогда и только тогда, когда 24 делится на 3", "23 делится на 6 тогда и только тогда, когда 23 делится на 3"* истинны, а высказывания *"24 делится на 6 тогда и только тогда, когда 24 делится на 5", "21 делится на 6 тогда и только тогда, когда 21 делится на 3"* ложны.

Высказывания **A** и **B,** образующие составное высказывание $\mathbf{A \leftrightarrow B}$, могут быть совершенно не связаны по содержанию, например: *"три больше двух"* (**A**), *"пингвины живут в Антарктиде"* (**B**). Отрицаниями этих высказываний являются высказывания *"три не больше двух"* ($\overline{\mathbf{A}}$), *"пингвины не живут в Антарктиде"* ($\overline{\mathbf{B}}$). Образованные из высказываний **A** и **B** составные высказывания $\mathbf{A \Leftrightarrow B}$ и $\overline{\mathbf{A}} \Leftrightarrow \overline{\mathbf{B}}$ истинны, а высказывания $\mathbf{A \Leftrightarrow} \overline{\mathbf{B}}$ и $\overline{\mathbf{A}} \Leftrightarrow \mathbf{B}$ — ложны.

Порядок выполнения логических операций задается круглыми скобками. Но для уменьшения числа скобок договорились считать, что сначала выполняется операция отрицания ("не"), затем конъюнкция ("и"), после конъюнкции — дизъюнкция ("или") и в последнюю очередь — импликация.

С помощью логических переменных и символов логических операций любое высказывание можно формализовать, то есть заменить логической формулой.

Определение *<u>логической формулы</u>*:

1. Всякая логическая переменная и символы "истина" ("1") и "ложь" ("0") — формулы.
2. Если A и B — формулы, то $\overline{A}$, $A \cdot B$, $A \vee B$, $A \rightarrow B$, $A \Leftrightarrow B$ — формулы.
3. Никаких других формул в алгебре логики нет.

В п. 1 определены **элементарные формулы**; в п. 2 даны **правила образования из любых данных формул новых формул.**

В качестве примера рассмотрим высказывание *"если я куплю яблоки или абрикосы, то приготовлю фруктовый пирог"*. Это высказывание формализуется в виде **(A v B)** $\rightarrow$ **C**. Такая же формула соответствует высказыванию *"если Игорь знает английский или японский язык, то он получит место переводчика"*.

Как показывает анализ формулы **(A v B)** $\rightarrow$ **C**, при определённых сочетаниях значений переменных **A, B** и **C** она принимает значение "истина", а при некоторых других сочетаниях — значение "ложь" (разберите самостоятельно эти случаи). Такие формулы называются **выполнимыми**.

Некоторые формулы принимают значение "истина" при любых значениях истинности входящих в них переменных. Таковой будет, например, формула **A v** $\overline{A}$, соответствующая высказыванию *"Этот треугольник прямоугольный или косоугольный"*. Эта формула истинна и тогда, когда треугольник прямоугольный, и тогда, когда треугольник не прямоугольный. Такие формулы называются **тождественно истинными формулами** или **тавтологиями**. Высказывания, которые формализуются тавтологиями, называются **логически истинными высказываниями.**

В качестве другого примера рассмотрим формулу **A** $\cdot$ $\overline{A}$, которой соответствует, например, высказывание *"Катя самая высокая девочка в классе, и в классе есть девочки выше Кати"*.

Очевидно, что эта формула ложна, так как либо **A,** либо $\overline{A}$ обязательно ложно. Такие формулы называются **тождественно ложными формулами** или **противоречиями**. Высказывания, которые формализуются противоречиями, называются **логически ложными высказываниями.**

Если две формулы A и B одновременно, то есть при одинаковых наборах значений входящих в них переменных, принимают одинаковые значения, то они называются **равносильными**.

Равносильность двух формул алгебры логики обозначается символом "=" или символом "≡"

Замена формулы другой, ей равносильной, называется **равносильным преобразованием** данной формулы.

Математический аппарат алгебры логики очень удобен для описания того, как функционируют аппаратные средства компьютера, поскольку основной системой счисления в компьютере является двоичная, в которой используются цифры 1 и 0, а значений логических переменных тоже два: “1” и “0”.

Из этого следует два вывода:

1. одни и те же устройства компьютера могут применяться для обработки и хранения как числовой информации, представленной в двоичной системе счисления, так и логических переменных;
2. на этапе конструирования аппаратных средств алгебра логики позволяет значительно упростить логические функции, описывающие функционирование схем компьютера, и, следовательно, уменьшить число элементарных логических элементов, из десятков тысяч которых состоят основные узлы компьютера.

Алгоритмический язык программирования Паскаль

Алфавит языка Паскаль

Алфавит Паскаля содержит следующие символы:

- Заглавные и строчные латинские буквы и символ "подчеркивание":

A…Z, a…z, _

- Арабские цифры: 0…9

- Двадцать два специальных символа:

( ) [ ] { } + - * / . , ; : = > < # $ ^ @ '

Русские буквы и другие символы могут использоваться только для записи комментариев и в символьных константах.

Структура программы. Комментарии

В общем виде программа на Паскале состоит из заголовка (содержит слово program и имя программы), раздела описаний и выполняемой части.

Раздел описаний начинается сразу за заголовком программы. Выполняемая часть начинается со слова BEGIN и заканчивается словом END, после которого обязательно ставится точка (конец программы):

```
program <имя программы>;
   < РАЗДЕЛ ОПИСАНИЙ >
BEGIN
   < ВЫПОЛНЯЕМАЯ ЧАСТЬ >
END.
```

Операторы могут располагаться в программе произвольным образом, но обязательно должны заканчиваться точкой с запятой.

Для пояснения текста программы в нее могут включаться комментарии. Комментарий – это произвольная последовательность символов, находящаяся:

1. между фигурными скобками.
2. между круглыми скобками со звездочками: (* … *).
3. после двух символов //.

Пример простой программы:

```
program primer;                    (* заголовок программы *)
```

```
  var x,y,z:real;              // описание переменных
begin                        { начало выполняемой части }
  readln(x,y);                     // ввод чисел X и Y
  z:=x*y;                           // вычисление Z
  writeln(z);              // вывод на экран значения Z
end.                                 { конец программы }
```

Консольные приложения в среде программирования Delphi

После запуска Delphi на экране появляются несколько окон, из которых нас пока интересуют только два: Главное окно и Окно кода программы.

Главное окно Delphi осуществляет управление проектом. Здесь располагается главное меню Delphi.

Командным кнопкам соответствуют следующие горячие клавиши:

F9 – компилирует и выполняет программу,

Ctrl+F2 – завершение работы программы в случае зависания,

F8 – пошаговое выполнение программы,

F7 – пошаговое выполнение программы с отслеживанием работы вызываемых подпрограмм.

Окно кода предназначено для создания и редактирования текста программы. Для создания консольного приложения необходимо выполнить `File=>New=>Other…` и в появившемся диалоговом окне на странице `New` выбрать `Console Application`. Появится окно редактора кода. В окне находится шаблон кода проекта, которому присваивается по умолчанию имя `Project2.dpr`. Первоначально окно кода содержит минимальный исходный текст (версия Delphi 7):

```
program Project2;
{$APPTYPE CONSOLE}
uses
  SysUtils;
begin
  { TODO -oUser -cConsole Main : Insert code here }
end.
```

Идентификаторы

Идентификаторы в языке Паскаль – это имена констант, переменных, меток, типов, процедур и функций. Идентификатор может содержать буквы, цифры и знак подчеркивания и начинаться только с буквы или знака подчеркивания. Пробелы и специальные символы не могут входить в идентификатор.

Важно помнить, что соответствующие заглавные и строчные буквы в идентификаторах и служебных словах не различаются. Таким образом, следующие три идентификатора обозначают одну и ту же переменную: `index`, `INDEX`, `IndEx`.

В качестве идентификаторов нельзя использовать служебные слова. Служебные слова будут рассматриваться по мере изучения языка.

Пример идентификаторов:

правильно	не правильно
punkt_1	1_punkt
nomer1	#1
fort	for (служебное слово)

Оператор присваивания

Оператор присваивания имеет вид: <переменная>:=<выражение>;

где символ := означает присвоить.

Пример:

```
a:=2; b:=3;
c:=a+b;  {c=5}
```

Основные типы данных в Паскале

REAL - ВЕЩЕСТВЕННЫЕ ЧИСЛА. Это числа, содержащие целую и дробную части, разделяемые точкой. Для типа REAL используются следующие арифметические операции:

+ сложение; - вычитание; * умножение; / деление.

Список операций приведен в порядке повышения приоритета. Так, например, в выражении a:=2+6/2*3; сначала выполнится деление 6/2=3, затем умножение 3*3=9, а затем сложение 2+9=11. Выражение в скобках имеет высший приоритет и выполняется первым.

Пример:

```
a:=2+6/(2*3); {рез.=3}
a:=(2+6)/2*3; {рез.=12}
```

INTEGER - ЦЕЛЫЕ ЧИСЛА. Для типа INTEGER вместо операции деления / определены две специальные операции:

DIV деление с отбрасыванием дробной части;

MOD взятие остатка от целочисленного деления.

Пример:

```
b:=5 div 2; {рез.=2}
b:=5 mod 2; {рез.=1}
```

BOOLEAN - ЛОГИЧЕСКИЙ ТИП. Объекты типа BOOLEAN могут принимать только два значения: TRUE - "истина", и FALSE - "ложь".

CHAR - СИМВОЛЫ. В переменную этого типа может быть помещен любой символ. Значение переменной CHAR задается в апострофах.

Пример: a:='a';

STRING - СТРОКИ. Массив символов.

Пример: `a:='абракадабра';`

Переменные

Каждая переменная должна быть описана в разделе описаний. Описание переменных начинается со слова VAR.

```
VAR <имя_переменной>: <тип_переменной>;
```

Например:

```
Var a: real;
    i,j: integer;
    b: Boolean;
```

Константы

Константы описываются в разделе описаний. Описание константы начинается со слова CONST. Значение константы нельзя изменить в теле программы.

Например:

```
Const  e=2.71828;   // Раздел
Var    x: real;     // описаний
       begin
         x:=e;      // Допустимое действие
         e:=3.14;   // Ошибка!
       end.
```

Основные математические функции

Стандартные математические функции языка Паскаль:

Идентификатор (имя) функции	Функция
PI	Число π
Abs(x)	Абсолютная величина |x|
Exp(x)	Экспонента e^x
Sqr(x)	x^2
Sqrt(x)	Квадратный корень x
Ln(x)	Натуральный логарифм *ln* x
Sin(x)	*Sin* x (угол в радианах)
Cos(x)	*Cos* x (угол в радианах)
ArcTan(x)	*Arctg* x (значение в радианах)
Round(x)	Округляет число до ближайшего целого
Trunc(x)	Отбрасывает дробную часть числа
Int(x)	Целая часть числа

`Frac(x)`	Дробная часть числа
`Randomize`	Инициация счётчика случайных чисел
`Random(x)`	Случайное число в диапазоне 0…(x-1) (число x – целое)

Очень много математических функций добавлено в Delphi. Все они находятся в модуле math, который необходимо подключить:

```
uses SysUtils, math;
```

Некоторые из функций модуля math:

`Power(x,y)`	x^y
`Log10(x)`	Десятичный логарифм *lg* x
`LogN(n,x)`	$log_n x$
`RandomRange(min,max)`	Случайное число в диапазоне min…max (числа max и min – целые)

Процедуры ввода и вывода

Операторы вывода:

`WRITE`. Процедура `Write` выводит на экран выражения без перевода строки. Символьные (текстовые) константы заключаются в апострофы. Например, в результате выполнения следующего фрагмента:

```
A := 7;
Write('A=', a);
Write('The end.');
```

на экране будет напечатано следующее:

```
A= 7The end.
```

`WRITELN`. Процедура `Writeln` отличается от `Write` только тем, что она после завершения вывода переводит текущую позицию в начало следующей строки. Например:

```
A := 7;
Writeln ('A=', a);
Write('The end.');
```

В результате выполнения этого фрагмента на экране будет напечатано следующее:

```
A= 7
The end.
```

Переменные типа `real` выводятся с помощью оператора `write` в экспоненциальной форме. Например:

```
c:=13.4;
```

```
write('C=',c);
```

На экран выведется:

```
C=1.34000000000000E+0001
```

Для вывода чисел в нормальной форме необходимо использовать форматный вывод: после имени переменной ставится ":" и указывается общее количество позиций под число, и, через еще одно ":" – количество позиций под дробную часть. Например:

```
write('C=',c:8:3);
```

На экран выведется:

```
C= 13.400
```

Операторы ввода:

READ. Процедура Read выполняет ввод с клавиатуры значений переменных.

Пример:

`Read(a);` - в этом месте выполнение программы приостанавливается, программа ожидает ввода с клавиатуры переменной a.

READLN. Процедура Readln отличается от Read только тем, что после завершения ввода она переводит текущую позицию в начало следующей строки.

Логические выражения

В логических выражениях используются следующие операции сравнения:

< меньше; > больше; = равно;

<= меньше или равно; >= больше или равно; <> не равно.

Результатом операции сравнения является "истина" (TRUE) или "ложь" (FALSE).

Например:

`(X+1)<>2` - имеет значение "ложь", если `X=1`, и значение "истина", если значение переменной `X` отлично от 1.

Логические операции:

NOT логическое НЕ;

AND логическое И;

OR логическое ИЛИ.

Запись таких логических выражений, как `min < X < max` осуществляется следующим образом: `(min < X) and (X < max)`

Результатом логической операции and является "истина", когда "истине" равны оба сравнения.

Условный оператор IF

Оператор IF предназначен для выбора одного из двух возможных действий в зависимости от некоторого условия.

Структура условного оператора:

```
IF   <условие>
THEN <оператор_1>
ELSE <оператор_2>;
```

где <условие> – логическое выражение.

Если <условие> истинно (TRUE) выполняется <оператор_1>, иначе (<условие>=FALSE) выполняется <оператор_2>.

Простой пример:

```
Program primer_1;
  Var  a: real;
  Begin
    readln(a);
    if a>=0
      then writeln('positive number')
      else writeln('negative number');
  End.
```

Условный оператор может быть также без ELSE-альтернативы:

```
IF   <условие>
THEN <оператор>;
```

Составной оператор

В условном операторе IF после THEN и ELSE можно выполнить только один оператор. Чтобы выполнить группу операторов нужно использовать *составной оператор,* т.е. заключить группу операторов между BEGIN и END;.

Пример. Вычислить и распечатать значение k только для случая a>b

```
if a>b then
          begin
            k:=a+b;
            writeln('k=',k);
          end;
```

Оператор безусловного перехода GOTO. Метки

Оператор GOTO передаёт управление оператору, которому предшествует соответствующая *метка*.

Метка – это идентификатор или целое число.

Каждая метка должна быть предварительно описана в разделе описаний:

```
LABEL <метка>;
```

Структура оператора:

```
GOTO  <метка>;
<метка>: <оператор>;
```

Один оператор может быть помечен несколькими метками, которые в этом случае отделяются друг от друга двоеточиями.

Пример:

```
Program K;
Var  V, S: Real;
LABEL  1, Stop;
  Begin
 1: V:=3.62; S:=0;
    IF S>V then GOTO Stop
                  else GOTO 1;
Stop: writeln('the end')
  End.
```

Видно, что приведенная программа при выполнении зацикливается. S никогда не будет больше v и программа всегда будет возвращаться на метку 1.

Т.к. подобные ошибки субъективного характера могут возникать при использовании GOTO, применение на практике этого оператора ограничено.

Оператор выбора CASE

Оператор выбора позволяет выбрать одно из нескольких возможных продолжений программы.

Структура оператора:

```
CASE <ключ_выбора> OF
     <выбор_1>: <оператор_1>;
     <выбор_2>: <оператор_2>;
     . . .
     <выбор_k>: <оператор_k>;
  ELSE <оператор>
  END;
```

где

<ключ_выбора> - значение переменной, которая проверяется оператором CASE;

<выбор_1...k> - переменные того же типа, что и <ключ_выбора>.

В последовательности операторов <выбор_1...k> отыскивается такой, значение которого равно значению <ключ_выбора>. Соответствующий найденному выбору оператор выполняется, после чего оператор CASE завершает свою работу. Если переменная, равная

значению <ключ_выбора>, не будет найдена, управление передается оператору, стоящему за словом ELSE.

Часть ELSE <оператор> может отсутствовать.

Любому из операторов от 1 до k может предшествовать не одна, а несколько значений <выбора>, разделенных запятыми.

Пример

```
 Program SGAU;
   Var student: integer;
     BEGIN
          readln(student);
       CASE student of
            0:     WriteLn('политех');
            1,2,3: WriteLn('Самолет');
            4:     WriteLn('чугун');
            5:     WriteLn('THE BEST');
         Else      WriteLn('...!');
       End;
     END.
```

ОПЕРАТОРЫ ЦИКЛА

Цикл - конструкция языка, позволяющая несколько раз выполнять один оператор или группу операторов.

В Паскале имеется три оператора цикла: WHILE, REPEAT и FOR.

Для того чтобы прервать выполнение цикла используется оператор break.

Оператор цикла WHILE

Структура:

```
WHILE  <условие>  DO  <оператор>;
```

где <условие> – логическое выражение.

Если <условие> истинно (TRUE) выполняется <оператор>, который будет выполняться в цикле до тех пор, пока <условие> не станет ложным (FALSE).

Таким образом, цикл выполняется пока <условие> истинно.

Пример. Программа табулирования функции $y=ax^2$:

```
var x,xk,dx,y,a:real;
begin
write('xn=');readln(x);
write('xk=');readln(xk);
write('dx=');readln(dx);
write('a='); readln(a);
```

```
while x<=xk do
  begin
  y:=a*x*x;
  writeln(x:7:2,' ',y:7:2);
  x:=x+dx;
  end;
readln;
end.
```

Оператор цикла REPEAT

Структура:

```
REPEAT  <оператор_1>; <оператор_2>; ... <оператор_k>;
UNTIL   <условие>;
```

Отличия REPEAT от ранее рассмотренного оператора цикла WHILE:

1. Условие проверяется в конце (т.о. цикл выполняется хотя бы один раз).

2. Признаком окончания цикла является выполнение условия (имеет значение TRUE). Таким образом, цикл выполняется пока условие ложно.

3. В теле цикла может содержаться произвольное количество операторов (REPEAT и UNTIL - ограничители).

Если для организации цикла использовать оператор REPEAT…UNTIL, то приведенный выше пример табулирования функции $y=ax^2$ будет иметь вид:

```
repeat
  y:=a*x*x;
  writeln(x:7:2,' ',y:7:2);
  x:=x+dx;
until x>xk;
```

Оператор цикла FOR

Структура:

```
FOR <параметр>:=<нач.знач.> TO <кон.знач.> DO <оператор>;
```

где <параметр> - переменная цикла типа INTEGER или CHAR.

Значение <параметра> изменяется в цикле от <нач.знач.> до <кон.знач.> с шагом равным 1 и это определяет количество выполнений <оператора>.

Пример.

```
for i:=1 to 3 do write(i:2);
```

Переменная i изменяется в цикле от 1 до 3 с шагом 1 и оператор write выполняется 3 раза.

Результат работы оператора for: 1 2 3.

Существует другая форма оператора цикла:

```
FOR <параметр>:=<нач.знач.> DOWNTO <кон.знач.> DO <оператор>;
```

Замена служебного слова TO на DOWNTO означает, что шаг изменения переменной <параметр> равен -1.

МАССИВЫ

Массив - это совокупность данных одного типа, занимающая непрерывную область оперативной памяти. Массив имеет имя и состоит из ячеек. Каждая ячейка имеет свой номер (индекс массива). Индексы у массива могут иметь тип integer или char.

Чтобы осуществить доступ к ячейке массива, нужно указать имя массива и номер ячейки в квадратных скобках.

