

МИНИСТЕРСТВО ВЫСШЕГО И СРЕДНЕГО  
СПЕЦИАЛЬНОГО ОБРАЗОВАНИЯ РСФСР

Куйбышевский ордена Трудового Красного знамени  
авиационный институт им. академика С.П. Королева

## ВВЕДЕНИЕ В БАЗЫ ДАННЫХ

ПОД ОБЩЕЙ РЕДАКЦИЕЙ ПРОФ В.А. КОМАРОВА,  
ДОЦ. А.В. СОЛОВОВА

Утверждено редакционным  
советом института  
в качестве методических  
указаний к лабораторной  
работе № II

Куйбышев 1984

Министерство высшего и среднего специального  
образования РСФСР

Куйбышевский ордена Трудового Красного Знамени  
авиационный институт имени академика С.П. Королева

ВВЕДЕНИЕ В БАЗЫ ДАННЫХ

Куйбышев 1984

В работе рассмотрены теоретические основы баз данных, знание которых необходимо инженеру-пользователю САПР; дано описание пакета программных модулей БАНК-2, предназначенного для создания и эксплуатации баз данных на ЭММ типа СМ-2; приведена структура базы данных СТУДЕНТ и предложена схема индивидуального задания для работы с этой базой данных.

Методические указания разработаны на кафедре конструкции и проектирования летательных аппаратов и предназначены для студентов 0535 специальности, изучающих курс "Основы САПР", а также могут быть использованы при обучении студентов других машиностроительных специальностей института, на ЦК инженеров и преподавателей.

Составители: Р.Н. Муртазин, А.В. Соловов

Рецензенты: В.А. Цеголев, кафедра технической кибернетики  
Куйбышевского авиационного института

**ц е л ь р а б о т ы:** познакомить студентов с принципами построения информационного обеспечения САПР – базами данных и программными средствами для управления ими. Привить некоторые навыки работы с системами управления базами данных.

**Порядок проведения занятия:** предварительно, во внеаудиторное время, студенты изучают данное руководство и выполняют индивидуальное задание, выданное на предыдущем занятии. Аудиторное занятие проводится в два этапа: I этап (2 ч.) – по учебному расписанию; II этап (0,5ч.) – по скользящему графику.

Первый этап аудиторного занятия имеет примерно следующую схему:  
проверка выполнения индивидуального задания;  
демонстрация работы с базой данных СТУДЕНТ на ЭВМ СМ-2;  
программированный контроль знаний для допуска ко II этапу;  
составление скользящего графика для выполнения II этапа.

Второй этап аудиторного занятия студенты выполняют по скользящему графику:

в установленное время студент является на ВЦ;  
под руководством системного программиста производится открытие базы данных СТУДЕНТ;  
в режиме диалога с ЭВМ производится поиск необходимой информации по ранее подготовленным запросам.

После окончания всей работы каждый студент оформляет отчет, который сдает на следующем аудиторном занятии.

## I. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ РАБОТЫ

### I.1. История появления баз данных

До конца 60-х годов в сфере обработки информации говорили о файлах данных и о наборах данных, а разработку информационных систем

вели в интересах решения отдельных задач, которые практически никак не были связаны. О физической организации (т.е. о физическом размещении информации на конкретных машинных носителях) приходилось заботиться при написании прикладных программ. Если структура данных или запоминающие устройства изменялись, то программист должен был соответствующим образом изменять и свои программы, заново их отлаживать.

Развитие системного программного обеспечения, в частности системы ввода-вывода, позволило создать определенную независимость прикладных программ по отношению к физическому размещению данных, но сохранялось еще такое положение, когда каждый программист для решения новой задачи создавал новые информационные массивы. Это делалось несмотря на то, что в большинстве случаев в существующих наборах данных имелась вся информация (или большая ее часть) для новой задачи.

Рост числа решаемых на ЭВМ задач и, как следствие, увеличение объемов потребной информации до гигантских размеров, обусловили необходимость создания таких информационных систем, которые обеспечивали бы централизованное хранение данных и коллективное их использование. Такие системы стали называть базами данных. дальнейшее их развитие привело к созданию банков данных.

## 1.2. Основные понятия, термины и определения

Элемент данных представляет собой наименьшую единицу данных. Характеризуется именем и может состоять из любого количества байтов или машинных слов. Доступ к элементу производится по его имени.

Запись представляет собой совокупность элементов данных. Характеризуется именем.