Одномерный массив

Описание массива:

```
Var a: array [1..7] of Integer;
Const b: array [0..3] of Real=(1.1,1.2,-3.2,0.5);
```

	1	2	3	4	5	6	7	
a	-2	4	6	8	-5	12	17	a[6]:=1 2;

	0	1	2	3
b	1.1	1.2	-3.2	0.5

В качестве индекса массива можно использовать арифметические выражения, имеющие целый результат

```
i:=2;
a[2*i]:=8;          // a[4]:=8;
b[2*i-1]:=0.5;      // b[3]:=0.5;
```

Примеры

1. Ввод элементов массива

```
Var a: array [1..10] of Integer;
    i: Integer;
begin
    for i:=1 to 10 do
        begin
        Write('a[',i:2,']=');
        ReadLn(a[i]);
        end;
end.
```

2. Вывод элементов массива

```
for i:=1 to 10 do  WriteLn('a[',i,']=',a[i]);
```

3. Операция присваивания с массивами

Для массивов одного типа и одной размерности.

```
Var a,b: array [1..10] of Integer;
< Ввод массива a >
b:=a;
```

4. Сумма элементов массива

```
S:=0;
for i:=1 to 10 do  S:=S+a[i];
```

5. Произведение элементов массива

```
P:=1;
for i:=1 to 10 do  P:=P*a[i];
```

6. Подсчет отрицательных элементов массива

```
n:=0;
for i:=1 to 10 do
if a[i] < 0 then n:=n+1;
```

7. В массиве из 10 элементов найти самое большое число

```
max:=a[1];
for i:=2 to 10 do
if a[i] > max then max:=a[i];
```

8. Найти порядковый номер ячейки, содержащей самое большое число

```
k:=1;
for i:=2 to 10 do
if a[i] > a[k] then k:=i;
```

10. Сортировка массива - расположить элементы массива либо по возрастанию, либо по убыванию

1	2	3	4	5
-2	4	6	8	-5

```
for i:=1 to 5 do
for j:=1 to 5-i do
  begin
   if a[j] > a[j+1] then
      begin
        r:=a[j];
```

```
          a[j]:=a[j+1];
          a[j+1]:=r;
        end;
    end;
```

Двумерные массивы (матрицы)

Двумерный массив - это совокупность ячеек памяти, расположенных по строкам, доступ к каждой ячейке памяти осуществляется путем задания имени матрицы и двух индексов (1-ый - строка, 2-ой - столбец).

Описание:

```
Var a: array [1..3,1..5] of Real;
Const b: array [1..2,1..3] of Integer=((1,2,3),(4,5,6));
```

		1	2	3	4	5
a	1	-2	4	6	8	-5
	2	3	8	4	0	0.1
	3	4.22	3	9.9	6	1.3

При обработке значений, хранящихся в матрице, необходимо использовать двойной цикл.

Примеры

1. Ввод двумерного массива (матрицы) 3x5

```
Var a:   array [1..3, 1..5] of Real;
    i,j: Integer;
begin
    for i:=1 to 3 do
      for j:=1 to 5 do
        begin
          Write('Введите a[',i:2,j:2,']=');
          ReadLn(a[i,j]);
        end;
 end.
```

2. Вывод двумерного массива (матрицы) 3x4

```
    for i:=1 to 3 do
      for j:=1 to 4 do
        Writeln('a[',i:2,j:2,']=',a[i,j]);
```

3. Сумма элементов матрицы

```
S:=0;
for i:=1 to 3 do
for j:=1 to 5 do  S:=S+a[i,j];
```

4. Просуммировать элементы главной диагонали матрицы

```
S:=0;
for i:=1 to 3 do S:=S+a[i,i];
```

5. Транспонирование матрицы

```
for i:=1 to 3 do  // Матрица
for j:=1 to 3 do  // д.б. квадратной
     if i<j then
     begin
        r:=a[i,j];
        a[i,j]:=a[j,i];
        a[j,i]:=r;
     end;
```

Тип данных CHAR

Переменная типа `char` представляет собой любой символ и занимает один байт памяти.

Всего символов 256. Каждому символу соответствует цифровой код в интервале 0..255.

Соответствие между символом и его кодом установлено стандартом ANSI. Первая половина символов с кодами 0..127 стандартная, а вторая половина с кодами 128..255 меняется для различных шрифтов.

Например:

коды 48..57 - символы `'0'...'9'`

коды 97..122 - символы `'a'...'z'`

Для работы с кодами переменных типа `char` используются переменные типа `BYTE`, занимающие 1 байт памяти и предназначенные для хранения целых чисел из диапазона от 0 до 255.

Существуют функции преобразования символа в код и наоборот.

Функция `CHR` Число -> Символ Результат:

```
Var a: byte;                          z
begin
   a:=122;
   writeln(chr(a));
end.
```

Функция `ORD` Символ -> Число Результат:

```
Var a: byte;                        122
    b: char;
begin
   b:='z';
   a:=ord(b);
   writeln(a);
```

```
end.
```

Функция UPCASE преобразует маленькие латинские буквы в большие

```
upcase(a:char): char;
```

СТРОКИ

Строка (string) - это последовательность элементов типа char.

Описание:

```
Var s: string;
```

Можно описать т.н. короткую строку произвольной длины в пределах 255 символов:

```
Var s: string[N];
```

Эта запись означает, что машина выделяет в памяти N+1 байт.

Например:

```
Var s: string[100];
```

Приведенное описание переменной s эквивалентно описанию:

```
Var s: array[0..100] of char;
```
но не идентично.

Количество фактически введенных или существующих символов в переменной типа string постоянно находится в её нулевой ячейке. При любых операциях, приводящих к изменению длины строки, число в нулевой ячейке обновляется.

Пример:

```
Var a: string[9];
begin
  a:='самолёт';
end.
```

0	1	2	3	4	5	6	7	8	9	
7	с	а	м	о	л	ё	т			a

```
writeln(a[3]);
```
выведет букву ”м”

```
writeln(a[0]);
```
выведет символ, код которого = 7.

```
writeln(ord(a[0]));
```
выведет цифру 7

В случае массива CHAR отслеживания количества символов в нулевой ячейке не происходит, поэтому операции и функции, применимые к данным типа string, не подходят к переменным типа array of char.

Основные операции для типа string

1. Вывод и ввод строки `writeln(s); readln(s);`

2. Две строки можно сравнивать с помощью условного оператора if

```
if a=s then ...
```

Можно применить операцию < или >, тогда строки сравниваются посимвольно. Более длинная строка, в случае совпадения первых символов с более короткой, считается больше.

3. Операция конкатенации - сцепления двух строк.

```
b:='само';
c:='лет';
a:=b+c;
d:=b+'вар';
```

Функции для работы со строками

1. Функция определения длины строки

`k:=lenght(a);` результат - число типа `integer`.

2. Функция `copy(st,from,count)`. Она копирует из строки `st` `count` символов, начиная с `from`.

```
a:='самолет'; //записать "оле" в другую строку
b:=copy(a,4,3); for i:=1 to 4 do write(b,'! ');
```

3. Процедура `delete(st,from,count)`. Из строки `st` удаляет `count` символов, начиная с символа `from`.

```
a:='арбат'; //надо, чтобы осталось "арба"
delete(a,5,1);
```

4. Процедура `insert(subst,st,from)` - вставка подстроки `subst` в строку `st`, начиная с символа `from`.

```
b:='шайтан';
insert(a,b,7);
```

5. Функция `pos` (от position)

```
p:=pos(subst,str);
```

2 входных параметра: строка `subst` и `str`. Функция возвращает целое число - позицию первого символа подстроки `subst` в строке `str`.

```
str:='гидроэлектростанция';
subst:='электро';
p:=pos(subst,str);
```

Результат: `p=6`

Если подстрока не обнаружена, то `p=0`.

Массивы строк

Пример:

```
var s: array [1..4] of string[5]; // массив из 4 строк
                                  // string[5]
one,two,first,second,i,j: integer;
begin
  s[1]:='наука';
  s[2]:='умеет';
  s[3]:='много';
```

```
 s[4]:='гитик';
 writeln('In what rows are your cards?');
 readln(one);
 readln(two);
 for i:=1 to 5 do
 for j:=1 to 5 do
     if (s[one,i]=s[two,j]) and (i<>j) then
 begin first:=i; second:=j; end;
 writeln(s[one,first],' ',s[two,second]);
end.
```

ПОДПРОГРАММЫ

Подпрограмма – самостоятельная часть программы, предназначенная для решения конкретной задачи.

В Паскале 2 вида подпрограмм:

1. Функции.
2. Процедуры.

Для того чтобы прервать выполнение подпрограммы используется оператор `exit`.

Функции

Предназначена для выдачи одного единственного результата. Возвращаемое значение передается из функции с помощью оператора `Result` или присваивается внутри функции ее имени. Описание функции начинается со слова `function`.

Структура программы с функцией:

```
<Раздел описаний>
 Function <имя функции>(<описание параметров>):<тип функции>;
 begin
   <тело функции>
   result:=… {или}  <имя функции >:=…
 end;
begin
<тело программы>
<переменная>:=<имя функции>(<список параметров>);
end.
```

Пример подпрограммы-функции:

Программа определения площади треугольника по формуле Герона.

```
Var m,n,k,s: real;
  Function str(a,b,c: real): real;
  Var p: real;
```

```
  begin
    p:=(a+b+c)/2;
    str:=sqrt(p*(p-a)*(p-b)*(p-c));
  end;
begin
  writeln('Введите стороны треугольника');
  readln(m,n,k);
  s:=str(m,n,k);    //вызов функции
end.
```

Функция str может иметь и такой вид:

```
  Function str(a,b,c: real): real;
  Var p: real;
  begin
    p:=(a+b+c)/2;
    result:=sqrt(p*(p-a)*(p-b)*(p-c));
  end;
```

Формальные и фактические параметры

Параметры, которыми манипулирует функция, называются формальными, а параметры, которые передаются функции при ее вызове из основной программы, называются фактическими. Содержимое фактических параметров при вызове функции копируется формальными параметрами.

В нашем примере m,n,k – фактические параметры, a,b,c – формальные параметры.

Глобальные и локальные параметры

Глобальные параметры – параметры, определенные в основной программе, доступны в любой точке программы.

Локальные параметры – параметры, определенные в теле подпрограммы, доступны только в самой подпрограмме.

В нашем примере m,n,k,s – глобальные параметры, a,b,c,p – локальные параметры.

Т.о. обмен информацией между основной программой и подпрограммой может быть реализован через глобальные параметры.

```
Var a,b: integer;         // Глобальные параметры
Function aib: integer;    // Локальных параметров нет
begin
  Result:=a+b;
end;
begin
  readln(a,b);
```

```
      writeln('a+b=',aib);
    end.
```

Передача параметров в функцию по ссылке

Механизм передачи параметров в функцию по ссылке позволяют сделать доступными в теле функции параметры, объявленные в основной программе. Значения этих параметров могут меняться после окончания работы функции. Для этой цели в заголовок функции перед списком передаваемых ей параметров ставится ключевое слово Var.

```
Var a,b,str,hyp: real;
    Function s(var x,y: real): real;
    begin
         Result:=x*y/2;
         x:=sqr(x);
         y:=sqr(y);
    end;
begin
  readln(a,b);
  str:=s(a,b);    // Площадь прямоуг. треугольника.
  hyp:=sqrt(a+b); // Гипотенуза, т.к. из функции s
end.              // вернулись квадраты переменных a и b
```

Процедуры

Отличия процедуры от функции:

1. Вместо ключевого слова function - слово procedure;

2. У процедуры не определяется тип;

3. Имеются входные и выходные параметры;

4. Для вызова процедуры в основной программе просто указывается ее имя со списком параметров.

Пример.

```
{Процедура, вычисляющая сумму и произведение 3-х чисел}
Var a,b,c,d,e: integer; // глобальные параметры
procedure sumpr(x,y,z: integer; var v,w: integer);
  begin                  // x,y,z - входные параметры
   v:=x+y+z; w:=x*y*z    // v,w - выходные параметры
  end;
begin
 readln(a,b,c);
 sumpr(a,b,c,d,e);      // вызов процедуры
 writeln('sum=',d,' mult=',e);
```

```
end.
```

ТИПЫ ДАННЫХ

Тип определяет множество допустимых значений переменной.

В Паскале имеются следующие типы данных: простые типы, структурированные типы, строки и указатели.

ПРОСТЫЕ ТИПЫ

К простым относятся порядковые, вещественные (real) типы и тип дата-время.

Порядковые типы данных

Порядковые типы: `integer, char, boolean,` а также перечисляемый тип и тип-диапазон.

К ним применимы следующие процедуры и функции:

Процедура `dec(x,y)` - уменьшает значение числа `x` на `y`.

Пример: `x:=5; y:='d'; dec(x,2); dec(y,3);` рез.: `x=3 y='a'`.

Процедура `inc(x,y)` - увеличивает значение числа `x` на `y`.

Процедуры `dec` и `inc` могут иметь только один параметр: `dec(x)`, `inc(x)`. В этом случае значение `x` уменьшается (увеличивается) на единицу. Часто вместо `i:=i+1;` используют `inc(i);`

Функция `pred(x)` - возвращает значение, предшествующее `x`.

Пример: `x:='b'; y:=pred(x);` рез.: `y='a'`

Функция `succ(x)` - возвращает значение, следующее за `x`.

Оператор TYPE

В Паскале существует ключевое слово `TYPE`, которое позволяет использовать существующие в Паскале типы данных, а также описывать свои.

Задание типа осуществляется перед описанием переменных:

```
type <тип>=<допустимые значения>;
var  <переменная>:<тип>;
```

Перечисляемый тип данных

Определяется набором идентификаторов, разделенных запятой и указываемых в круглых скобках.

Раздел описаний:

```
type week=(Monday,Tuesday,Wednesday,Thursday,Friday,
Saturday,Sunday);
var  day: week;
begin
  day:=Sunday;
  if day < Saturday  then  writeln('Week-day');
  if day > Friday  then  writeln('Week end');
```

```
end.
```

Значениями переменной `day` могут быть только идентификаторы, перечисленные в типе `week`.

Известный нам логический тип является частным случаем перечисляемого типа:

```
type boolean=(False,True);
```

Перечисляемый тип можно использовать в качестве индекса массива.

Переменные перечисляемого типа не могут вводиться оператором `read` и выводиться оператором `write`.

Тип-диапазон

Тип-диапазон есть подмножество своего базового типа. Задается min и max значениями, например:

```
type digit= '0'..'9'; //базовый тип char
```

Тип-диапазон можно не описывать в разделе `type`, а сразу указывать при описании переменной:

```
var month: 1..12; //базовый тип integer
```

Тип-диапазон имеет все свойства базового типа, но с ограничениями по min и max значениям.

Пример.

```
type WeekEnd=Saturday..Sunday; //базовый тип week
var: w: WeekEnd;
begin
  w:=Saturday;
end.
```

Код переменной типа диапазон равен её порядковому номеру в базовом типе (нумерация начинается с нуля). Так `Ord(w)` вернет значение 5.

Тип дата-время

`TDateTime` – тип дата-время, предназначен для хранения даты и времени. Вещественное число. В целой части храниться дата, в дробной – время. Дата - количество суток, прошедших с 30.12.1899 г., а время – как часть суток, прошедших с 0 часов. Над данными типа `TDateTime` определены те же операции, что и над вещественными числами.

Функции для типа дата-время:

`Date: TDateTime;` - возвращает текущую дату;

`Now: TDateTime;` - возвращает текущую дату и время;

`DateToStr(D: TDateTime): String;` – преобразует дату в строку символов в формате короткой даты, который устанавливается в Windows (обычно `dd.mm.yyyy`);

`FormatDateTime(F: String; D: TdateTime): String;` – преобразует дату и время в строку символов в формате, указанном в F.

Пример:

```
var D:TDateTime; s1,s2,s3:string;
begin
D:=Now;
s1:=DateToStr(D);
s2:=FormatDateTime('dd.mm.yy hh:mm:ss',D);
s3:=FormatDateTime('dd mmmm yyyy',D);
end;
```

Результат работы программы:

s1=’12.03.2002’

s2=’12.03.02 16:45:07’

s3=’12 Март 2002’

Преобразование типов

В подпрограммы нельзя передавать в явном виде массивы. В этом случае тип массива нужно преобразовывать в простой тип.

Пример.

Неверной будет запись:

```
procedure sum(a:array[1..10] of real);
```

Мы должны переопределить тип переменной a в простой тип:

```
type massive=array[1..10] of real;
Var a: massive;  summa: real;
    i: integer;
      procedure sum(b:massive; var s:real);
      begin
       for i:=1 to 10 do s:=s+b[i];
      end;
    begin
       for i:=1 to 10 do readln(a[i]);
       sum(a,summa);
       writeln('summa=',summa:1:2);
    end.
```

СТРУКТУРИРОВАННЫЕ ТИПЫ

Четыре: массивы, записи, множества, файлы.

Записи

Записи позволяют хранить в одной переменной данные, имеющие различный тип.

Структура объявления типа записи:

```
type <запись> = RECORD
```

```
  <поля записи>
END;
```

После объявления типа можно описать переменную типа запись:

```
Var <переменная> : <запись>;
```

Пример:

```
type Info = record
  Name:
  string[25];
  Income: real;
  Gr:
  TdateTime;
end;
var person: Info;
    student: array [1..20] of Info;
```

Поля записи

Для обработки доступна как вся запись, так и отдельные ее поля.

При обращении к отдельным полям указывается имя всей записи и имя отдельного поля, разделенные точкой.

```
person.Gr:=StrToDate('11.05.85');
student[1].Name:='Куролесов';
for i:=1 to 20 do
    student[i].Income:=10000;
writeln(student[1].Income:1:2);
```

Оператор WITH...DO

Применяется при совместной обработке нескольких полей записи.

Идентификатор записи указывается однократно в начале фрагмента программы обработки записи. Это позволяет более компактно представлять переменные записей.

Например, вместо:

```
person.Name:='Одуванчиков';
person.Income:=1000;
```

можно записать:

```
WITH person DO
begin
  Name:='Одуванчиков';
  Income:=1000;
end;
```

Множества

Множество – это набор логически связанных друг с другом объектов одного типа. Элементов в множестве может быть от 0 до 256.

Описание множества:

```
type <тип>=set of <диапазон>;
var <множество>:<тип>;
```

Пример:

```
type digit=set of 0..9;
var s1,s2,s3,s4:digit;
    AlphaBet: set of 'a'..'z'; // можно и так
```

Конкретные значения множественного типа задаются списком элементов множества, заключенных в квадратные скобки:

```
s1:=[1,2,3,4];
s2:=[3..5];
s3:=[]; // пустое множество
```

Операции над множествами

1. Объединение двух множеств, знак "+"

Результат: множество, содержащее элементы первого множества, дополненное недостающими элементами из 2-го множества.

```
s4:=s1+s2; // s4 = [1,2,3,4,5]
```

2. Разность двух множеств, знак "-"

Результат: множество, содержащее элементы первого множества, не принадлежащие 2-му.

```
s4:=s1-s2; // s4 = [1,2]
```

3. Пересечение двух множеств, знак "*"

Результат: множество, содержащее элементы, общие для 2-х множеств.

```
s4:=s1*s2; // s4 = [3,4]
```

4. Операции отношения: =, <> , >= ,<=

Два множества являются эквивалентными, когда все их элементы одинаковы, при этом порядок следования элементов в множестве несущественен.

```
s2=[5,4,3];  - TRUE      s2=[1,3,5];  - FALSE
s2 >= [];    - TRUE      s2 >= [4,5]; - TRUE
s2 >= [1,4]; - FALSE //элементы множеств не совпадают
```

5. Оператор in – проверка элемента на принадлежность множеству.

```
4 in s2;     - TRUE
3*3 in s2;   - FALSE
```

Оператор in можно использовать для более компактной записи некоторых логических выражений:

```
in [множество]
```

Например, выражение 3≤ x ≤5 вместо

```
If (3<=x) and (x<=5) then …
```

можно записать как

```
if x in [3..5] then …
```

Или ещё вариант: `if c in ['+','-','/','*'] then …`

Файлы

Файл – именованная область внешней памяти компьютера (диска, дискеты). Для доступа к конкретному файлу в программе используется переменная файлового типа.

Переменные файлового типа можно объявить тремя способами:

`<имя>: File of <тип>;` - типизированный файл, где `<тип>` – любой тип данных

`<имя>: TextFile;` - текстовый файл

`<имя>: File;` – нетипизированный файл

Процедуры обработки файлов:

1. `AssignFile(<переменная файлового типа >,<имя файла>);`

`AssignFile` связывает переменную файлового типа с именем файла.

Например

```
Var f: TextFile;   // переменная файлового типа
begin
AssignFile(f,'d:\kc\512a\1.txt');
AssignFile(f,'1.txt'); - файл находится в текущей папке.
```

2. `Rewrite(f)` - создает новый файл или стирает содержимое в уже существующем файле.

3. `Reset(f)` - открывает для чтения файл, если он существует. Чтобы проверить, существует ли файл, необходимо использовать функцию `FileExists`. Пример:

```
if FileExists('1.txt') then – файл существует
                       else – файл не существует
```

4. `Read(f,…)` и `Write(f,…)` – чтение данных из файла и запись данных в файл. `Readln` и `Writeln` – только для текстовых файлов.

5. `Append(f)` – инициирует запись в существующий файл типа `TextFile`.

6. `EOF(f)` – конец файла. Логическая функция. `TRUE` – если достигнут конец файла.

7. `CloseFile(f)` - закрывает файл.

Пример:

```
Var f: TextFile;
    s: string;
    a:real;
begin
```

```
 AssignFile(f,'1.txt');
 if not FileExists('1.txt')
    then rewrite(f)
    else begin reset(f); append(f); end;
 readln(s,a);
// добавляем строку s и переменную a в файл 1.txt :
 writeln(f,s,' ',a:1:2);
end.
```

УКАЗАТЕЛИ

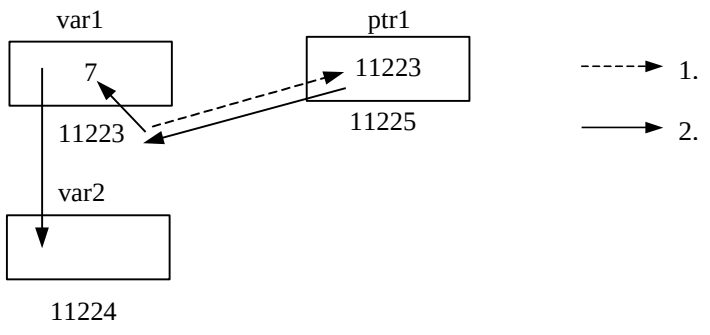
Указатели относятся к т.н. динамически размещаемым переменным. Оперативная память компьютера представляет собой совокупность ячеек для хранения информации – байтов. Каждый байт имеет собственный номер или адрес. Указатели содержат адреса байтов.

Если за указателем стоит значок ^, то имеется ввиду *значение*, на которое указывает указатель, а если значка ^ нет, то имеется ввиду *адрес*, по которому размещены данные. Для получения адреса переменной используется функция `addr`.

Имеется 2 типа указателей:

1. Типизированные. Указывают на переменные определенного типа. Для объявления перед типом ставится значок ^.

```
Var ptr1: ^integer; // указатель на переменные типа integer
    var1, var2: integer;
begin
  var1:=7;
var2:=var1; // присваиваем var2 значение var1 напрямую
// или с помощью указателей:
ptr1:=addr(var1);//1. ptr1 присваивается адрес var1 (11223)
var2:=ptr1^;//2. переменной var2 присваивается значение
            //   переменной var1 на которую указывает ptr1
end.
```



Две следующие записи идентичны:

```
var1:=10;
```

или

```
ptr1^:=10;
```

2. Нетипизированные, т.е. не указывающие на определенный тип. Имеют свой тип Pointer.

```
Var ptr2: Pointer; //указатель на переменные любого типа
```

МОДУЛИ (UNIT)

Модуль (UNIT) – набор процедур и функций, хранящихся в отдельном файле. Файлы, содержащие модули, имеют расширение pas.

Структура:

```
UNIT <имя модуля>;
INTERFACE //интерфейсная часть
< Подключение модулей, объявление меток и переменных,
  заголовки функций >
IMPLEMENTATION //раздел реализации
< Подключение модулей, объявление меток и переменных,
  тела функций >
END.
```

Подключение модуля к основной программе и к другим модулям осуществляется с помощью оператора USES, который помещается в разделе описаний перед описанием типов, переменных, констант и меток:

```
USES <имя модуля>;
```

Для создания в проекте нового модуля необходимо выполнить File=>New=>Unit. В Окне кода программы появиться шаблон кода модуля, которому присваивается по умолчанию имя Unit1:

```
unit Unit1;
interface
implementation
end.
```

Пример модуля, содержащего процедуру округления дробной части числа с указанной точностью:

```
unit Unit1;
interface
 procedure okrugl(s:integer;var a:real);
implementation
 procedure okrugl(s:integer;var a:real);
  //Округление дробной части числа a с точностью s
  begin
```

```
  a:=round(a*s)/s;
  end;
end.
```

Программа, использующая модуль Unit1:

```
uses Unit1;      // подключение модуля Unit1
var b:real;
begin
readln(b);
okrugl(100,b);  // вызов процедуры из модуля Unit1
writeln(b:1:6);
readln;
end.
```

Лабораторный практикум по информатике

Лабораторная работа №1 «Сложное выражение»

Задание: Создать консольное приложение Delphi, реализующее ввод исходных данных, вычисление и вывод на экран результатов расчета.

Выполнение:

Для создания консольного приложения необходимо выполнить `File=>New=>Other…` и в появившемся диалоговом окне на странице `New` выбрать `Console Application`. Появится окно редактора кода, показанное на рис.1. В окне находится шаблон кода проекта, которому присваивается по умолчанию имя `Project2.dpr`. В теле программы (между операторами **`begin`**…**`end`**) имеется приглашение к вводу: *`Insert code here`*.

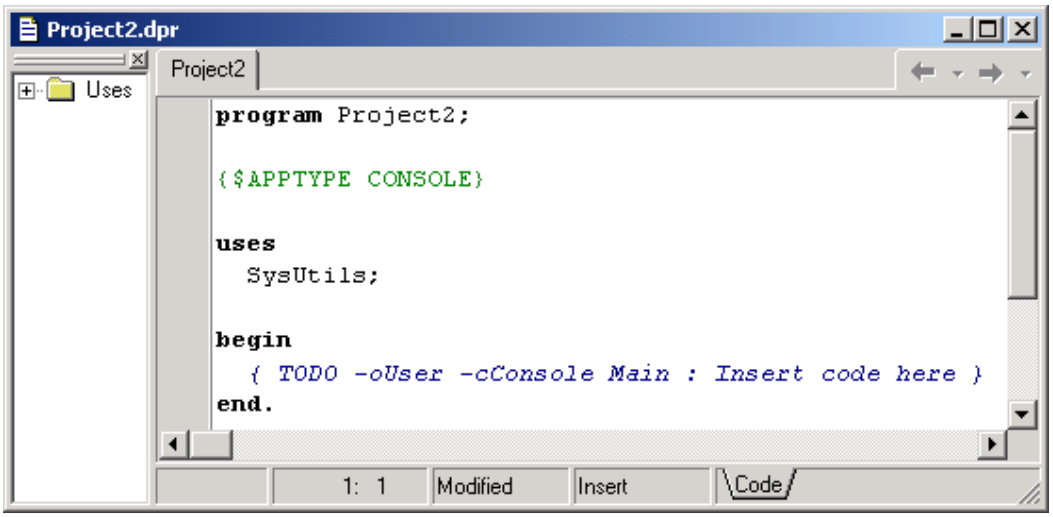


Рис. 1. Редактор консольного приложения (версия Delphi 7)

Сохранение проекта

1. Выполните команду `Save Project As…` из пункта меню `File`.
2. В появившемся окне выберите в пункте `Папка:` свой рабочий каталог (например, `d:\kc\515`).
3. Щелкните по кнопке  и создайте в своём рабочем каталоге папку `ЛР1` (`d:\kc\515\ЛР1`).
4. Откройте папку `ЛР1`.
5. В пункте `Имя файла:` по умолчанию предлагается сохранить проект с именем `Project2.dpr`. Замените `Project2.dpr` на `Lr1.dpr` и нажмите клавишу `Сохранить`.

Ввод текста программы

1. Дополним шаблон кода проекта собственными операторами. Для создания паузы после завершения работы программы можно использовать оператор `Readln` без

параметров, при достижении которого программа ожидает нажатия клавиши Enter. Текст программы будет выглядеть следующим образом (вставляемые операторы выделены):

```
program Project2;
{$APPTYPE CONSOLE}
uses
  SysUtils;
var x,y,z:integer;
begin
  write('x='); readln(x);
  write('y='); readln(y);
  z:=x+y;
  writeln('sum=',z);
  readln;
end.
```

2. Запустите программу на выполнение посредством нажатия клавиши F9. Если программа написана без ошибок, появиться окно консольного приложения – окно MS DOS.
3. Введите значения исходных данных (в нашем примере это значения переменных x и y, например, 3 и 2).
4. После ввода данных на экране появиться результат работы программы (в нашем примере надпись sum=5).
5. Нажмите клавишу Enter для завершения работы программы.
6. Измените текст программы в соответствии со своим заданием.

Задания к лабораторной работе №1

№	Расчетные формулы	Исходные данные
1	$s=\left\lvert x^{y/x}-\sqrt{\frac{y}{x}}\right\rvert;\quad w=(y-x)\frac{y-\frac{z}{y-x}}{1+(y-x)^2}$	x = 1.82 y = 18 z = -3.29
2	$s=1+x+\frac{x^2}{2}+\frac{x^3}{3}+\frac{x^4}{4};w=x(\sin x+\cos y)$	x = 0.33 y = 0.02
3	$y=e^{-bt}\sin(at+b)-\sqrt{\lvert bt+a\rvert};\ s=b\sin\left(at^2\cos(at)\right)-1$	a = -0.5 b = 1.7 t = 0.44

4	$w=\sqrt{x^2+b}-\frac{b^2\sin^3(x+a)}{x};\ y=\cos^2(x^3)-x/\sqrt{a^2+b^2}$	a = -0.5 b = 15.5 x = -2.9
5	$s=x^3 tg^2((x+b)^2)+\frac{a}{\sqrt{x+b}};\ \ g=\frac{bx^2-a}{e^{ax}-1}$	a = 16.5 b = 3.4 x = 0.61
6	$r=\frac{x^2(x+1)}{b}-\sin^2(x+a);\ s=\sqrt{\frac{xb}{a}}+\cos((x+b)^3)$	a = 0.7 b = 0.05 x = 0.5
7	$y=\sin^3((x^2+a)^2)-\sqrt{\frac{x}{b}};\ z=\frac{x^2}{a}+\cos((x+b)^3)$	a = 1.1 b = 0.04 x = 0.2
8	$y=b\cdot tg^2 x-\frac{a}{\sin^2(x/a)};\ d=a\cdot e^{\sqrt{a}}\cdot\cos(bx/a)$	a = 3.2 b = 17.5 x = -4.8
9	$f=\ln(a+x^2)+\sin^2(x/b);\ \ z=e^{-cx}\cdot\frac{x+\sqrt{x+a}}{x-\ln(\lvert x-b\rvert)}$	a = 10.2 x = 2.2 b = 9.2 c = 0.5
10	$y=\frac{a^{2x}+b^{-x}\cdot\cos((a+b)x)}{x+1};\ r=\sqrt{x^2+b}-b^2\sin(x+a)/x$	a = 0.3 b = 0.9 x = 0.61
11	$z=\sqrt{ax\cdot\sin(2x)+e^{-2x}(x+b)};\ w=\cos^2(x^3)-x/\sqrt{a^2+b^2}$	a = 0.5 b = 3.1 x = 1.4
12	$s=x\cdot ctg^2(x-a)+\frac{b}{\sqrt{x+b}};\ \ w=\frac{bx-a}{e^{b-x}-1}$	a = 16.5 b = 3.4 x = 0.61
13	$u=\frac{a^2x+e^{-x}\cos(bx)}{bx-e^{-x}\sin(bx)+1};\ f=e^{2x}\ln(a+x)-b^{3x}\ln(b-x)$	a = 0.5 b = 2.9 x = 0.3
14	$a=\frac{2\cos\left(x-\frac{x}{y}\right)}{1/2+\sin^2(y)};\quad b=1+\frac{z^2}{3+z^2/5}$	x = 1.42 y = -1.27 z = 3.5

15	$s = e^{-ax}\sqrt{x+1} + e^{x}\sqrt{x+1{,}5};\ z = \dfrac{\sin x}{\sqrt{1+m^2\sin^2 x}} - cm \cdot ln(mx)$	a = 0.5 x = 1.7 m = 0.7 c = 2.1

Лабораторная работа № 2 «Разветвляющиеся алгоритмы»

1. С клавиатуры задано целое трехзначное число. Определите, есть ли среди цифр задуманного числа одинаковые.
2. С клавиатуры заданы коэффициенты A, B, C биквадратного уравнения. Напишите программу решения этого уравнения.
3. С клавиатуры вводятся координаты точки (x,y) на плоскости. Определите номер квадранта, в котором находится точка. Если точка лежит на оси, то укажите на какой.
4. Идет k-ая секунда суток (k задается с клавиатуры). Определите, сколько полных часов и полных минут прошло к этому моменту от начала суток.
5. С клавиатуры задаются A, B, C – ребра кирпича и X, У – стороны прямоугольного отверстия. Определить, пройдет ли кирпич в отверстие.
6. С клавиатуры заданы три целых числа A, B, C. Перераспределите их значения так, что A, B, C станут соответственно наименьшим, средним и наибольшим значениями.
7. С клавиатуры заданы длины сторон двух прямоугольников. Определите, можно ли первый прямоугольник целиком поместить во втором.
8. С клавиатуры заданы координаты двух точек. Определить, какая из них находится ближе к началу координат.
9. С клавиатуры заданы координаты двух точек. Определить, лежат ли они на одной окружности с центром в начале координат.
10. Дано трехзначное число. Написать программу определения является ли оно палиндромом («перевертышем»), т.е. числом, десятичная запись которого читается одинаково слева направо и справа налево. Верно ли, что это число содержит одинаковые цифры? Например: числа 121, 222 являются палиндромами.
11. Дано четырехзначное число. Написать программу определения является ли оно палиндромом («перевертышем»), т.е. числом, десятичная запись которого читается одинаково слева направо и справа налево. Верно ли, что все его цифры числа различны? Например: число 1221 является палиндромом.

12. Дано пятизначное число. Написать программу определения является ли оно палиндромом («перевертышем»), т.е. числом, десятичная запись которого читается одинаково слева направо и справа налево. Верно ли, что это число содержит ровно три одинаковые цифры числа? Например: число 12321 является палиндромом.

13. Дано натуральное число n (100<n<1000). Определите:

а) является ли это число четным;

б) сумма цифр этого числа кратна 3;

в) все ли цифры числа различны.

14. Дано натуральное четырехзначное число n.

а) выбросить из записи числа цифры 0 и 5, оставив прежним порядок остальных цифр. Например: из числа 1509 должно получиться 19.

б) проверьте, являются ли исходное и полученное в пункте а) числа нечетными.

15. Дано натуральное трехзначное число n. Определите:

а) является ли оно числом Армстронга, т.е. сумма его цифр, возведенных в 3 степень, равна самому числу (например: 153=13 +53+33)

б) найти произведение четных цифр этого числа.

16. Дано натуральное четырехзначное число n. Определите:

а) все ли цифры этого числа различны;

б) сумма цифр числа и само число кратны 7.

17. Дано натуральное трехзначное число n. Определите:

а) все ли цифры этого числа четные;

б) больше ли цифра сотен цифры единиц.

18. Дано натуральное четырехзначное число n. Определите:

а) найти количество нечетных цифр этого числа;

б) является ли сумма его цифр двузначным числом.

Лабораторная работа № 3 «Циклические алгоритмы»

Варианты работ:

Вариант №1

1. С клавиатуры вводится последовательность из N целых чисел. Найти наибольшее число из всех отрицательных и его порядковый номер в последовательности. В случае совпадения наибольших значений вывести номер первого из них.
2.Определить порядковый номер элемента последовательности 21, 22, 23,…,2n, значение которого превысит 100.
3. Найти все простые числа из заданного промежутка [a,b], где a и b задаются с клавиатуры.

Вариант №2

1.С клавиатуры вводится последовательность из целых чисел, 0 - конец последовательности. Найти наименьшее число и его порядковый номер в последовательности. В случае совпадения значений вывести номер последнего из них.
2. Вычислить сумму всех элементов последовательности Фибоначчи, не превосходящих N, где N задается с клавиатуры.
3. С клавиатуры вводится последовательность из N целых чисел. Определить, сколько среди них встретилось совершенных чисел. Натуральное число называется совершенным, если оно равно сумме всех своих положительных делителей, кроме самого себя. Например, 28 - совершенное число, т.к. 28=1+2+4+7+14 .

Вариант №3

1.С клавиатуры вводится последовательность из целых чисел, 0 – конец последовательности. Определить, сколько раз в последовательности встретились пары равных соседних чисел. Если такие пары были, то вывести на печать номера последней пары.
2.Определить количество элементов последовательности 21, 22, 23,…,2n, сумма которых не превышает 200.

Вариант №4

1.С клавиатуры вводится последовательность из целых чисел, 0 – конец последовательности. Определить, является ли последовательность возрастающей.
2.Вычислить порядковый номер элемента последовательности 1+ 1/(1+2)+1/(2+3)+…+1/(n+ (n +1)), значение которого станет меньше 0,002.
3. Для всех натуральных трехзначных чисел:
а) проверьте, является ли оно числом

3. С клавиатуры вводится последовательность из N целых чисел. Определить, сколько среди них встретилось палиндромов. Назовем число палиндромом, если его запись читается одинаково справа налево и слева направо (как, например, 4884, 393, 55, 1).

Армстронга, т.е. сумма его цифр, возведенных в 3 степень, равна самому числу (например: 153=13 +53+33)
б) найдите произведение всех чисел, которые кратны 5 и содержат хотя бы одну четную цифру.

Вариант №5

1.С клавиатуры вводится последовательность из целых чисел, 0 - конец последовательности. Найти наименьшее число и его порядковый номер в последовательности. В случае совпадения значений вывести номер последнего из них.
2.Определить количество элементов последовательности 21, 22, 23,…,2n, которые попадают в интервал [10, 150].
3. Для всех натуральных чисел n (100<n<1000). Найдите все числа а) у которых сумма цифр числа кратна 3;
б) все цифры числа различны

Вариант №6

1.Вводится последовательность из N целых чисел. Найти два наименьших по значению числа.
2.Население города увеличивается ежегодно на 3% каждый год. В 1995 году население города составляло 10500 человек. Напечатайте на экране год, в котором численность населения города превысит 15000.
3.С клавиатуры вводится последовательность из N целых чисел.
Найти:
-вывести номера тех элементов, значения которых больше заданного числа А;
-подсчитать количество положительных элементов кратных К.
(N,A,K- задаваемые константы)

Лабораторная работа №4 «Числовые ряды »

Задания

1. Получить n-ое число Фибоначчи. Задается натуральное n (n<=30). Требуется получить n-ый член рекуррентной последовательности { F0=1, F1=1, Fk+1= Fk+Fk-1 для k>1 } .

2. Получить первое число Фибоначчи, превышающее заданный предел. Задается натуральное M. Требуется получить член рекуррентной последовательности { F0=1, F1=1, Fk+1= Fk+Fk-1 для k>1 }, превышающий M
3. Вычислить кубический корень. Для заданного вещественного X>0 построить рекуррентную последовательность {Y0=X,Yi+1=(2*Yi+X/Yi2)/3} до достижения требуемой точности Eps=10-4 , т.е. получить такое Yi+1 , что | Yi+1 - Yi | < Eps
4. Вычислить корень k-ой степени. Для заданного вещественного X>0 и натурального k построить рекуррентную последовательность {Y0=X,Yi+1=(Yi*(k-1)+X/Yik-1)/k} до достижения требуемой точности Eps=10-4 , т.е. получить такое Yi+1 , что | Yi+1 - Yi | < Eps.
5. Найти для заданного значения X сумму ряда:
   S=x + (2*x3)/7 - (4*x6)/9 + (6*x9)/11 – (8*x12)/13 + …+ (14*x21)/19
6. Найти сумму членов ряда. Суммирование продолжать пока разность между текущим и предыдущим значениями суммы остается больше 0,005. Вывести на экран значение суммы, последнего члена ряда и его порядковый номер:
   S= (1*3)/(2*4) + (1*3*5)/(2*4*6) + (1*3*5*7)/(2*4*6*8) +…..
7. Найти сумму членов ряда. Суммирование продолжать пока разность между текущим и предыдущим членами остается больше 0,04. Вывести на экран значение суммы, последнего члена ряда и его порядковый номер:
   S= 1/(2*1) + 1/(2*2) +1/(2*3) +1/(2*4) +1/(2*5) +….
8. Найти минимальное количество слагаемых в сумме членов ряда, при котором эта сумма станет больше 120. Вывести на экран значение суммы, последнего члена ряда и его порядковый номер: S= 1*1 + 1*2*2 +1*2*3*3 + 1*2*3*4*4 + …..