Ключ записи состоит из набора элементов, используемых для идентификации записи. Различают основной и дополнительный ключи. Основной ключ однозначно определяет запись в информационном поле. Дополнительный ключ идентифицирует все записи, обладающие определенным свойством.

Файл - это совокупность всех логических записей заданного типа.

Физическое описание данных определяет способ физического хранения данных во внешней памяти ЭВМ. Например, для данных, хранящихся на магнитных дисках, должны быть определены номер цилиндра, номер дорожки, смещение от начала дорожки и длина данных в байтах или машинных словах.

Логическое описание данных указывает на вид, в котором данные представляются программисту или пользователю. Например, последовательность байтов может быть определена конкретным программистом следующим

образом: первые 30 байтов содержат фамилию студента, следующие 20 байтов – имя, следующие 30 байтов – отчество и т.д.

Физическая независимость данных определяет следующее обстоятельство: аппаратура внешних запоминающих устройств (ВЗУ) и способы физического хранения данных могут изменяться без исправления прикладных программ.

Логическая независимость данных означает, что добавление новых элементов данных или расширение общих логических структур осуществляется без исправления уже существующих программ.

База данных (БД) определяется как совокупность взаимосвязанных, хранящихся вместе данных, к которым предъявляют следующие требования: минимальная избыточность, которая допускает использование данных оптимальным образом для одного или нескольких приложений;

логическая и физическая независимость данных;

использование общего управляемого способа для добавления, модификации и поиска данных.

Схема базы данных описывает логическую структуру БД. Представляет собой таблицу типов используемых данных, содержит имена элементов и их атрибуты, указывает на существующие связи между элементами. Схема базы данных составляется на специальном входном языке БД, обрабатывается системными программами БД и затем в виде отдельного файла хранится на ВЗУ ЭВМ. Этот файл используется при обработке каждого запроса к БД.

Подсхема базы данных содержит логическое описание тех данных БД, которые использует конкретный прикладной программист. На основе одной схемы базы данных можно составить множество различных подсхем.

Администратор базы данных – это специалист, отвечающий за создание и дальнейшее ведение БД. Он составляет описание схемы БД, определяет полномочия категорий пользователей, способы физического размещения данных. С его помощью программисты описывают свои подсхемы.

Системами управления базами данных (СУБД) называют специальное программное обеспечение для организации работы пользователей с информацией, размещенной в БД.

### 1.3. Системы управления базами данных

Требования к СУБД:

1. Получение требуемых логических файлов из имеющихся данных и существующих между ними связей.

2. Обеспечение времени ответа, удовлетворяющего пользователя.

3. Сохранение целостности данных в случае машинных сбоев.

4. Обеспечение безопасности и секретности данных. Под безопасностью данных понимают защиту данных от случайного или преднамеренного доступа к ним лиц, не имеющих на это право, и от неавторизованной модификации данных или их уничтожения. Под секретностью понимают право отдельных лиц или организаций определять, когда, как и какое количество соответствующей информации может быть передано другим лицам или организациям.

При обработке запросов на поиск, запись или модификацию информации СУБД используют схему базы данных и подсхему пользователя. Последовательность работы СУБД при запросе на чтение записи в БД показана на рис.1.

Ниже в соответствии с цифрами в кружках (см.рис.1) приведено описание всех II операций, которые являются наиболее важными в процессе чтения записи.

1. Прикладная программа передает запрос в СУБД на чтение записи. В запросе сообщаются полномочия программиста (доступ к секретным данным, право на модификацию записей и т.п.) и ключ поиска.
2. СУБД по подсхеме прикладной программы А определяет форматы элементов, на которые выдан запрос.
3. СУБД по схеме базы данных определяет элементы данных, которые соответствуют подсхеме и содержатся в запросе.
4. По описанию физической организации СУБД определяет физические адреса данных.
5. По найденным физическим адресам выдается команда операционной системе на считывание данных из базы данных.
6. Операционная система взаимодействует с физической памятью, где хранятся данные.
7. Запрошенные данные передаются из внешней памяти ЭВМ в системные буферы.
8. СУБД выделяет из найденной физической записи логическую запись, запрошенную программой А. Далее, используя подсхему прикладной программы А и схему базы данных, производит необходимые преобразования данных в соответствии с форматами, установленными программистом в подсхеме.
9. СУБД передает данные в рабочую область программы А.
10. СУБД передает в область "состояние" информацию о результате выполнения операции или о возникающих при чтении данных ошибках.