Лабораторная работа №5 «Числовые последовательности »

Задания

1. С клавиатуры вводится последовательность из N целых чисел. Найти наибольшее число из всех отрицательных и его порядковый номер в последовательности. В случае совпадения наибольших значений вывести номер первого из них.
2. С клавиатуры вводится последовательность из целых чисел, 0 - конец последовательности. Найти наименьшее число и его порядковый номер в последовательности. В случае совпадения значений вывести номер последнего из них.
3. С клавиатуры вводится последовательность из целых чисел, 0 – конец последовательности. Определить, сколько раз в последовательности встретились

пары равных соседних чисел. Если такие пары были, то вывести на печать номера последней пары.

4. С клавиатуры вводится последовательность из целых чисел, 0 – конец последовательности. Определить, является ли последовательность возрастающей.
1. С клавиатуры вводится последовательность из N целых чисел. Определить, сколько среди них встретилось палиндромов. Назовем число палиндромом, если его запись читается одинаково справа налево и слева направо (как, например, 4884, 393, 55, 1).
5. С клавиатуры вводится последовательность из N целых чисел. Определить, сколько среди них встретилось совершенных чисел. Натуральное число называется совершенным, если оно равно сумме всех своих положительных делителей, кроме самого себя. Например, 28 - совершенное число, т.к. 28=1+2+4+7+14 .
6. С клавиатуры вводится последовательность из N целых чисел. Определить, сколько среди них встретилось чисел Армстронга. Натуральное число из n цифр является числом Армстронга, если сумма его цифр, возведенных в n-ую степень, равна самому числу (как, например, 153=13+53+33).
7. С клавиатуры вводится последовательность из N целых чисел. Определить, сколько среди них встретилось дружественных чисел. Два натуральных числа являются дружественными, если каждое из них равно сумме делителей другого, кроме самого этого числа.

Лабораторная работа №6 «Одномерные массивы»

Задания

1. Массив целых чисел, состоящий из 20 элементов, задан случайным образом числами из промежутка [-45,65].

   Найти:

   - -сумму элементов, имеющих нечетное значение;
   - -вывести индексы тех элементов, значения которых больше заданного числа А;
   - -подсчитать количество положительных элементов кратных К.

   Числа А и К вводятся с клавиатуры.
2. Массив целых чисел, состоящий из 20 элементов, задан случайным образом числами из промежутка [-15,85].

   Найти:

   - -сумму элементов, имеющих нечетные индексы;
   - - номер первого отрицательного элемента;

- количество элементов массива, значения которых кратны 5 .

3. Массив целых чисел, состоящий из 15 элементов, задан случайным образом числами из промежутка [-50,55].

   Найти:

   - -сумму положительных элементов, значения которых меньше 10;
   - -номер последнего отрицательного элемента;
   - -индексы тех элементов, значения которых больше значения предыдущего элемента.

4. Массив целых чисел, состоящий из 25 элементов, задан случайным образом числами из промежутка [-30,30].

   Найти:

- -сумму отрицательных элементов;
- -количество тех элементов, значения которых положительны и не превосходят заданного числа A;
- -номера последней пары соседних элементов с разными знаками.

5. С клавиатуры вводится массив из N целых чисел. Определить, сколько среди них встретилось палиндромов. Назовем число палиндромом, если его запись читается одинаково справа налево и слева направо (как, например, 4884, 393, 55, 1).
6. С клавиатуры вводится массив из N целых чисел. Определить, сколько среди них встретилось совершенных чисел. Натуральное число называется совершенным, если оно равно сумме всех своих положительных делителей, кроме самого себя. Например, 28 - совершенное число, т.к. 28=1+2+4+7+14 .
7. С клавиатуры вводится массив из N целых чисел. Определить, сколько среди них встретилось чисел Армстронга. Натуральное число из n цифр является числом Армстронга, если сумма его цифр, возведенных в n-ую степень, равна самому числу (как, например, 153=13+53+33).
8. С клавиатуры вводится массив из N целых чисел. Определить, сколько среди них встретилось x чисел-близнецов. Числами-близнецами считаются пары простых чисел, разность которых равна 2. Например: 3 и 5, 7 и 9, 11 и 13.
9. 1. Массив целых чисел, состоящий из 20 элементов, задан случайным образом числами из промежутка [-45,65].

   Найти:

   -сумму элементов, имеющих нечетное значение;

   -вывести индексы тех элементов, значения которых больше заданного числа A;

-подсчитать количество положительных элементов кратных К.

Числа А и К вводятся с клавиатуры.

10. Массив целых чисел, состоящий из 15 элементов, задан случайным образом числами из промежутка [0,100]. Отсортировать в порядке убывания элементы, стоящие на нечетных местах, методом обмена.
11. Имеются сведения о багаже пассажира (название и вес каждого предмета). Всего предметов N. Выбрать из заданных предметов любые такие, чтобы их суммарный вес не превышал 30 кг. Вывести на печать название и вес выбранных предметов.
12. Массив целых чисел, состоящий из 20 элементов, задан случайным образом числами из промежутка [-15,85].

    Найти:

    -сумму элементов, имеющих нечетные индексы;

    - номер первого отрицательного элемента;

    количество элементов массива, значения которых кратны 5 .
13. Массив целых чисел, состоящий из 15 элементов, задан случайным образом числами из промежутка [0,100].

    Отсортировать в порядке убывания только четные элементы массива методом выбора.
14. Дана последовательность вещественных чисел, содержащая n=15 элементов. Построить из неё новую последовательность, так чтобы в её начале располагались все отрицательные, а затем все положительные числа и нули. Относительный порядок расположения как отрицательных, так и неотрицательных должен быть сохранён.
15. Массив целых чисел, состоящий из 15 элементов, задан случайным образом числами из промежутка [-50,55].

    Найти:

    -сумму положительных элементов, значения которых меньше 10;

    -номер последнего отрицательного элемента;

    -индексы тех элементов, значения которых больше значения предыдущего элемента .
16. Массив целых чисел, состоящий из 15 элементов, задан случайным образом числами из промежутка [0,100]. Отсортировать в порядке убывания элементы, стоящие на нечетных местах, методом обмена.

17. Массив содержит сведения о зарплате сотрудников. Определить, на сколько нужно повысить зарплату сотруднику с минимальной зарплатой, чтобы достичь среднего уровня зарплаты.
18. Массив целых чисел, состоящий из 25 элементов, задан случайным образом числами из промежутка [-30,30].

    Найти:

    -сумму отрицательных элементов;

    -количество тех элементов, значения которых положительны и не превосходят заданного числа A;

    -номера последней пары соседних элементов с разными знаками.
19. Массив целых чисел, состоящий из 10 элементов, является частичноупорядоченным.

    Отсортировать его по возрастанию методом «пузырька», исключив лишние просмотры.
20. Два массива содержат сведения о росте учеников в классе и список этих учеников. Выведите на печать фамилию самого высокого и самого низкого ученика.
21. Массив P целых чисел, состоящий из 20 элементов, задан случайным образом числами из промежутка [-25,30].

    заменить первый отрицательный элемент нулем;

    умножить все элементы, кратные 3, на третий элемент массива;

    из элементов массива P сформировать массив M той же размерности

    по правилу: если номер четный, то M[i]=i*P[i] ,

    если нечетный, то M[i]= -P[i].
22. Массив целых чисел, состоящий из 15 элементов, задан случайным образом числами из промежутка [-100,100].

    Отсортировать в порядке возрастания отрицательные элементы массива методом обмена.
23. Массивы содержат сведения о фамилии и зарплате сотрудников. Определить, на сколько нужно повысить зарплату сотруднику с минимальной зарплатой, чтобы достичь среднего уровня зарплаты.

Лабораторная работа №7 «Двумерные массивы»

1. Имеются сведения о количестве мест и количестве проданных билетов в каждом из 15-ти вагонов поезда. Считая заданной среднюю стоимость

билета, оценить потери от недогрузки поезда, определить число свободных мест в самом незагруженном вагоне и его номер.

2. Имеется учет ежедневных выплат в банке за неделю. Определить среднюю выплату за неделю, напечатать дни, когда выплата была ниже средней. Упорядочить в порядке убывания выплат и выдать на печать список ежедневных выплат.
3. В 6 городах Самарской области взяты пробы воздуха и определено процентное содержание в нем 5 вредных элементов. Вывести на печать название каждого элемента и название города с наименьшим содержанием этого вредного элемента.
4. Известно количество абитуриентов, подавших заявления на каждую из n=8 специальностей института и план приема по каждой специальности. Определить конкурс по каждой специальности отдельно и общий конкурс по институту.
5. Имеются сведения о названиях и ценах на n=5 видов товаров по m=6 магазинам. Требуется составить список 3 магазинов, имеющих минимальные цены по заданному виду товара. Список должен включать название магазина, название и цену выбранного товара.
6. Каждое из N фермерских хозяйств представило свой перечень из M машин разных наименований на приобретение их в единственном экземпляре (M и N заданы). Составить общий перечень необходимых машин с указанием их количества, расположив список в порядке убывания потребности в них.
7. Каждое из N предприятий выпускает 5 видов товаров (одинаковых для всех предприятий). Известны названия предприятий и товаров, стоимость единицы и общий объем выпуска товаров каждого вида на каждом предприятии. Для каждого предприятия определить вид произведенного товара с максимальной общей стоимостью.
8. В автопарке 10 шоферов, о каждом из которых известно: ежемесячное количество рейсов и ежемесячное количество порожних рейсов в течение года. Для каждого шофера определить общий процент порожних рейсов за год и номер месяца с максимальным количеством порожних рейсов.
9. Список участниц конкурса красоты содержит 10 фамилий. Каждый из 5 судей ставит оценки претенденткам. Вывести на экран список участниц, занявших 1, 2 и 3 места (по сумме баллов).

10. Известны среднемесячные температуры за год. Вывести на экран список названий месяцев, в которых средняя температура была выше 7 градусов. Список месяцев расположить в порядке убывания их среднемесячных температур.
11. Дана прямоугольная матрица вещественных чисел размером N*M (N,M-константы). Требуется уменьшить на 1 все числа в тех строках матрицы, которые не содержат отрицательных чисел.
12. Дана квадратная матрица целых чисел размером N*N (N – константа),заданная случайным образом, числами из промежутка от –15 до 75. Удалить все столбцы матрицы, которые содержат нули.
13. Дана прямоугольная матрица целых чисел размером N*M (N,M-константы). Вставить строку из нулей перед всеми строками, первый элемент которых делится на 3.
14. Дана прямоугольная матрица вещественных чисел размером N*M (N,M-константы). Требуется прибавить 1 ко всем числам в тех столбцах матрицы, в которых есть хотя бы одно нулевое значение.
15. Дана квадратная матрица целых чисел размером N*N (N – константа),заданная случайным образом, числами из промежутка от –10 до 5. Удалить все строки матрицы, которые содержат нули.
16. Дана прямоугольная матрица целых чисел размером N*M (N,M-константы). Вставить второй столбец перед всеми столбцами, в которых нет отрицательных элементов.
17. Дана прямоугольная матрица вещественных чисел размером N*M (N,M-константы). Требуется заменить минимальный элемент в каждой строке матрицы на противоположный по знаку.
18. Дана квадратная матрица целых чисел размером N*N (N – константа),заданная случайным образом числами из промежутка от –5 до 15. Удалить все столбцы матрицы, которые содержат максимальный элемент.
19. Дана прямоугольная матрица целых чисел размером N*M (N,M-константы). Вставить первую строку между средними строками.
20. Дана прямоугольная матрица вещественных чисел размером N*M (N,M-константы). Требуется изменить знак на обратный у всех чисел тех столбцов матрицы, которые содержат хотя бы одно отрицательное число.

21. Дана квадратная матрица целых чисел размером N*N (N – константа),заданная случайным образом, числами из промежутка от –10 до 10. Удалить все строки матрицы, которые содержат минимальный элемент.
22. Дана прямоугольная матрица целых чисел размером N*M (N,M-константы). Вставить столбец из нулей после столбцов с минимальными элементами.
23. Дана прямоугольная матрица вещественных чисел размером N*M (N,M-константы). Требуется увеличить в два раза все числа в тех строках матрицы, которые содержат только положительные числа.
24. Дана квадратная матрица целых чисел размером N*N (N – константа), заданная случайным образом, числами из промежутка от –10 до 15. Удалить все столбцы матрицы, у которых сумма элементов по столбцам максимальна.
25. Дана прямоугольная матрица целых чисел размером N*M (N,M-константы). Поменять местами в каждой строке первый отрицательный и последний положительный элементы.
26. Дана прямоугольная матрица вещественных чисел размером N*M (N,M-константы). Требуется уменьшить в два раза все числа в тех столбцах матрицы, которые содержат хотя бы одно положительное число.
27. Дана квадратная матрица целых чисел размером N*N (N – константа),заданная случайным образом, числами из промежутка от –10 до 15. Удалить все столбцы матрицы, у которых сумма элементов по строкам минимальна.
28. Дана прямоугольная матрица целых чисел размером N*M (N,M-константы). Поменять местами первый столбец и столбец, в котором находится последний нулевой элемент.

Лабораторная работа №8 «Операции над строками»

1. Дана строка символов до точки. Группы символов в ней между группами пробелов считаются словами. Посчитать, сколько слов содержит данная строка.
2. Дана строка символов. Удвойте в заданном тексте все буквы. Пробелы и знаки препинания оставьте без изменения.
3. Введена строка символов. Проверить правильность написания сочетаний «ЧА» и «ЩА». Если надо, то исправить ошибки их написания. Полученную строку вывести на экран дисплея.

4. Дана строка символов. В заданной последовательности найдите все слова, имеющие заданное окончание.
5. Дана строка символов. Замените в заданном тексте все сочетания «min» на «мах».
6. Введена строка символов. Проверить правильность написания сочетаний «ЧУ» и «ЩУ». Если надо, то исправить ошибки их написания. Полученную строку вывести на экран дисплея.
7. Дана строка символов. В заданной последовательности найдите все слова, начинающиеся с заданной приставки.
8. Дана строка символов. Посчитать, сколько предложений содержит данная строка.
9. Введена строка символов. Проверить правильность написания сочетаний «ЖИ» и «ШИ». Если надо, то исправить ошибки их написания. Полученную строку вывести на экран дисплея.
10. Дана строка символов. В заданной последовательности найдите все повторяющиеся слова.
11. Дана строка символов. В заданной последовательности каждые n символов отделите знаком «!».
12. Введена строка символов. Удалите из нее все кратные рядом стоящие одинаковые символы, оставив по одному. Например: МАААММАААА=>МАМА.
13. Дана строка символов. Словом текста считается любая последовательность букв кириллицы. Вывести все слова, в которых гласные буквы образуют палиндром. Расположение гласных может быть произвольным. Малые и большие буквы считаем эквивалентными. Например: слово- Африка, абракадабра. Симметрия: (а,и,а), (а,а,а,а,а).
14. Дана строка символов. Словом текста считается любая последовательность букв латинского алфавита. Найти и вывести в лексикографическом порядке все слова минимальной длины.
15. Дана строка символов. Словом текста считается любая последовательность цифр. Удалить все симметричные слова (палиндромы). Слова из одной цифры палиндромами не считаются.
16. Дана строка символов. Словом текста считается любая последовательность букв латинского алфавита. Найти и вывести в

обратном лексикографическом порядке все слова, содержащие более двух букв «а».

17. Дана строка символов. Словом текста считается любая последовательность букв латинского алфавита. Найти и вывести на печать самое длинное слово палиндром.
18. Дана строка символов. Словом текста считается любая последовательность букв кириллицы. Найти и вывести на печать в лексикографическом порядке все слова, начинающиеся на гласную букву.
19. Дана строка символов. Словом текста считается любая последовательность букв и цифр. Удалить из неё каждое нечетное слово нечетной длины.
20. Дана строка символов. Словом текста считается любая последовательность букв и цифр. Вывести в обратном лексикографическом порядке все слова, встречающиеся в тексте по одному разу.
21. Дана строка символов. Словом текста считается любая последовательность букв и цифр. Поменять местами первое и последнее слово в строке.
22. Дана строка символов. Словом текста считается любая последовательность букв и цифр. Найти и вывести на печать в лексикографическом порядке все слова, в которых встречается буква «а».
23. Введена строка символов. Определить длину самой длинной подстроки из подряд стоящих букв «и».
24. Введена строка символов. Группу символов, разделенную с одной или обеих сторон одним или несколькими пробелами и не содержащую внутри себя пробелов, назовем словом. «Перевернуть» каждое слово, сохранив неизменным их порядок в строке.
25. Введена строка символов. Распечатать все слова четной длины, отличные от первого слова. Если таких слов нет, то выдать соответствующее текстовое сообщение
26. Введена строка символов. Распечатать по алфавиту все слова нечетной длины, в которых нет буквы «К». Если таких слов нет, то выдать соответствующее текстовое сообщение

27. Введена строка символов. Распечатать все слова нечетной длины, имеющие не менее двух букв «а». Если таких слов нет, то выдать соответствующее текстовое сообщение.
28. Введена строка символов. Напечатать в алфавитном порядке только те слова, которые начинаются на гласную букву. Если таких слов нет, то выдать соответствующее текстовое сообщение.
29. Введена строка символов. Напечатать только те слова, которые начинаются на гласную букву. Если таких слов нет, то выдать соответствующее текстовое сообщение.
30. Введена строка символов. «Перевернуть» каждое слово, сохранив неизменным их порядок в строке.
31. Введена строка символов. Распечатать все слова четной длины, в которых нет буквы «К». Если таких слов нет, то выдать соответствующее текстовое сообщение.
32. Введена строка символов. Распечатать в лексикографическом порядке все слова четной длины. Если таких слов нет, то выдать соответствующее текстовое сообщение.

Лабораторная работа №9 «Записи»

Задача 1.

Задан массив записей со сведениями о студентах.

Определить в программе следующий тип записи о студенте:

```
type
    TPol= (Man,Woman);
    Tmark= 2..5;
   TSTUD = record
             Fam: string[30];
           Name: string[20];
             Day: 1..31;
             Mon: 1..12;
             Year:1900..3000;
           Pol: TPol;
             Matem: Tmark;
           Info: Tmark;
  Fizica: Tmark;
       end;
```

Подсчитать и вывести на экран:

1 вариант: количество студенток и студентов.

2 вариант: количество отличников по каждому предмету.

3 вариант: количество студентов с именами «Иван» и «Мария».

4 вариант: количество студентов, родившихся в заданном году (год вводится с клавиатуры).

Задача 2.

Багаж пассажира характеризуется ФИО пассажира, количеством вещей и общим весом вещей. Задать массив записей, содержащий сведения о багаже всех пассажиров.

Подсчитать и вывести на экран:

1 вариант: найдите пассажиров, у которых средний вес одной вещи не более 10 кг.

2 вариант: найдите число пассажиров, у которых количество вещей превосходит среднее число вещей.

3 вариант: найдите всех пассажиров, у которых багаж состоит из 1 вещи весом менее 100 кг, но более 10 кг.

4 вариант: найдите число пассажиров, у которых более 2 вещей общим весом менее 5 кг.

Задача 3.

Дан массив записей, в котором хранятся данные о расписании поездов: номер поезда, название (откуда-куда, например, Обнинск-Москва), время отправления и время прибытия на станцию(часы, минуты).

Подсчитать и вывести на экран:

1 вариант: название всех поездов, которые отправляются из Москвы с 10 до 12 часов.

2 вариант: по заданному времени определить, какие из поездов находятся в данный момент в пути.

3 вариант: название всех поездов, которые прибывают в Москву с 22 до 24 часов.

4 вариант: по заданному времени и номеру поезда определить, где в данный момент он находится(на станции отправления, в пути или на станции прибытия).

Вариант 1

Описать структуру с именем STUDENT, содержащую следующие поля:

- NAME – фамилия и инициалы;
- GROUP – номер группы;

- SES – успеваемость (массив из 5 элементов).

Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив STUD1, состоящий из 10 структур типа STUDENT;
- сортировка записей по возрастанию содержимого поля GROUP;
- вывод на дисплей фамилий и номеров групп для всех студентов, включенных в массив, если средний балл студента больше 4.0;если таких студентов нет, вывести соответствующее сообщение.

Вариант 2

Описать структуру с именем AEROFLOT, содержащую следующие поля:

- NAZN – название пункта назначения рейса;
- NUMR – номер рейса;
- TIP – тип самолета.

Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив AIRPORT, состоящий из 7 структур типа AEROFLOT;
- сортировка записей по возрастанию номера рейса;
- вывод на дисплей номеров рейсов и типов самолетов, вылетающих в пункт назначения, название которого введено с клавиатуры; если таких рейсов нет, вывести соответствующее сообщение.

Вариант 3

Описать структуру с именем WORKER, содержащую следующие поля:

- NAME – фамилия и инициалы работника;
- POST – название занимаемой должности;
- YEAR – год поступления на работу.

Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив TABL, состоящий из 10 структур типа WORKER,
- сортирует записи в алфавитном порядке фамилий;
- вывод на дисплей фамилий работников, чей стаж работы превышает значение, введенное с клавиатуры; если таких работников нет, вывести соответствующее сообщение.

Вариант 4

Описать структуру с именем TRAIN, содержащую следующие поля:

- NAZN – название пункта назначения;
- NUMR – номер поезда;
- TIME – время отправления.

Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив RASP, состоящий из 8 структур типа TRAIN;
- Сортировка записей в алфавитном порядке по названиям пунктов назначения;
- вывод на дисплей информации о поездах, отправляющихся после введенного с клавиатуры времени; если таких поездов нет, вывести соответствующее сообщение.

Вариант 5

Описать структуру с именем MARSH, содержащую следующие поля:

- BEGST – название начального пункта маршрута;
- TERM – название конечного пункта маршрута;
- NUMER – номер маршрута.

Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив TRAFIC, состоящий из 8 структур типа MARSH;
- сортировка записей по номерам маршрутов;
- вывод на дисплей информации о маршруте, номер которого введен с клавиатуры; если таких маршрутов нет, вывести соответствующее сообщение.

Вариант 6

Описать структуру с именем NOTE, содержащую следующие поля:

- NAME – фамилия, имя;
- TELE – номер телефона;
- BDAY – день рождения (массив из трех чисел).

Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив BLOCKNOTE, состоящий из 8 структур типа NOTE;
- сортировать записи по датам дней рождения;
- вывод на дисплей информации о человеке, номер телефона которого введен с клавиатуры; если такого нет, вывести соответствующее сообщение.

Вариант 7

Описать структуру с именем ZNAK, содержащую следующие поля:

- NAME – фамилия, имя;

- ZODIAC – знак Зодиака;
- BDAY – день рождения (массив из трех чисел).

Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив BOOK, состоящий из 8 структур типа ZNAK;
- сортировать записи по датам дней рождения;
- вывод на дисплей информации о человеке, чья фамилия введена с клавиатуры;
- если такого нет, вывести соответствующее сообщение.

Вариант 8

Описать структуру с именем PRICE, содержащую следующие поля:

- TOVAR – название товара;
- MAG – название магазина, в котором продается товар;
- STOIM – стоимость товара в рублях.

Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив SPISOK, состоящий из 8 структур типа PRICE;
- сортировать записи в алфавитном порядке по названиям товаров;
- вывод на дисплей информации о товаре, название которого введено с клавиатуры; если таких товаров нет, вывести соответствующее сообщение.

Вариант 9

Описать структуру с именем ORDER, содержащую следующие поля:

- PLAT – расчетный счет плательщика;
- POL – расчетный счет получателя;
- SUMMA – перечисляемая сумма в рублях.

Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив SPISOK, состоящий из 8 структур типа ORDER;
- сортировать записи по расчетным счетам плательщиков;
- вывод на дисплей информации о сумме, снятой с расчетного счета плательщика, введенного с клавиатуры; если такого расчетного счета нет, вывести соответствующее сообщение.

Лабораторная работа №10 «Текстовые и типизированные файлы»

```
program newf;
const n=10;
var
     f1: file of integer;           {файл исходных данных }
```

```
     f2: file of integer;             { файл результатов  }
     i,a:integer;
begin
     assign( f1,'a.dat');
     assign( f2,'b.dat');
 { I. Задаём файл исходных данных }
       rewrite( f1);               {открываем файл для записи }
      for i:=1 to n do
      begin
      readln(a);              {вводим данные с клавиатуры  }
      write( f1,a);           { записываем в файл }
      end;
      close( f1);
{ закрываем файл }
{  II. Читаем данные из файла }
  reset( f1);                 {открываем файл для чтения  }
    while not eof(f1) do
    begin
     read(f1,a);              { читаем из файла }
     write(a,',');
     end;
     close(f);

{     ...........Программа ......}
end.
```

1. Дан файл f, компоненты которого являются целыми числами. Никакая из компонент файла не равна нулю. Файл f содержит столько же отрицательных чисел, сколько и положительных. Переписать компоненты файла f в файл g так, чтобы в файле g сначала шли все положительные числа, а потом все отрицательные. Массивы не использовать.
2. Даны два упорядоченных по возрастанию файла целых чисел. Получить один файл-результат, в котором присутствуют все числа из исходных файлов.
3. Дан файл f, компоненты которого являются целыми числами. Никакая из компонент файла не равна нулю. Файл f содержит столько же отрицательных чисел, сколько и положительных. Используя вспомогательный файл h, переписать

компоненты файла f в файл g так, чтобы в файле g числа шли в следующем порядке: два положительных, два отрицательных, два положительных, два отрицательных и т. д.

4. Даны два упорядоченных по возрастанию файла символов. Получить один файл-результат, в котором присутствуют все буквы из исходных файлов.
5. Дан файл f, компоненты которого являются целыми числами. Никакая из компонент файла не равна нулю. Числа в файле идут в следующем порядке: десять положительных, десять отрицательных, десять положительных, десять отрицательных и т. д. Переписать компоненты файла f в файл g так, чтобы в файле g числа шли в следующем порядке: пять положительных, пять отрицательных, пять положительных, пять отрицательных и т. д.
6. Даны два упорядоченных по убыванию файла целых чисел. Получить один файл-результат, в котором присутствуют все числа из исходных файлов.
7. Дан текстовый файл, в котором хранятся данные об учениках класса: фамилия, имя, отчество, адрес (улица, дом, квартира) и домашний телефон (если есть). Вывести на экран фамилию, имя и адрес только тех учеников, до которых нельзя дозвониться. Создайте текстовый файл данных в текстовом редакторе BP.
8. Создайте типизированный файл записей, отражающих результаты сессии: фамилия, группа, три предмета, три оценки. По запросу об отличниках сессии выдайте список по алфавиту фамилий студентов по каждой группе, сдавших экзамены на «отлично».
9. Дан текстовый файл, в котором хранятся данные о работающих на фирме: фамилия, имя, отчество, адрес (улица, дом, квартира) и дата поступления на работу (месяц, год). Вывести на экран данные тех из них, кто на сегодняшний день проработал уже не менее 5 лет. Создайте текстовый файл данных в текстовом редакторе BP.
10. Создайте типизированный файл записей об учебной литературе: наименование, автор, учебная дисциплина, число экземпляров, цена. По запросу о наличии учебной литературы по заданной дисциплине выдайте список изданий (по алфавиту), указывая число их экземпляров и общую стоимость издания.
11. Дан текстовый файл, в котором хранятся данные о клиентах пункта проката: фамилия, имя, отчество, адрес (улица, дом, квартира) и что взял (только один предмет). Вывести на экран данные тех из них, кто взял на прокат телевизор. Создайте текстовый файл данных в текстовом редакторе BP.

12. Создайте типизированный файл записей, отражающих результаты сессии: фамилия, группа, три предмета, три оценки. По запросу о лучшей группе выдайте упорядоченный (в порядке убывания) список групп, которые имеют наивысший суммарный средний балл по всем экзаменам.
13. Дан текстовый файл, в котором хранятся данные о клиентах автостоянки: фамилия, имя, отчество, марка автомобиля, год выпуска автомобиля. Вывести на экран данные тех из них, чей автомобиль старше 10 лет. Создайте текстовый файл данных в текстовом редакторе BP.
14. Создайте типизированный файл записей об учебной литературе: наименование, автор, учебная дисциплина, число экземпляров, цена. По запросу о наличии учебной литературы заданного автора выдайте список всех наименований (по алфавиту), указывая число их экземпляров и общую стоимость издания.

Лабораторная работа №11 Создание простого приложения в Delphi

Интерактивная среда программирования Delphi позволяет создавать различные приложения для операционной системы Windows. После запуска Delphi на экране появляются четыре окна (рис.1): Главное окно, Окно формы, Окно Инспектора объектов и Окно кода программы.

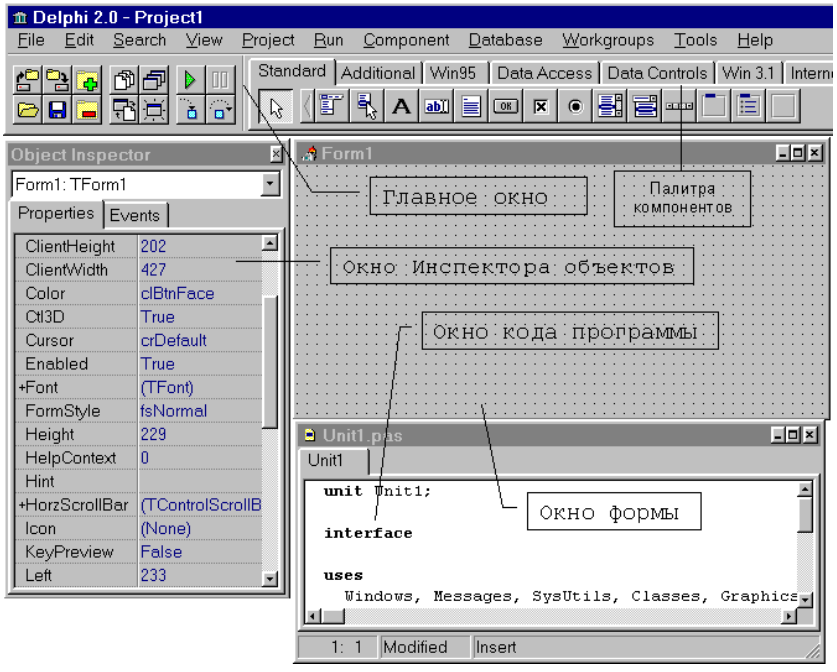


Рис.1. Окна Delphi

Главное окно Delphi содержит в себе главное меню Delphi, набор пиктографических командных кнопок и палитру компонентов. Компонентами в Delphi называются потомки класса `TComponent`.

Окно формы изначально пустое и содержит стандартные для Windows интерфейсные элементы – кнопки вызова системного меню, максимизации, минимизации и закрытия окна, полосу заголовка и очерчивающую рамку.

Окно инспектора объектов предназначено для изменения параметров компонентов.

Окно кода программы предназначено для создания и редактирования текста программы.

Размещать компоненты на форме очень просто. Требуется только щелкнуть на нужной вкладке палитры компонентов, затем на кнопке с пиктограммой соответствующего компонента и после этого щелкнуть в окне формы.

Создадим простейшее приложение, которое реализует следующую постановку задачи: "Ввести значения двух катетов и вычислить значение гипотенузы". Результат работы программы должен соответствовать окну, представленному на рис. 2.

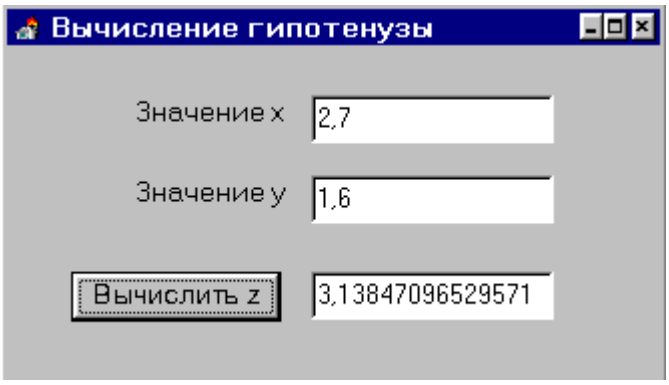


Рис.2.

Для разработки данного приложения необходимо выполнить следующие действия:

1. Выберите страницу `Standard` палитры компонентов.
2. Поместите указатель мыши на компонент с изображением буквы A.
3. Щелкните на кнопке, затем щелкните на вашей форме. Появится надпись `Label1`.
4. Для изменения стандартной надписи `Label1` в строке `Caption` окна Инспектора объектов введите надпись `Значение X`.
5. Для изменения цвета и шрифта надписи щелкните мышью по свойству `Font` окна Инспектора объектов и с помощью кнопки в правой части строки раскройте диалоговое окно настройки шрифта. В списке `Размер` этого окна выберите высоту шрифта, а с помощью списка `Цвет` выберите нужный цвет.
6. Аналогично пунктам 3…5 создайте надпись `Значение Y`.
7. Поместите указатель мыши на компонент `Edit` и создайте на форме три поля `Edit` (`Edit1`, `Edit2`, `Edit3`).
8. Поместите указатель мыши на компонент с изображением кнопки с надписью `OK`.

9. Щелкните по этой кнопке, а затем щелкните на форме. Появится кнопка с надписью `Button1`. Поменяйте название кнопки с помощью окна Инспектора объектов.
10. Измените название формы `Form1` на `Вычисление гипотенузы`.
11. Дважды щелкните мышью по вновь вставленному компоненту (кнопке). В ответ Delphi активизирует окно кода программы с процедурой
    ```
    procedure TForm1.Button1Click(Sender: TObject);
    ```
12. Напишите в пустой процедуре фрагмент программы ввода исходных данных, вычисления и вывода на экран результатов расчета
    ```
    var x,y,z:real;
    begin
    x:=strtofloat(Edit1.text);
    y:=strtofloat(Edit2.text);
    z:=sqrt(sqr(x)+sqr(y));
    Edit3.text:=floattostr(z);
    end;
    ```
13. Выполните команду `Save Project As…` из меню `File`, указав при этом свою рабочую папку.

В окне кода программы окончательный текст модуля будет выглядеть следующим образом:

```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls,
Forms, Dialogs, StdCtrls;
type
  TForm1 = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    Edit1: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
    Button1: TButton;
    procedure Button1Click(Sender: TObject);
```

```
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
{$R *.DFM}
procedure TForm1.Button1Click(Sender: TObject);
var x,y,z:real;
begin
x:=strtofloat(Edit1.text);
y:=strtofloat(Edit2.text);
z:=sqrt(sqr(x)+sqr(y));
Edit3.text:=floattostr(z);
end;
end.
```

Задания к лабораторной работе №11

№	Расчетные формулы	Исходные данные
1	$s=\left\|x^{y/x}-\sqrt{\frac{y}{x}}\right\|;\quad w=(y-x)\frac{y-\frac{z}{y-x}}{1+(y-x)^2}$	x = 1.82 y = 18 z = -3.29
2	$s=1+x+\frac{x^2}{2}+\frac{x^3}{3}+\frac{x^4}{4};w=x(\sin x+\cos y)$	x = 0.33 y = 0.02
3	$y=e^{-bt}\ \sin(at+b)-\sqrt{\lvert bt+a\rvert};\ s=b\sin\left(at^2\cos(at)\right)-1$	a = -0.5 b = 1.7 t = 0.44
4	$w=\sqrt{x^2+b}-\frac{b^2\sin^3(x+a)}{x};\ y=\cos^2(x^3)-x/\sqrt{a^2+b^2}$	a = -0.5 b = 15.5 x = -2.9
5	$s=x^3tg^2((x+b)^2)+\frac{a}{\sqrt{x+b}};\quad g=\frac{bx^2-a}{e^{ax}-1}$	a = 16.5 b = 3.4 x = 0.61

6	$r = \frac{x^2(x+1)}{b} - \sin^2(x+a);\ s = \sqrt{\frac{xb}{a}} + \cos((x+b)^3)$	a = 0.7 b = 0.05 x = 0.5
7	$y = \sin^3((x^2+a)^2) - \sqrt{\frac{x}{b}};\ z = \frac{x^2}{a} + \cos((x+b)^3)$	a = 1.1 b = 0.04 x = 0.2
8	$y = b \cdot tg^2 x - \frac{a}{\sin^2(x/a)};\ d = a \cdot e^{\sqrt{a}} \cdot \cos(bx/a)$	a = 3.2 b = 17.5 x = -4.8
9	$f = \ln(a+x^2) + \sin^2(x/b);\ \ z = e^{-cx} \cdot \frac{x+\sqrt{x+a}}{x - \ln(\lvert x-b \rvert)}$	a = 10.2 x = 2.2 b = 9.2 c = 0.5
10	$y = \frac{a^{2x} + b^{-x} \cdot \cos((a+b)x)}{x+1};\ r = \sqrt{x^2+b} - b^2 \sin(x+a)/x$	a = 0.3 b = 0.9 x = 0.61
11	$z = \sqrt{ax \cdot \sin(2x) + e^{-2x}(x+b)};\ w = \cos^2(x^3) - x/\sqrt{a^2+b^2}$	a = 0.5 b = 3.1 x = 1.4
12	$s = x \cdot ctg^2(x-a) + \frac{b}{\sqrt{x+b}};\ \ w = \frac{bx-a}{e^{b-x}-1}$	a = 16.5 b = 3.4 x = 0.61
13	$u = \frac{a^2 x + e^{-x}\cos(bx)}{bx - e^{-x}\sin(bx) + 1};\ f = e^{2x}\ln(a+x) - b^{3x}\ln(b-x)$	a = 0.5 b = 2.9 x = 0.3
14	$a = \frac{2\cos\left(x - \frac{x}{y}\right)}{1/2 + \sin^2(y)};\quad b = 1 + \frac{z^2}{3 + z^2/5}$	x = 1.42 y = -1.27 z = 3.5
15	$s = e^{-ax}\sqrt{x+1} + e^x\sqrt{x+1{,}5};\ z = \frac{\sin x}{\sqrt{1+m^2\sin^2 x}} - cm \cdot ln(mx)$	a = 0.5 x = 1.7 m = 0.7 c = 2.1

Лабораторная работа №12 Создание приложений с переключателями (ветвление)

Для размещения на форме переключателей используются следующие компоненты страницы Standard:

1. **TCheckBox** – независимый переключатель или флаг выбора ☑;
2. **TRadioButton** – зависимые переключатели ⦿;
3. **TRadioGroup** – контейнер зависимых переключателей.

Наличие выбора у двух первых компонентов определяет логическое свойство `Checked`:

`if CheckBox1.Checked then …` – если флаг выбора установлен,

`if not TradioButton1.Checked then …`– если зависимый переключатель не выбран.

Так как переключатели `TRadioButton` зависимые, то на форме должно быть размещено как минимум два переключателя. Если в одном из компонентов `TRadioButton` свойство `Checked` принимает значение `true`, то у других компонентов оно автоматически принимает значение `false`.

Вместо переключателей `TRadioButton` удобно использовать контейнер зависимых переключателей `TradioGroup`**.** Чтобы создать в контейнере переключатели, следует использовать свойство `Items`. Свойство `ItemIndex` по умолчанию имеет значение `–1` (не выбран ни один переключатель). Если значение `ItemIndex` равно `0`, то выбран первый зависимый переключатель, если равно `1`, то второй и т. д.

Создадим приложение, которое реализует следующую постановку задачи:

Ввести значения двух переменных X и У.

Вычислить значение Z в зависимости от предложенного выбора. Результат работы программы должен соответствовать окну, представленному на рис. 3.