II. Прикладная программа обрабатывает полученные данные.



Р и с. 1. Схема работы СУБД при чтении информации

Режим записи информации от прикладной программы в базу данных будет происходить в той же последовательности, при этом операции 7,8,9 выполняются в обратном порядке.



#### 1.4. Структуры данных

Логическое описание данных часто представляют в виде древовидных и сетевых структур.

Древовидная структура данных определяется как иерархия узлов с полярными связями, в которой самый верхний уровень имеет один узел, называемый корнем, а все остальные узлы связываются с одним и только одним узлом на более высоком уровне по отношению к ним самим.

Сбалансированное дерево — это такая древовидная структура данных, в которой каждый узел имеет одинаковое количество ветвей (рис.2,а).

Несбалансированное дерево — древовидная структура, где каждый узел имеет неодинаковое количество ветвей (рис.2,б)  
В позиций физической организации данных сбалансированное дерево проще, чем несбалансированное.



Рис. 2. Древовидные структуры данных

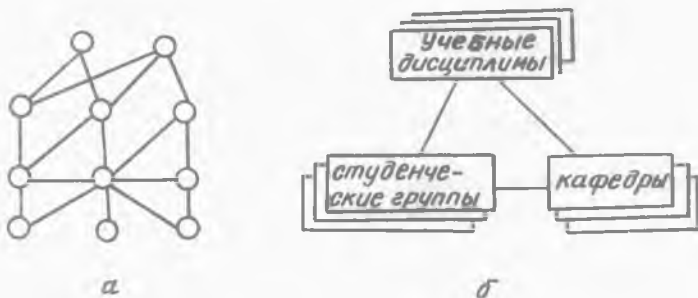
Двоичное дерево — древовидная структура, в которой допускается не более двух ветвей (см. рис.2,а).

Сетевая структура данных определяется как иерархическая структура с порожденными элементами, имеющими более одного исходного элемента (рис.3,а). В сетевой структуре данных могут быть связи не только между элементами разных уровней, но и между элементами одного уровня.

Пример. На фрагменте возможного вида сетевой структуры базы данных института (рис. 3,б) показаны связи: студенческие группы изучают различные учебные дисциплины, которые ведут кафедры института, причем каждая группа студентов обучается на разных кафедрах, а каждая кафедра ведет учебную работу в разных группах.

Необходимо отметить, что с древовидными и сетевыми структурами очень сложно работать при наличии большого количества связей. Так, на рис.3,б показана схема связей только между тремя большими группами данных. Если же попытаться изобразить структуру связей между всеми

учебными дисциплинами, студенческими группами и кафедрами, то полученная схема будет представлять чрезвычайно запутанную картину со множеством перекрестных связей. Вот почему в последнее время все более широкое применение находит так называемая **нормализованная форма представления структур данных**.



Р и с 3. Сетевые структуры данных

Все сложные древовидные и сетевые структуры данных можно привести к простым двумерным таблицам путем введения избыточных элементов данных. Такой процесс называется **нормализацией**, получающиеся при этом таблицы - отношениями, а построенные с использованием таких структур базы данных - **реляционными**. Например, сетевую структуру данных (см. рис.3,б) можно привести к таблицам двух видов, шапки которых показаны ниже.

<i>Дисциплины</i>		
<i>Код</i>	<i>Название</i>	<i>Прочие данные</i>

<i>Кафедры</i>		
<i>Код</i>	<i>Название</i>	<i>Прочие данные</i>

<i>Группы</i>		
<i>Код</i>	<i>Численность</i>	<i>Прочие данные</i>

<i>Дисциплины-группы</i>	
<i>Код дисциплины</i>	<i>Код группы</i>

<i>Дисциплины-кафедры</i>	
<i>Код дисциплины</i>	<i>Код кафедры</i>

<i>Кафедры-группы</i>	
<i>Код кафедры</i>	<i>Код группы</i>

Таблицы первого вида (слева) наполняют содержательной информацией, а таблицы второго вида (справа) отражают связи между различными данными.

Нормализация сложных структур данных позволяет более четко и в простой логической форме определить все многообразие связей между различными данными. Появляющаяся при этом избыточность информации на уровне логического описания данных вовсе не означает, что должна сохраняться и при физическом размещении данных на машинных носителях.

## 2. ПАКЕТ ПРОГРАММНЫХ МОДУЛЕЙ БАНК-2

### 2.1. Назначение и состав пакета

Пакет программных модулей (ППМ) БАНК-2 предназначен для разработки информационно-поисковых систем на ЭВМ типа СМ-2 без программирования процессов создания, модификации, поиска и выдачи данных. ППМ БАНК-2 манипулирует с тремя структурными единицами данных: элементом, записью и массивом.

Э л е м е н т определяется в ППМ БАНК-2 как минимальная единица данных, доступная пользователю. Элемент характеризуется именем и значением. Имя может иметь от 1 до 6 символов. З н а ч е н и я элементов могут быть целого типа (от -32768 до +32767), вещественного типа ( $10^{-32}$  -  $10^{+38}$ ), символьного типа (до 126 символов). З а п и с ь включает в себя один и более элементов и содержит только их значения. Имена элементов данных хранятся в схеме БД. Порядок следования элементов определяется при задании схем БД. М а с с и в состоит из записей одинаковой структуры. В ППМ БАНК-2 используют два типа массивов: управляющие и рабочие. В рабочих массивах хранится основная информация. Однако некоторые элементы этих массивов могут принимать различные значения, и поэтому в схеме БД определяются как ключевые, а в управляющих массивах их значения расширяются.

ППМ БАНК-2 состоит из 4-х программ, выполняющих функции проектирования, установки, аварийного запуска и управления базами данных. Опишем кратко назначение и практические приемы работы с каждой из программ.

### 2.2. Программа проектирования БД

Программа проектирования БД формирует на магнитном диске (ИД) файл со схемой какой-либо создаваемой базы данных. На основе логического описания данных, подготовленного пользователями БД, схему БД разрабатывает администратор БД. Схема включает в себя перечень элементов БД и их тип, элементный состав массивов и наличие связей между

Имя, описание категорий и полномочий различных групп пользователей БД. Описание схемы БД ведется с использованием восьми операторов: ПАКЕТ, УРОВНИ, ЭЛЕМЕНТЫ, МАССИВ, ИМЯ, ЗАПИСЬ, РАБЛЕР, КОНЕЦ. Ниже приведены форматы этих операторов и краткие комментарии.

ПАКЕТ <имя БД> , <пароль> ;

с помощью этого оператора администратор назначает имя БД (не более 6 символов) и ее пароль (целое число в диапазоне от 0 до 32000). В дальнейшем для работы с этой БД необходимо правильно указать ее имя и пароль.

УРОВНИ:

<уровень полномочий> <кодированное слово первой группы пользователей> ,

.....

<уровень полномочий> <кодированное слово последней группы пользователей> ;

данный оператор используется в целях обеспечения секретности и безопасности создаваемой БД. Все пользователи БД разделяются на несколько категорий (не более 15). Для каждой категории пользователей администратор БД назначает какое-либо слово (не более 6 символов). Каждому кодированному слову ставится в соответствие определенный уровень полномочий (целое число от 1 до 15), характеризующий возможности данной категории пользователей по доступу к различным элементам базы данных. Чем выше уровень полномочий, тем большую возможность доступа к информации представляет знание соответствующего кодированного слова.

ЭЛЕМЕНТЫ

<имя первого элемента> <тип> (уровень по чтению, уровень по записи) ,

.....

<имя последнего элемента> <тип> (уровень по чтению, уровень по записи);

имя элемента в этом операторе не должно содержать более 6 символов. Тип элемента: Ц - целое число; Р - реальное (вещественное) число; *СNN* - символьный элемент, *NN* - его длина в машинных словах ( 2 символа в одном машинном слове). Уровни элемента по чтению и записи определяют категории тех пользователей, которые могут выполнять над этим элементом указанные действия. Например, строка ФАМ СЮ (5,10) определяет элемент с именем ФАМ, состоящий из 20 символов. Значение этого элемента могут читать пользователи с уровнем полномочий 5

и выше, модификацию или запись нового значения могут осуществлять пользователи с уровнем полномочий не ниже IO.

МАССИВ:

ввод данного оператора информирует систему о том, что далее следует описание структуры массива. Сначала вводят описание управляющих массивов, затем - описание рабочих массивов.

ИМЯ: <имя массива> , <тип> , <пароль> ;

имя массива не должно содержать более 6 символов; паролем является целое число от 0 до 32000; тип массива: У - управляющий, Р - рабочий.

ЗАПИСЬ:

<имя элемента> (связь),

.....  
.....