В случае, если не включен флаг выбора, то после нажатия кнопки *`Вычислить Z`* в окне переменной Z (`Edit3`) должна появиться надпись `'Включи флажок!'`

В случае, если не выбран ни один из вариантов, то после нажатия кнопки *`Вычислить Z`* в окне переменной Z (`Edit3`) должна появиться надпись `'Выбери вариант!'`

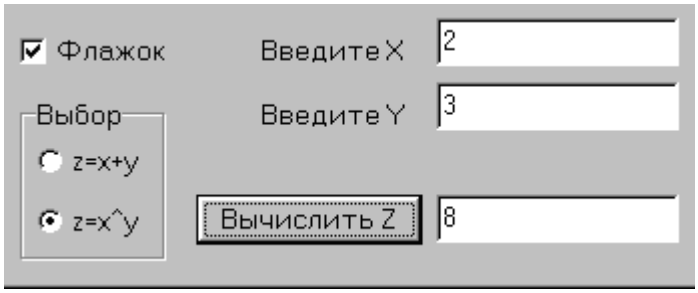


Рис. 3.

Окончательно текст модуля будет выглядеть следующим образом:

```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls,
Forms, Dialogs, StdCtrls, ExtCtrls, math;
type
  TForm1 = class(TForm)
    CheckBox1: TCheckBox;
    RadioGroup1: TRadioGroup;
    Label1: TLabel;
    Label2: TLabel;
    Edit1: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
    Button1: TButton;
    procedure Button1Click(Sender: TObject);
var Form1: TForm1;
implementation
procedure TForm1.Button1Click(Sender: TObject);
var x,y,z:currency;
begin
x:=strtocurr(Edit1.text);
y:=strtocurr(Edit2.text);
if CheckBox1.checked=true then
   begin
       if RadioGroup1.ItemIndex=0 then z:=x+y;
       if RadioGroup1.ItemIndex=1 then z:=power(x,y);
          Edit3.text:=currtostr(z);
       if RadioGroup1.ItemIndex<0 then
          Edit3.text:='Выбери вариант!';
   end
else Edit3.text:='Включи флажок!';
end;
end.
```

1. Задание:

1. Разместить на форме 3 компонента Edit, 3 компонента RadioButton, 2 компонента CheckBox, кнопку Button, а также необходимое для пояснений количество компонентов Label.
2. В первых двух компонентах Edit вводятся значения вещественных переменных x и y.
3. При щелчке на кнопке Button в третьем компоненте Edit отображается результат вычисления с точностью до второго знака после запятой, зависящий от состояния переключателей CheckBox:
   - x+y, если выбран первый переключатель CheckBox,
   - x^y, если выбран второй переключатель CheckBox,
   - если выбраны оба переключателя CheckBox, в компоненте Edit отображаются сумма x+y и степень x^y через пробел,
   - если не выбран ни один из переключателей CheckBox, в компоненте Edit появляется надпись 'Поставь птичку!'.
4. При щелчке на компонентах RadioButton происходят следующие события:
   - при щелчке на первом компоненте RadioButton форма окрашивается в жёлтый цвет,
   - при щелчке на втором компоненте RadioButton форма окрашивается в красный цвет,
   - при щелчке на третьем компоненте RadioButton форма окрашивается в свой первоначальный цвет.

2. Задание:

1. Разместить на форме 2 компонента Edit, компонент SpinEdit (страница Samples), компонент ScrollBar, 2 компонента RadioButton, кнопку Button, а также необходимое для пояснений количество компонентов Label.
2. В первом компоненте Edit ввести значение вещественной переменной x.
3. С помощью компонента SpinEdit ввести значение целой переменной y. Установить диапазон изменения переменной y в пределах 1…10. Начальное значение y установить равным 1.
4. С помощью компонента ScrollBar ввести значение целой переменной z. Установить диапазон изменения переменной z в пределах 1…20,
5. Создать событие на перемещение бегунка ScrollBar, отражающее в компоненте Label изменение величины переменной z.

6. При щелчке на кнопке Button во втором компоненте Edit отображается результат вычисления с точностью до второго знака после запятой, зависящий от состояния переключателей RadioButton:
   - x+y+z, если выбран первый переключатель RadioButton,
   - x^y, если выбран второй переключатель RadioButton.

3. Задание:

1. Разместить на форме 2 компонента Edit, компонент Memo, 3 компонента RadioButton, 2 компонента CheckBox, кнопку Button, а также необходимое для пояснений количество компонентов Label.
2. В компонентах Edit вводятся значения вещественных переменных x и y.
3. При щелчке на кнопке Button в компоненте Memo отображается результат вычисления с точностью до третьего знака после запятой, зависящий от состояния переключателей CheckBox:
   - x+y, если выбран первый переключатель CheckBox,
   - x^y, если выбран второй переключатель CheckBox,
   - если выбраны оба переключателя CheckBox, в компоненте Memo в первой строке отображается сумма x+y, а во второй строке – степень x^y,
   - если не выбран ни один из переключателей CheckBox, в компоненте Memo появляется надпись 'Поставь птичку!'.
4. При щелчке на компонентах RadioButton происходят следующие события:
   - при щелчке на первом компоненте RadioButton форма окрашивается в жёлтый цвет,
   - при щелчке на втором компоненте RadioButton форма окрашивается в зелёный цвет,
   - при щелчке на третьем компоненте RadioButton форма окрашивается в свой первоначальный цвет.

4. Задание:

1. Разместить на форме 2 компонента Edit, 2 компонента CheckBox, 2 кнопки Button, 1 компонент ScrollBar, а также необходимое для пояснений количество компонентов Label.
2. В первом компоненте Edit вводится произвольный текст.
3. При щелчке на первой кнопке Button во втором компоненте Edit, в зависимости от состояния переключателей CheckBox, отображаются:

- все цифры из первого компонента Edit, если выбран первый переключатель CheckBox,
- все маленькие латинские буквы из первого компонента Edit, если выбран второй переключатель CheckBox,
- все цифры и маленькие латинские буквы из первого компонента Edit, если выбраны оба переключателя CheckBox,
- если не выбран ни один из переключателей CheckBox, во втором компоненте Edit появляется надпись 'Поставь птичку!'.

4. У компонента ScrollBar установить диапазон изменения значений в пределах 0…255.
5. Создать событие на перемещение бегунка ScrollBar, изменяющее цвет формы от чёрного до красного.
6. При щелчке на второй кнопке Button форма окрашивается в свой первоначальный цвет.

5. Задание:

1. Разместить на форме 2 компонента Edit, 2 компонента CheckBox, 2 кнопки Button, компонент Memo, компонент SpinEdit (страница Samples), а также необходимое для пояснений количество компонентов Label.
2. В компонентах Edit вводятся значения вещественных переменных x и y.
3. При щелчке на первой кнопке Button в компоненте Memo отображается результат вычисления с точностью до третьего знака после запятой, зависящий от состояния переключателей CheckBox:
   - x+y, если выбран первый переключатель CheckBox,
   - x^y, если выбран второй переключатель CheckBox,
   - если выбраны оба переключателя CheckBox, в компоненте Memo в первой строку отображается сумма x+y, а во второй строку – степень x^y,
   - если не выбран ни один из переключателей CheckBox, в компоненте Memo появляется надпись 'Поставь птичку!'.
4. У компонента SpinEdit установить диапазон изменения значений в пределах 0…255. Начальное значение оставить равным 0. Шаг изменения значений (свойство Increment) установить равным 10.
5. При изменении значения в компоненте SpinEdit происходит следующее событие:
   - форма меняет свой цвет с чёрного до красного,

6. При щелчке на второй кнопке Button форма окрашивается в свой первоначальный цвет.

6. Задание:

1. Разместить на форме компонент Memo, компонент Edit, кнопку Button, 3 компонента RadioButton, 2 компонента ScrollBar, а также необходимое для пояснений количество компонентов Label.
2. В компоненте Memo в первой строке вводится значение вещественной переменной x, а во второй строке – вещественной переменной y.
3. При щелчке на кнопке Button в компоненте Edit отображается результат вычисления с точностью до третьего знака после запятой, зависящий от состояния переключателей RadioButton:
   - x+y, если выбран первый переключатель RadioButton,
   - x*y, если выбран второй переключатель RadioButton,
   - x^y, если выбран третий переключатель RadioButton,
4. У первого компонента ScrollBar установить диапазон изменения значений в пределах 8…16.
5. У второго компонента ScrollBar установить диапазон изменения значений в пределах 0…255.
6. Создать событие на перемещение первого бегунка ScrollBar, изменяющее размер шрифта в компоненте Memo от 8 до 16 (свойство Font.Size).
7. Создать событие на перемещение первого бегунка ScrollBar, изменяющее цвет шрифта в компоненте Memo от чёрного до красного (свойство Font.Color).

7. Задание:

1. Разместить на форме компонент Edit, компонент StringGrid, 1 компонент ScrollBar, 2 компонента CheckBox, кнопку Button, а также необходимое для пояснений количество компонентов Label.
2. В компоненте StringGrid отдельно в каждой ячейке вводится значение вещественных переменных x, y и z.

3. При щелчке на кнопке Button в компоненте Edit отображается результат вычисления с точностью до второго знака после запятой, зависящий от состояния переключателей CheckBox:

- x+y+z, если выбран первый переключатель CheckBox,
- x^y, если выбран второй переключатель CheckBox,
- если выбраны оба переключателя CheckBox, в компоненте Edit отображаются сумма x+y+z и степень x^y через пробел,
- если не выбран ни один из переключателей CheckBox, в компоненте Edit появляется надпись 'Поставь птичку!'.

4. Создать событие на перемещение бегунка ScrollBar, изменяющее цвет формы при перемещении на 1/3 от чёрного до красного, при перемещении на вторую треть от чёрного до зелёного и при перемещении на последнюю треть от чёрного до синего.

8. Задание:

1. Разместить на форме компонент Edit, компонент Memo, компонент StringGrid, 3 компонента RadioButton, 2 компонента CheckBox, кнопку Button, а также необходимое для пояснений количество компонентов Label.
2. В компоненте StringGrid вводятся значения вещественных переменных x и y.
3. В компоненте Edit вводится значение вещественной переменной z.
4. При щелчке на кнопке Button в компоненте Memo отображается результат вычисления с точностью до второго знака после запятой, зависящий от состояния переключателей CheckBox:

- x+y+z, если выбран первый переключатель CheckBox,
- x^y, если выбран второй переключатель CheckBox,
- если выбраны оба переключателя CheckBox, в компоненте Memo отображаются в первой строке сумма x+y+z , а во второй – степень x^y через пробел,
- если не выбран ни один из переключателей CheckBox, в компоненте Memo появляется надпись 'Поставь птичку!'.

5. При щелчке на компонентах RadioButton происходят следующие события:

- при щелчке на первом компоненте RadioButton форма окрашивается в жёлтый цвет,
- при щелчке на втором компоненте RadioButton форма окрашивается в красный цвет,

- при щелчке на третьем компоненте RadioButton форма окрашивается в свой первоначальный цвет.

9. Задание:

1. Разместить на форме компонент Edit, компонент Memo, компонент StringGrid, 3 компонента RadioButton, 2 компонента CheckBox, кнопку Button, а также необходимое для пояснений количество компонентов Label.
2. В компоненте Memo вводятся значения вещественных переменных x и y.
3. В компоненте Edit вводится значение вещественной переменной z.
4. При щелчке на кнопке Button в компоненте StringGrid отображается результат вычисления с точностью до второго знака после запятой, зависящий от состояния переключателей CheckBox:

- x+y+z, если выбран первый переключатель CheckBox,
- x^y, если выбран второй переключатель CheckBox,
- если выбраны оба переключателя CheckBox, в компоненте StringGrid отображаются сумма x+y+z в одной ячейке и степень x^y в другой,
- если не выбран ни один из переключателей CheckBox, в компоненте StringGrid появляется надпись 'Поставь птичку!'.

5. При щелчке на компонентах RadioButton происходят следующие события:

- при щелчке на первом компоненте RadioButton форма окрашивается в жёлтый цвет,
- при щелчке на втором компоненте RadioButton форма окрашивается в красный цвет,
- при щелчке на третьем компоненте RadioButton форма окрашивается в свой первоначальный цвет.

Лабораторная работа №13 Создание приложений с управлением значения числовой величины

Задание

Разместить на форме 3 компонента ScrollBar, 3 компонента Label, 1 компонент ComboBox, 1 компонент Edit, 2 кнопки Button.

Написать программу изменения цвета формы следующими способами:

1. С помощью перемещения трёх бегунков `ScrollBar`. Каждый бегунок изменяет составляющую цвета формы в соответствии с палитрой RGB: при перемещении первого бегунка изменяется от минимума до максимума красный цвет, второго – зелёный, третьего – синий;
2. С помощью выбора одного из трёх стандартных цветов в раскрывающемся списке `ComboBox` (наборы цветов для каждого варианта задания приведены ниже);
3. При нажатии кнопки *Применить к форме* цвету формы присваивается значение целой переменной, введенной в компоненте `Edit1` (необходимо помнить, что значение переменной типа Integer не может превышать 2 147 483 647);
4. При нажатии кнопки *Первоначальный цвет формы* цвет формы меняется на первоначальный (`clBtnFace`).

Результат работы программы представлен на рисунке.

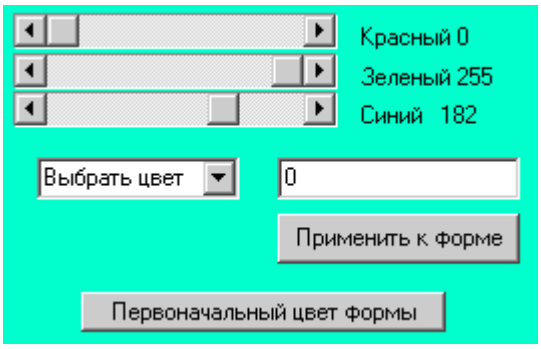


Рис. 4.

Варианты наборов стандартных цветов для списка выбора `ComboBox`:

1.	Yellow (жёлтый цвет) Maroon (густой, насыщенный красно-коричневый цвет) Olive (оливковый цвет)
2.	Purple (фиолетовый цвет) Navy (темно-синий цвет (цвет формы морских офицеров)) Teal (зеленовато-голубой цвет)
3.	Gray (grey – серый цвет) Lime (ярко-зелёный цвет – цвет лайма настоящего (разновидность лимона)) Blue (синий цвет)
4.	Aqua (голубой цвет) MoneyGreen (цвет долларовой купюры) Silver (серебристый цвет)
5.	Cream (светло-желтый цвет (кремовый)) SkyBlue (небесно-голубой цвет)

	Red (красный цвет)
6.	White (белый цвет) Green (зелёный цвет) Fuchsia (фуксин – анилиновый краситель красно-фиолетового цвета)
7.	Red (красный цвет) Yellow (жёлтый цвет) Teal (зеленовато-голубой цвет)

```
    unit Unit1;
    interface
    uses
      Windows, Messages, SysUtils, Classes, Graphics, Controls,
Forms, Dialogs,  StdCtrls, ExtCtrls;
    type
      TForm1 = class(TForm)
        ScrollBar1: TScrollBar;
        ScrollBar2: TScrollBar;
        ScrollBar3: TScrollBar;
        Label1: TLabel;
        Label2: TLabel;
        Label3: TLabel;
        procedure ScrollBar1Change(Sender: TObject);
    private
        { Private declarations }   public
        { Public declarations }    end;
    var   Form1: TForm1;
    implementation {$R *.DFM}
  procedure TForm1.ScrollBar1Change(Sender: TObject);
  begin
  Form1.Color:=RGB(ScrollBar1.Position,ScrollBar2.Position,
  ScrollBar3.Position);
  end;
end.
```

Заметим, что процедура изменения значения величины `ScrollBar1Change` создается только для одного компонента `TscrollBar`. У двух других компонентов в

окне инспектора объектов в разделе Events у события OnChange из списка выбирается значение ScrollBar1Change.

Лабораторная работа №14 Создание редактора RTF-текста

Компонент **TRichEdit** (страница Win32 или Win95 у ранних версий)– редактор RTF-текста. Текст формата RTF хранит информацию, управляющую свойствами каждого абзаца и сменой шрифта по ходу текста. Для хранения атрибутов шрифта используются объекты класса TTextAttributes. Эти атрибуты распространяются на весь текст через свойство редактора DefAttributes или на выделенную часть текста – через свойство SelAttributes. Атрибуты абзаца хранятся в объектах класса TParaAttributes.

Компонент **TSpeedButton** (страница Additional) – кнопка для инструментальных панелей.

Создадим приложение, реализующее работу с текстовым редактором. Редактируемый текст должен содержать такие атрибуты форматирования, как выделенный текст, наклонный текст, подчеркнутый текст.

Фрагмент работы такого приложения представлены на рис. 5.

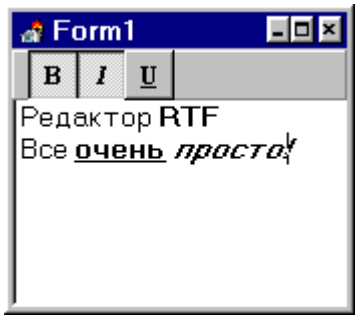


Рис. 5.

Для разработки данного приложения размещаем на форме компонент TRichEdit (страница Win95) – редактор текста в формате RTF. Свойство Align устанавливаем alBottom – заполнение нижней части формы.

В верхней части формы размещаем три кнопки TSpeedButton со страницы Additional. В свойстве Caption первой кнопки задаем латинскую букву B, у второй I, а у третьей – U. В свойстве кнопок Font задаем полужирный шрифт Times размером 10 пунктов, у второй кнопки – наклонный, у третьей – подчеркнутый. Свойство AllowAllUp у кнопок задается True, т.е. утопленную кнопку можно освободить повторным щелчком. Свойство GroupIndex первой кнопки задаем равным 1, у второй 2, а у третьей 3. Таким образом, кнопки не будут принадлежать одной группе.

Окончательно окно формы приложения представлено на рис. 6.

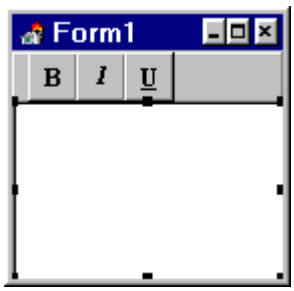


Рис. 6.

Двойным щелчком по кнопкам TSpeedButton создаем три события OnClick и формируем тексты соответствующих процедур. Для кнопки SpeedButton1 (**B**):

```
procedure TForm1.SpeedButton1Click(Sender: TObject);
begin
    CurrText:=RichEdit1.SelAttributes;
    if SpeedButton1.Down then
    CurrText.Style := CurrText.Style + [fsBold];
  else
    CurrText.Style := CurrText.Style - [fsBold];
end;
```

Аналогично для кнопок SpeedButton2 (***I***) и SpeedButton3 (**U**) только параметр [fsBold] заменяется на [fsItalic] и [fsUnderline] соответственно.

Переменная CurrText, предназначенная для хранения атрибутов текста, описывается в разделе private:

```
    CurrText: TTextAttributes;
```

На данном этапе программа будет работать следующим образом. В окне TRichEdit вводится произвольный текст. Нажатием кнопок задаются свойства текста: выделенный, наклонный и полужирный в любой комбинации. Однако в программе пока не реализована одна очень удобная функция редактора Word: при выделении любого фрагмента текста положение кнопок TspeedButton (нажатое или отжатое состояние) показывает наличие соответствующих свойств у выделенного текста. Для реализации этой функции необходимо написать следующую процедуру:

```
procedure TForm1.SelectionChange(Sender: TObject);
begin
SpeedButton1.Down:=fsBold in RichEdit1.SelAttributes.Style;
SpeedButton2.Down:=fsItalic in RichEdit1.SelAttributes.Style;
SpeedButton3.Down:=fsUnderline in RichEdit1.SelAttributes.Style;
end;
```

Данная процедура описывается как метод класса TForm1 (до раздела private) и как событие OnSelectionChange компонента RichEdit1 (в Окне инспектора объектов на странице Events).

Метод SelectionChange должен активизироваться при создании формы. Двойным щелчком по форме создаем событие OnCreate:

```
procedure TForm1.FormCreate(Sender: TObject);
begin
SelectionChange(Self);
end;
```

Запустите программу на выполнение.

Набор тестов для самоконтроля по информатике

Архитектура ЭВМ

1. В структуру ЭВМ фон Неймана входят:

ВАРИАНТЫ ОТВЕТОВ:

а) устройство, выполняющее арифметические и логические операции

б) устройство управления

в) устройство, реализующее взаимодействие компьютеров в сети

г) память для хранения программ и данных

д) устройства для ввода/вывода информации

Правильные ответы:

а, б, г, д

2. Укажите, какие из следующих высказываний являются истинными.

ВАРИАНТЫ ОТВЕТОВ:

а) Появление второго поколения ЭВМ было обусловлено переходом от электронных ламп к транзисторам.

б) В ЭВМ первого поколения отсутствовало устройство управления.

в) В ЭВМ первого поколения отсутствовала оперативная память.

г) Машины третьего поколения — это семейства машин с единой архитектурой, то есть программно совместимых.

д) Компьютер с процессором Intel Pentium III относится к четвёртому поколению ЭВМ.

Правильные ответы:

а, г, д

3. Совокупность ЭВМ и программного обеспечения называется …

Правильный ответ:

вычислительной системой

4. Невозможно случайно стереть информацию на...

Правильный ответ:

CD-ROM

5. Устройством, в котором хранение данных возможно только при включенном питании компьютера, является…

ВАРИАНТЫ ОТВЕТОВ:

а) жесткий магнитный диск

б) оперативная память (ОЗУ)

в) постоянное запоминающее устройство

г) оптический диск

Правильный ответ:

б

6. Для временного хранения информации в персональном компьютере используется...

ВАРИАНТЫ ОТВЕТОВ:

1) операционная система

2) оперативная память (ОЗУ)

3) BIOS

4) ПЗУ

Правильный ответ:

2)

7. К внутренней памяти относятся:

ВАРИАНТЫ ОТВЕТОВ:

а) жёсткие магнитные диски

б) оперативная память

в) постоянная память

г) гибкие магнитные диски

д) кэш-память

Правильные ответы:

б, в, д

8. К внешним запоминающим устройствам (ВЗУ) относятся:

ВАРИАНТЫ ОТВЕТОВ:

а) жесткий диск

б) флэш-память

в) кэш-память

г) регистры

Правильные ответы:

а, б

9. При форматировании гибкий магнитный диск разбивается на …

Правильный ответ:

дорожки и сектора

10. На материнской плате персонального компьютера размещается …

ВАРИАНТЫ ОТВЕТОВ:

1) центральный процессор

2) жесткий диск (винчестер)

3) системный блок

4) блок питания

Правильный ответ:

1)

11. Устройствами вывода данных являются…

ВАРИАНТЫ ОТВЕТОВ:

а) Плоттер

б) Процессор

в) Блок питания

г) Монитор

д) Сканер

Правильные ответы:

а, г

12. ПЗУ является _____ памятью.

Правильный ответ:

энергонезависимой

13. COM - порты компьютера обеспечивают…

Правильный ответ:

синхронную и асинхронную передачу данных

14. Устройствами вывода данных являются…

ВАРИАНТЫ ОТВЕТОВ:

а) привод CD-ROM

б) жёсткий диск

в) монитор

г) сканер

д) лазерный принтер

Правильные ответы:

в, д

15. Устройствами ввода данных являются…

ВАРИАНТЫ ОТВЕТОВ:

а) жёсткий диск

б) джойстик

в) мышь

г) регистры

д) привод CD-ROM

Правильные ответы:

б, в

16. Один из физических каналов ввода/вывода компьютера – разъем – называется…

Правильный ответ:

портом

17. Характеристиками LCD мониторов персонального компьютера являются …

ВАРИАНТЫ ОТВЕТОВ:

a) физический размер экрана

b) угол обзора

c) объем хранимых данных

d) размер точки люминофора

Правильные ответы:

a, b

Информация. Сообщения, данные, сигнал.

1. Характеристика качества информации, заключающаяся в достаточности данных для принятия решений, есть …

ВАРИАНТЫ ОТВЕТОВ:

1) достоверность
2) объективность
3) полнота
4) содержательность

Правильный ответ:

3)

2. Характеристика качества информации _____ характеризует возможность ее получения.

Правильный ответ:

Доступность

3. Для информационной техники предпочтительнее _____ вид сигнала.

Правильный ответ:

цифровой

4. Семантическая мера количества информации определяется…

Правильный ответ:

тезаурусом

5. При проведении классификации информации по ее общественной значимости в списке БУДЕТ ОТСУТСТВОВАТЬ _____ информация

Правильный ответ:

тактильная

6. Вид, в котором данные хранятся, обрабатываются и передаются, называется формой _____ данных

Правильный ответ:

Представления

7. Количество информации, необходимое для определения различий двух равновероятных событий, называется одним...

ВАРИАНТЫ ОТВЕТОВ:

1) бодом

2) байтом

3) битом

4) баллом

Правильный ответ:

3)

8. Минимальное количество байт для двоичного кодирования числа 257_{10} равно...

Правильный ответ:

2

9. Максимальное неотрицательное целое число, кодируемое одним байтом равно…

Правильный ответ:

255_{10}

10. Максимальное шестнадцатеричное число, кодируемое одним байтом равно…

Правильный ответ:

FF

11. Количество информации, содержащееся в некотором сообщении, зависит от …

Правильный ответ:

используемого кода

Алгоритмы.

1. Символом

в блок-схемах алгоритмов обозначается …

ВАРИАНТЫ ОТВЕТОВ:

1) ввод/вывод данных

2) начало цикла

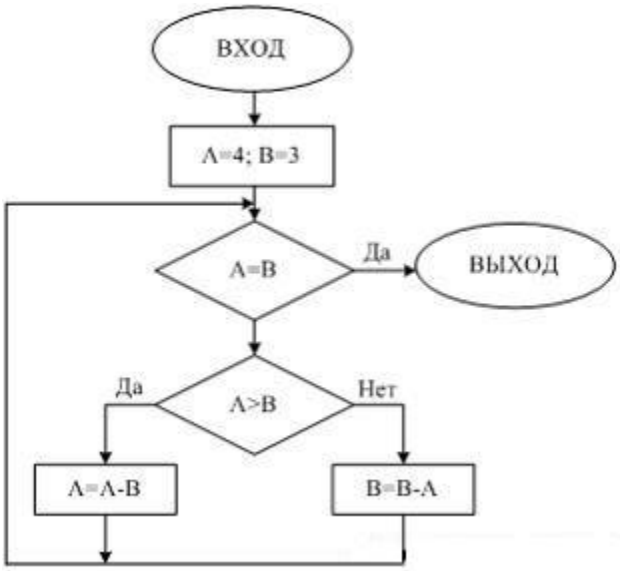
3) начало/конец алгоритма

4) проверка условия

Правильный ответ:

3)

2. В результате работы блок-схемы алгоритма



A и **B** примут значения …

Правильный ответ:

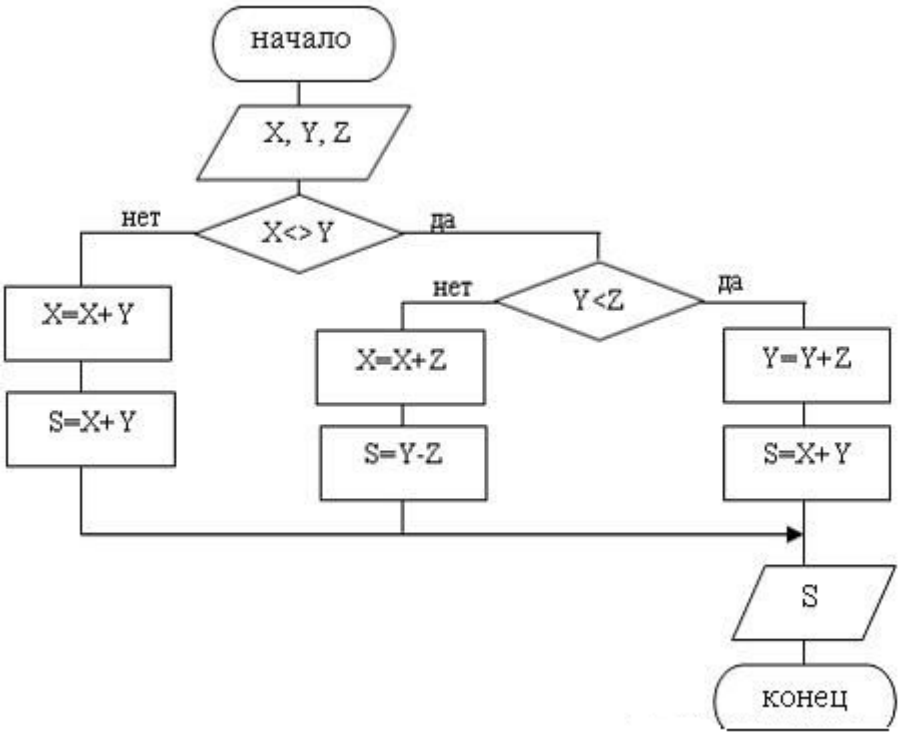
A=1, B=1

3. К свойствам алгоритма относятся...

Правильный ответ:

дискретность, детерминированность

4. Вычисленное по блок-схеме



значение переменной S для входных данных X=1, Y=2, Z=3 равно…

Правильный ответ:

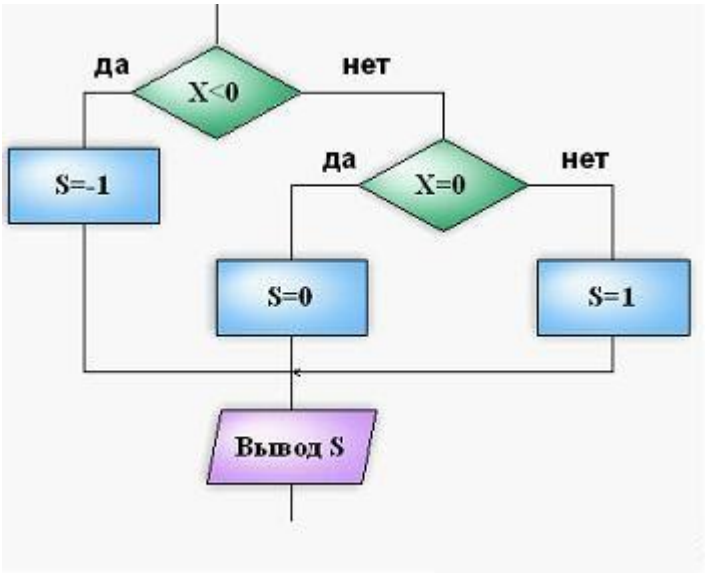
6

5. При разработке программного продукта описание последовательности действий, ведущих к решению поставленной задачи относится к этапу ???.

Правильный ответ:

разработки алгоритма

6. Результатом выполнения алгоритма, представленного фрагментом блок-схемы, для значения переменной X=14, будет следующая величина:…



Правильный ответ:

S=1

7. В алгоритме, вычисляющем произведение отрицательных чисел из N введенных с клавиатуры,

```
read(N);
P:=1
for i:=1 to N do
 Begin
   read(a);
   if (_???__)
    then P:=P*a;
end;
write(P);
```

необходимо вставить условие…

Правильный ответ:

a<0

8. Фрагмент алгоритма:

```
S:=0;
for i:=1 to 10 do
begin
     read(a);
    S:=S+a;
end;
S:=S/10
write(S);
```

выводит…

Правильный ответ:

среднее из десяти чисел, введенных с клавиатуры

9. В результате выполнения алгоритма

```
// «Вывод чисел»
for i:=2 to 6
   write(i);
```

будут выведены числа…

Правильный ответ:

2, 3, 4, 5, 6

10. В результате выполнения алгоритма

```
// «Вычисление значения переменной k»
k := 0;
for i:=2 to 6
    k := k + 1
write(k);
```

значение переменной k будет равно числу …

Правильный ответ:

5

11. Если задан фрагмент алгоритма

```
while a<>b do
begin    if a>b
         then      a=a-b
         else      b=b-a;
end;
write(a);
```

то при заданных начальных условиях a = 375; b=425 после выполнения алгоритма переменная a примет значение …

Правильный ответ:

25

12. Выражение: (a+в)+c = a+(b+c) соответствует _____ закону

1) разместительному

2) распределительному

3) сочетательному

4) переместительному

Правильный ответ:

3)

13. Сложное высказывание, реализованное из двух высказываний A и B посредством логической операцией конъюнкция, истинно тогда и только тогда, когда...

Правильный ответ:

истинны оба высказывания A и B

14. Значение логической функции: «A или 1» равно...

Правильный ответ:

1

15. В результате выполнения алгоритма

A:= 12

B:= 10

A:= 2*A–B

B:= A/2

переменные A и B примут значения…

ВАРИАНТЫ ОТВЕТОВ:

1) A = 12; B = 10

2) A = 14; B = 7

3) A = «2 * A – B»; B = «A / 2»

4) A = 24; B = 12

Правильный ответ:

2)

16. Значение переменной d после выполнения фрагмента алгоритма (операция mod (*x, y*) – получение остатка целочисленного деления *x* на *y*)

```
k := 30
выбор
|   при   mod( k, 12 ) = 7:   d := k;
|   при   mod( k, 12 ) < 5:   d := 2;
|   при   mod( k, 12 ) > 9:   d := 3;
|   иначе                     d := 1;
все
```

равно...

ВАРИАНТЫ ОТВЕТОВ:

1) 30

2) 2

3) 1

4) 3

Системы счисления

1. Выражению: $4_{10} + 8_{10} =$ соответствует запись:

1) $1100_2 + 100_2 =$

2) $1000_2 + 110_2 =$

3) $101_2 + 1000_2 =$

4) $100_2 + 1000_2 =$

Правильный ответ:

4)

2. Два младших разряда двоичной записи числа, кратного 4, имеют вид...

Правильный ответ:

00

3. Десятичному числу 37_{10} соответствует двоичное число...

Правильный ответ:

100101

4. Записанное в шестнадцатеричной системе счисления число $AF,8_{16}$ в десятичной системе будет иметь вид (с точностью до двух знаков после запятой)…

Правильный ответ:

$175,50_{10}$

5. При вычитании из восьмеричного числа 5…6 восьмеричного числа 467, получаем восьмеричное число 107. Это означает, что в уменьшаемом пропущена цифра…

Правильный ответ:

7

6. При вычитании из шестнадцатеричного числа B…C шестнадцатеричного числа AAA, получаем шестнадцатеричное число 152. Это означает, что в уменьшаемом пропущена цифра…

Правильный ответ:

F

Сети

1. Способ организации передачи информации для удалённого доступа к компьютеру с помощью командного интерпретатора называется…

Правильный ответ:

Telnet

2. Информационная или рекламная рассылка, автоматически рассылаемая по списку, без предварительной подписки называется…

Правильный ответ:

спамом

3. Протокол FTP предназначен для...

Правильный ответ:

передачи файлов

4. Укажите варианты беспроводной связи:

ВАРИАНТЫ ОТВЕТОВ:

а) Ethernet

б) Wi-Fi

в) IrDA

г) FDDI

Правильные ответы:

б, в

5. FTP - сервер – это компьютер, на котором...

Правильный ответ:

содержатся файлы, предназначенные для открытого доступа

6. В компьютерных сетях протокол POP3 работает на ??? уровне модели взаимодействия открытых систем.

Правильный ответ:
прикладном

7. Эталонная модель взаимодействия открытых систем OSI имеет ??? уровней.

Правильный ответ:
7

8. Одинаковые ключи для шифрования и дешифрования имеет _____ криптология.

Правильный ответ:
симметричная

9. Электронно-цифровая подпись позволяет ...

Правильный ответ:
удостовериться в истинности отправителя и целостности сообщения

10. Результатом реализации угроз информационной безопасности может быть…

Правильный ответ:
несанкционированный доступ к информации

11. Защита целостности кабельной сети относится к _____ методам защиты информации в сети.

Правильный ответ:
физическим

12. Из перечисленного к средствам компьютерной защиты информации относятся:

ВАРИАНТЫ ОТВЕТОВ:

а) пароли доступа

б) дескрипторы

в) установление прав доступа

г) запрет печати

Правильный ответ:

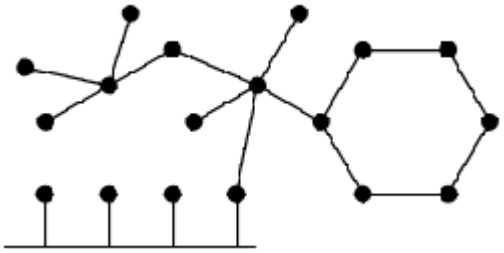
а, в

13. Устройство, выполняющее модуляцию и демодуляцию информационных сигналов при передаче их из ЭВМ в канал связи и при приеме в ЭВМ из канала связи, называется...

Правильный ответ:

модемом

14. Представленная на рисунке сеть



соответствует топологии…

Правильный ответ:

смешанная

15. Маршрутизатором называется…

Правильный ответ:

устройство, соединяющее сети разного типа, но использующие одну операционную систему

16. Канал связи в вычислительной сети – это...

Правильный ответ:

физическая среда передачи информации

17. Задача установления соответствия между символьным именем узла сети Интернет и его IP адресом решается с помощью службы _____ имен.

Правильный ответ:

Доменных

18. Для моделирования работы Интернет используется ??? структурная информационная модель.

Правильный ответ:

Сетевая

19. Сетевым протоколом является...

ВАРИАНТЫ ОТВЕТОВ:

1) программа

2) инструкция

3) набор программ

4) набор правил

20. Представленная на рисунке сеть

соответствует топологии …

ВАРИАНТЫ ОТВЕТОВ:

1) общая шина

2) звезда

3) полносвязная

4) треугольник

21. FTP – это…

ВАРИАНТЫ ОТВЕТОВ:

1) система адресов доменов, содержащих файлы

2) система адресов файловых архивов

3) IP-адреса компьютеров, содержащих файловые архивы

4) имя протокола сети, обслуживающего прием и передачу файлов

22. Программными средствами для защиты информации в компьютерной сети являются:

а) Firewall

б) Brandmauer

в) Sniffer

г) Backup

ВАРИАНТЫ ОТВЕТОВ:

1) в,г

2) а,г

3) а,б

4) б,в

Системное и служебное программное обеспечение. Операционные системы

1. Антивирусные программы, драйверы и архиваторы относятся к _____ программному обеспечению.

Правильный ответ:

служебному (сервисному)

2. Программы архивирования данных относятся к …

Правильный ответ:

сервисному программному обеспечению

3. Организация взаимодействия пользователя с компьютерной системой – это функция …

Правильный ответ:

операционной системы

4. Драйвер – это программа, которая позволяет …

Правильный ответ:

обеспечивать связь между операционной системой и внешними устройствами

5. При необходимости работы пользователя с совокупностью документов, используются ??? интерфейсы.

Правильный ответ:

многооконные

6. Расширение имени файла определяет его ...

Правильный ответ:

тип

7. Если размер кластера 512 байт, а размер файла 816 байт, то файл займет на диске

Правильный ответ:

два кластера

8. Преобразование отсканированного изображения в текстовый формат выполняется программой...

ВАРИАНТЫ ОТВЕТОВ:

1) Fine Reader

2) Acrobat Reader

3) MS Office Document Imagine

4) Ahead Nero

Правильный ответ:

1)

9. При необходимости работы пользователя с совокупностью документов, используются _____ интерфейсы.

ВАРИАНТЫ ОТВЕТОВ:

1) мультизадачные

2) многопользовательские

3) многопотоковые

4) многооконные

Правильный ответ:

4)

Программирование

1. В языках программирования константа – это

   1) величина, не изменяющая свое значение в процессе работы программы

   2) графический значок

   3) функция, всегда возвращающая одинаковое значение в процессе работы программы

   4) метка

Правильный ответ:

1)

2. Результатом компиляции программы на языке высокого уровня является...

Правильный ответ

объектный файл

3. Основное отличие компиляторов от интерпретаторов заключается в том, что…

Правильный ответ:

компиляторы генерируют целевой код, интерпретаторы нет

4. Транслятор, который осуществляет перевод исходной программы в эквивалентную ей результирующую программу на языке машинных команд, называется…

Правильный ответ:

компилятором

5. Исходными данными работы транслятора является…

Правильный ответ:

текст программы на языке программирования высокого уровня

6. Непосредственное покомандное выполнение исходной программы на языке высокого уровня происходит в процессе …

Правильный ответ:
интерпретации

7. Непосредственное покомандное выполнение исходной программы на языке высокого уровня происходит в процессе …

ВАРИАНТЫ ОТВЕТОВ:
1) транзакции
2) компиляции
3) интерпретации
4) идентификации

Правильный ответ:
3)

8. К критериям качества программы можно отнести…

Правильный ответ:
правильность, понятность, гибкость, эффективность

9. Обнаруженное при отладке программы нарушение формы записи программы приводит к сообщению о(б) ??? ошибке.

Правильный ответ:
синтаксической

10. На этапе отладки программы...

Правильный ответ:
проверяется корректность работы программы

11. Языком логического программирования является…

1) Pascal

2) Prolog

3) C++

4) Visual Basic

Правильный ответ:

2)

12. Для системного программирования наиболее подходят языки …

Правильный ответ:

С, С++ и Ассемблер

13. Ассемблер относится языкам ??? типа

Правильный ответ:

машинно-ориентированного

14. Языками декларативного программирования являются языки...

Правильный ответ:

логические

15. К языкам программирования высокого уровня **не относятся**...

Правильный ответ:

АССЕМБЛЕР, С (Си)

16. Верным является высказывание, утверждающее, что…

Правильный ответ:

доступ к элементу массива осуществляется по имени массива и номеру элемента

17. При структурном подходе к составлению программ могут использоваться понятия:

а) альтернативный выбор

б) цикл

с) подпрограмма

д) наследование

Правильный ответ:

а, б, с

18. Структурное программирование по-другому называют программированием без...

Правильный ответ:

GOTO

19. Программирование, основанное на модульной структуре программного продукта и типовых управляющих структурах алгоритмов, называется…

Правильный ответ:

структурным

20. Основой метода структурного программирования являются:

    а) принцип модульности разработки сложных программ

    б) использование композиции трех базовых элементов – линейной, ветвления и циклической структур

    в) использование композиции двух базовых элементов – ветвления и циклической структур

    д) использование большого количества подпрограмм

Правильный ответ:

а, б

21. Параметры, указываемые в момент вызова подпрограммы из основной программы, называются …

Правильный ответ:

фактическими

22. Методика анализа, проектирования и написания приложений с помощью классов, каждый из которых является целостным фрагментом кода и обладает свойствами и методами, называется ??? программированием.

Правильный ответ:
объектно-ориентированным

23. Понятие «наследование» характеризует …

Правильный ответ:
способность объекта сохранять свойства и методы класса-родителя

24. Пусть А базовый класс, В его подкласс. Концепция наследования в объектно-ориентированном подходе подразумевает, что:
    а) объекты класса В наследуют значения объектов класса А
    б) объекты класса В не могут обладать методами класса А без их повторного объявления
    в) общие для классов А и В структуры данных и методы могут быть определены только в классе А
    г) переменные и методы класса А могут быть использованы объектами класса В без их повторного определения в В
    д) в классе В должны быть перечислены наследуемые элементы класса А

Правильный ответ:
в, г

25. **Состояния**, определяющие значения всех переменных ??? и **методы,** определяющие его функциональные возможности, два основных компонента ???

Правильный ответ:
класса

26. Свойством ООП, которое может быть смоделировано с помощью таксономической классификационной схемы (иерархии) является

Правильный ответ:

наследование

27. Приведенный фрагмент программы …

вывод «введите число >0 и <1000»
ввод X
если X < 10
то Y := 1
иначе если X < 100
то Y := 2
иначе Y := 3
конец если

для введенного числа 0 < X < 1000

ВАРИАНТЫ ОТВЕТОВ:

1) проверяет правильность введенного числа

2) находит значение введенного числа

3) удваивает значение введенного числа

4) находит число знаков введенного числа

Правильный ответ:

4)

28. В результате выполнения фрагмента программы

X := 7
Y := 7
P := X=Y
Q := Y>X
P := P OR Q

значения переменных будут равны...

Правильный ответ:

29. P = True; Q = False

После выполнения фрагмента программы

```
X := 5
Z := 7
вывод ( "X=Z  ”, X=Z, ”  X=  ”, Z, Z+X );
```

на печать будет выведено...

Правильный ответ:

X=Z FALSE X= 712

30. Если в программе переменная принимает значение, равное 1.00E02, то она была описана как переменная ??? типа.

Правильный ответ:

вещественного

31. В результате выполнения фрагмента алгоритма

```
ввод X, A, B, C
Y := X^A+B*sin(C)
вывод Y
```

При вводе значений X, A, B, C, равных: 5, 2, 467 и 0 соответственно, значение Y будет равно…

Правильный ответ:

25

32. В результате выполнения фрагмента блок-схемы алгоритма

ввод X, A, B, C
Y := X^3+B*C+A
вывод Y

при вводе значений X, A, B, C, равных: 3, 2048, 2047 и -1 соответственно, значение Y будет равно…

Правильный ответ:
28

33. Значение Y в результате выполнения алгоритма

ввод A, B, C, X
Y := (A+C)/B*X
вывод Y

при вводе значений: 10, 3, 14, 4, будет равно…

Правильный ответ:
32

34. В результате выполнения алгоритма

A := 12
B := 10
A := 2 * A – B
B := A / 2

переменные A и B примут значения…

Правильный ответ:
A = 14; B = 7

35. Круглые скобки для определения порядка выполнения вычислений выражения $a^b*2+3.456\ y$ правильно расставлены в выражении …

Правильный ответ:

((a^b)*2)+(3.456*y)

36. После выполнения алгоритма

b:= 10
d:= 30
нц пока d >= b
| d := d - b
кц

значение переменной d равно...

Правильный ответ:

0

37. Укажите, сколько раз выполнится цикл в представленном фрагменте программы

a:=3; b:=7;
ПОКА (a / 2) ≤ (b / 3)
НЦ
a:=a+2;
b:=b+3;
КЦ;

Правильный ответ:

бесконечное число раз

38. Значение переменной d после выполнения фрагмента алгоритма (операция mod (*x, y*) – получение остатка целочисленного деления *x* на *y*)

k := 30
выбор
| **при** mod(k, 12) = 7: d := k;

| **при** mod(k, 12) < 5: d := 2;

| **при** mod(k, 12) > 9: d := 3;

| **иначе** d := 1;

все

равно...

Правильный ответ:

1

39. Значение переменной d после выполнения фрагмента алгоритма (операции mod (*x, y*) – получение остатка целочисленного деления *x* на *y*, div (*x, y*) –целочисленное деление *x* на *y*)

k := 30

выбор

| **при** div (k, 12) = 4: d := k;

| **при** div(k, 12) < 5: d := 2;

| **при** mod(k, 12) > 9: d := 3;

| **иначе** d := 1;

все

равно...

Правильный ответ:

2

40. После выполнения алгоритма

b:= 10

d:= 50

нц пока d >= b

| d := d - b

кц

значение переменной d равно...

Правильный ответ:

0

41. Деятельность, направленная на исправление ошибок в программной системе, называется …

ВАРИАНТЫ ОТВЕТОВ:

1) отладка

2) тестирование

3) рефакторинг

4) демонстрация

42. Основой метода структурного программирования являются:

а) принцип модульности разработки сложных программ

б) использование композиции трех базовых элементов – линейной, ветвления и циклической структур

в) использование композиции двух базовых элементов – ветвления и циклической структур

д) использование большого количества подпрограмм

ВАРИАНТЫ ОТВЕТОВ:

1) б, д

2) а, б

3) в, д

4) а, в

Электронные таблицы

1. Для того, чтобы формула **=A1*B1**, находящаяся в ячейке **C1** листа Excel, ссылалась на значение **A1** при копировании этой формулы в ячейку **H12**, необходимо...

ВАРИАНТЫ ОТВЕТОВ:

1) скопировать **C1** с помощью меню Правка/Копировать и затем вставить в **H12** с помощью меню Правка/Специальная вставка/кнопка Вставить связь

2) исправить формулу в **C1** на **=A1*B1**

3) исправить формулу в **C1** на **=$A1*$B1**

4) скопировать **C1** с помощью меню Правка/Копировать и затем вставить в **H12** с помощью меню Правка/Специальная вставка/Вставить значение

Правильный ответ:

2)

2. Представлен фрагмент электронной таблицы в режиме отображения формул.

	A	B
1	1	2
2	2	
3		=МАКС(A1:B2;A1+B2;A2+A1)

Значение в ячейке B3 будет равно...

Правильный ответ:

3

3. Представлен фрагмент электронной таблицы в режиме отображения формул.

	A	B
1	3	2
2	4	3
3		=ОСТАТ(A1+B1;A1)

Функция ОСТАТ(X; Y) вычисляет остаток целочисленного деления X на Y.

Значение в ячейке B3 будет равно...

Правильный ответ:

2

4. Ссылка **$A1** (MS Excel) является...

Правильный ответ:

смешанной

5. Из перечисленных функций:

(1) печать текстов

(2) построение диаграмм

(3) создание презентаций

(4) вычисление по формулам

(5) упаковка данных

к основным функциям электронных таблиц относятся…

Правильный ответ:

(2) и (4)

Базы данных

1. Представлена таблица базы данных Студенты.

Студенты : таблица

Номер зачетной книжки	Фамилия	Имя	Отчество
123560	Петров	Сергей	Николаевич
123561	Анисимова	Ольга	Дмитриевна
123564	Белкина	Екатерина	Андреевна
123565	Мишин	Олег	Валерьевич
123568	Иванов	Николай	Петрович

После применения фильтра

Студенты: фильтр

Номер зачетной книжки	Фамилия	Имя	Отчество
	>="А*" And <="М"		

будут отображены записи с фамилиями студентов...

ВАРИАНТЫ ОТВЕТОВ:

1) Анисимова, Белкина, Иванов, Мишин

2) Белкина, Иванов

3) Анисимова, Белкина, Иванов

4) только Анисимова, Мишин

Правильный ответ:

3)

2. Ключевые поля таблицы базы данных содержат данные, которые в этой таблице

ВАРИАНТЫ ОТВЕТОВ:

1) полностью совпадают

2) не повторяются

3) повторяются

4) являются нулевыми

Правильный ответ:

2)

3. Ключ к записям в БД может быть…

а)простым

б)составным

в)первичным

г)внешним

д)дополнительным

е)внутренним

ж)отчётным

з)запросным

Правильный ответ:

а, б, в, г

4. Ключ к записям в БД может быть…

   а)дополнительным

   б)простым

   в)включающим

   г)составным

   д)отчётным

   е)первичным

   ж)запросным

Правильный ответ:

б, г, е,

5. Фильтрация записей в базе данных – это...

Правильный ответ:

отображение в таблице только тех записей, которые соответствуют определенным условиям

6. В СУБД MS Access могут использоваться следующие виды запросов:

а) перекрёстные

б) промежуточные

в) на добавление

г) на выборку

д) на восстановление

Правильный ответ:

а, в, г

7. Отчеты в базе данных Access создаются на основе…

Правильный ответ:

таблиц или запросов

8. После проведения сортировки файла базы данных по убыванию по полю КЛАСС номер строки с фамилией ИВАНОВ будет …

Код	Фамилия	Имя	Класс	Школа
1	Иванов	Петр	10	135
2	Катаев	Сергей	9	195
3	Беляев	Иван	11	45
4	Носов	Антон	7	4

Правильный ответ:

2

9. Ключевое поле предназначено для…

Правильный ответ:

для создания связей между таблицами

10. Базы данных, реализующие сетевую модель данных, представляют зависимые данные в виде…

Правильный ответ:

наборов записей и связей между ними

11. Понятию «отношение» в реляционной базе данных соответствует…

Правильный ответ:

таблица

12. Реальный или представляемый объект, информация о котором должна сохраняться в базе данных и быть доступна называется …

Правильный ответ:

сущностью

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

Управление образовательных программ

_____/ А.В. Дорошин /

"_____"_____ 20___ г.

УТВЕРЖДАЮ

Проректор по учебной работе

_____/ Ф.В. Гречников /

"____" _____ 20___ г.

РАБОЧАЯ ПРОГРАММА

Наименование модуля (дисциплины)

Информатика

Цикл, в рамках которого происходит освоение модуля (дисциплины)	Математический и естественнонаучный цикл
Часть цикла	Базовая часть
Код учебного плана	160400.1.65-2011-О-П-5г06м
Факультет	1
Кафедра	"Общей информатики"

Курс	1		
Семестр	1		
Лекции (СЛ)	16		
Семинарские и практические занятия (СП)	0		
Лабораторные занятия (СЛР)	18	Экзамен	
Контроль самостоятельной работы (КСР)	0	Зачет	1
Самостоятельная работа (СРС)	74		
Всего с экзаменами	108		

Наименование стандарта, на основании которого составлена рабочая программа:

160400 "Проектирование, производство и эксплуатация ракет и ракетно-космических комплексов".

Соответствие содержания рабочей программы, условий ее реализации, материально-технической и учебно-методической обеспеченности учебного процесса по дисциплине всем требованиям государственных стандартов подтверждаем.

Составители:

Стенгач М.С., доцент, к.т.н. _____
(подпись)

Заведующий кафедрой:

Фурсов В.А., профессор, д.т.н.. _____
(подпись)

Рабочая программа обсуждена на заседании кафедры

"Общей информатики"

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

Коды компетенций из ФГОС-3 "Проектирование, производство и эксплуатация ракет и ракетно-космических комплексов" 160400: ОК-14, ОК-17, ПК-1, ПК-4, ПК-14.

1.2 Цели и задачи изучения модуля (дисциплины)

1. Сформировать у студентов представление о принципах устройства и работы ЭВМ,
2. Продемонстрировать эффективность использования вычислительной техники при решении прикладных задач,
3. Дать общее представление о современных информационных технологиях,
4. Подготовить к решению вычислительных задач, которые могут возникнуть на практике.

1.3 Требования к уровню подготовки студента, завершившего изучение данного модуля (дисциплины)

Студенты, завершившие изучение "Информатика", должны знать:
1. Основы архитектуры ЭВМ и принципы функционирования системного и прикладного программного обеспечения.
2. Общие характеристики процессов сбора, передачи, обработки и накопления информации. Технические и программные средства реализации информационных процессов.
3. Типичные подходы к программной реализации сложных приложений.
4. Синтаксис, основные типы данных и программные конструкции алгоритмического языка Pascal объектно-ориентированного стандарта.

Уметь:

- разрабатывать собственные алгоритмы и программно их реализовывать.
- создавать, отлаживать и выполнять программы для вычислительных задач.
- использовать стандартные функции, процедуры и компоненты визуальной среды программирования алгоритмического языка Pascal объектно-ориентированного стандарта.

1.4 Связь с предшествующими модулями (дисциплинами)

Для успешного усвоения данного курса студенты должны владеть знаниями по основам информатики и элементарной математике в объеме средней школы.

1.5 Связь с последующими модулями (дисциплинами)

Курс "Информатика" совместно с другими курсами составляет основу теоретической, фундаментальной подготовки студентов по направлению 160400 "Проектирование, производство и эксплуатация ракет и ракетно-космических комплексов".

2 Содержание рабочей программы (модуля)

Семестр 1		
СЛ 0,1481 16 часов 0,4443 ЗЕТ	Активные 0	
	Интерактивные 0	
	Традиционные 1	Архитектура ЭВМ. Процессор, оперативная память, системный блок, внешние устройства, адаптеры, быстродействие, объемы памяти, конфигурация.
		Основные положения теории информации. Понятие информации. Единицы измерения, хранения и передачи информации. Понятие алгоритма и его свойства. Основные алгоритмические конструкции.
		Алфавит языка Pascal. Структура программы. Комментарии. Идентификаторы. Основные типы данных. Переменные. Оператор присваивания. Основные математические функции. Процедуры ввода и вывода. Константы.
		Логические выражения. Условный оператор IF. Составной оператор. Оператор безусловного перехода GOTO. Оператор выбора CASE.
		Операторы цикла. Оператор цикла WHILE. Оператор цикла REPEAT. Оператор цикла FOR.
		Массивы. Одномерные массивы. Двумерные массивы (матрицы).
		Символьный тип данных. Строки. Основные операции над строками.
		Подпрограммы. Процедуры и функции. Формальные и фактические параметры. Гло-бальные и локальные параметры.
		Оператор TYPE. Простые типы. Порядковые типы данных. Перечисляемый тип данных. Тип-диапазон. Тип дата-время.
		Структурированные типы данных. Записи. Множества. Файлы.
		Модули. Указатели и динамическая память.

СП 0 0 часов 0 ЗЕТ	Активные 1	
	Интерактивные 0	
	Традиционные 0	
СЛР 0,1667 18 часов 0,5001 ЗЕТ	Активные 0	Работа в визуальной среде программирования алгоритмического языка Pascal объектно-ориентированного стандарта. Сложное математическое выражение. Комментарии. Ввод и вывод данных.
		Логические выражения.
		Табулирование функции с одной переменной.
		Работа с одномерными массивами.
		Работа с двумерными массивами
		Работа со строками
		Программа, содержащая функции
		Программа, содержащая процедуры.
		Программа, работающая с переменными типа дата-время
		Обработка файлов.
		Работа с модулями.
	Интерактивные 0	
	Традиционные 0	
КСР 0 0 часов 0 ЗЕТ	Активные 0	
	Интерактивные 0	
	Традиционные 0	
СРС 0,6852 74 часов 2,0556 ЗЕТ	Активные 0,5	Подготовка к зачету.
	Интерактивные 0,5	Подготовка к промежуточному контролю знаний (тестированию в системы дистанционного обучения Moodle) на сайте кафедры http://oi.ssau.ru/m
	Традиционные 0	

3 Инновационные методы обучения

Использование в учебном процессе системы дистанционного обучения Moodle: работа с материалами, размещенными на удалённых сетевых ресурсах, проведение дискуссий, презентаций, конференций и тестирования.

Применение рейтинговой системы оценки знаний студентов.

4 Технические средства и материальное обеспечение учебного процесса

1. Мультимедийный компьютерный класс с объединенными в локальную вычислительную сеть компьютерами, подключенными к глобальной сети.
2. Система дистанционного обучения Moodle.
3. Программное обеспечение Lazarus.

5 Учебно-методическое обеспечение

5.1 Основная литература

1. Цилькер Б.Я. Организация ЭВМ и систем: учебник для вузов. - СПб.: Питер: Питер принт, 2004. - 667 с. (гриф Минобразования России; 112 экземпляров)
2. Фаронов В.В. Delphi. Программирование на языке высокого уровня: учебник для вузов. - СПб.; М.; Нижний Новгород: Питер: Питер принт, 2005. - 639 с. (гриф Минобра-зования России; 5 экземпляров)
3. Фаронов В.В. Система программирования Delphi. - СПб.: БХВ-Петербург, 2005. - 888 с. (5 экземпляров)

5.2 Дополнительная литература

1. Конев Ф.Б. Информатика для инженеров: учебное пособие для вузов. - М.: Высшая школа, 2004. - 272 с.
2. Стенгач М.С. Работа в Delphi. Самара, СГАУ, 2003. - 28 с.
3. Земляной Н.С., Стенгач М.С. Создание приложений в Delphi. Самара, СГАУ, 2003. - 20 с.
4. Стенгач М.С. Статистические и финансовые функции Delphi. Самара, СГАУ, 2004. - 16 с.
5. Культин Н.Б. Delphi в задачах и примерах. - СПб.: БХВ-Петербург, 2005. - 288 с.

5.3 Электронные источники и интернет ресурсы

http://oi.ssau.ru/m - Система дистанционного обучения кафедры общей информатики
http://oi.ssau.ru/uchmetod.html - Учебно-методическое обеспечение кафедры общей информатики

5.4 Методические указания и рекомендации

Промежуточный контроль знаний студентов осуществляется по их текущему выполнению тестов в системе дистанционного обучения и выполнению лабораторных работ .

Итоговый контроль знаний студентов проводится в первом семестре в виде зачета. Критериями получения зачета являются посещаемость студента, выполненные тесты и принятые преподавателем лабораторные работы.