<имя элемента> (связь);

в данном операторе перечисляются те элементы, которые имеют связи с другими массивами. В круглых скобках для элемента управляющего массива задается количество связанных с этим элементом рабочих массивов, а для элемента рабочего массива - имя связанного с этим элементом управляющего массива.

РАЗМЕР: <число записей в массиве> ;

этот оператор вводится для определения размера области МД, в которой размещается данный массив.

КОНЕЦ

оператор предназначен для обозначения окончания процесса ввода схемы БД.

Подготовленную с помощью указанных операторов схему БД вводят в виде последовательного файла в ЭВМ и обрабатывают программой проектирования БД. В результате работы этой программы на МД создается специальный файл с описанием БД, который используется в других программах ШПМ БАНК-2.

### 2.3. Программа установки БД

Программа установки БД предназначена для заполнения, дополнения, копирования, загрузки, сортировки и реконфигурации БД.

Указанные функции реализуют 6 подпрограмм. Подпрограмма БАНК заполняет "пустую" базу данных подготовленной информацией. Информация в процессе записи заносится в заранее подготовленные файлы на МД. При необходимости добавления больших групп записей в уже существующую БД используется подпрограмма группового добавления записей. Подпрограмма реконфигурации базы данных создает на основе новой схемы новую БД, для заполнения которой используются данные,

хранящиеся в уже существующей БД. Подпрограмма сортировки используется для размещения информации по возрастанию значений ключевых элементов с целью минимизации времени поиска необходимых данных.

На случай возможного разрушения БД (например, в случае порчи МД) необходимо иметь ее копию. Подпрограмма копирования БД позволяет создать копию БД на магнитной ленте. Восстановление БД по этой копии (перепись БД с ленты на диск) производится подпрограммой загрузки БД. Эти же подпрограммы используются и для переноса БД на другую ЭМ.

#### 2.4. Программа аварийного запуска

Программа аварийного запуска предназначена для восстановления базы данных в случае сбоя в работе ЭВМ. Для этой же цели в начале работы с ней создают и затем постоянно ведут так называемые аварийный и архивный файлы. В а в а р и й н ы й файл записывают все вносимые за сеанс работы с БД изменения в данных. В а р х и в н ы й файл, расположенный, как правило, на магнитной ленте, записывают изменения в БД, внесенные за все сеансы работы с ней с момента снятия последней копии.

При машинном сбое либо могут быть потеряны результаты данного сеанса работы с БД, либо может произойти полное разрушение БД на магнитном диске. В первом случае с помощью программы аварийного запуска и записей аварийного файла можно восстановить результаты работы, предшествующие машинному сбою. Во втором случае эту цель можно достичь путем совместной обработки программой аварийного запуска копии БД и архивного файла.

#### 2.5. Программа управления БД

**2.5.1. Назначение.** Программа управления имеет имя ДИСПТ и выполняет в диалоговом режиме работы с ЭВМ следующие функции: поиск записей по характеристикам, определяемым пользователем; добавление новых записей; вычеркивание (уничтожение) записей; замена значений элементов в записях; выдача найденной информации на экран дисплея или на печать.

**2.5.2. Запуск ДИСПТ.** Программа допускает одновременную работу с несколькими базами данных (до 8-ми), поэтому перед запуском необходимо определить число инициализируемых баз. Кроме этого, при запуске программы определяют терминалы, с которых можно обратиться с запросом к базе, указывается адрес МД для размещения служебных файлов ДИСПТ и ряд других параметров. Запуск ДИСПТ производит преподаватель или системный программист. Форма диалога приведена в прил.1.

**2.5.3. Инициализация БД.** Открывает пользователю доступ к интересующей его БД. Формат команды инициализация:

БДН <имя БД> , <пароль БД> , <кодированное слово категории пользователей> , режим ; .

В графе режима указывается режим работы с БД, например, I (только чтение данных) или 3 (замена значений элементов).

2.5.4. Создание файла записей. Эта функция выполняется программой ДИСПТ при вводе команды:

ФЗП <имя файла> , <код защиты> , размер, КНЦ ; .

Здесь имя файла задается пользователем (не более 6 символов), им же назначается код защиты (любое целое положительное число). Размер файла устанавливается в секторах МД (целое число больше нуля) в зависимости от величины создаваемого файла. После выполнения этой команды на МД создается файл, в который заносят все записи, найденные в процессе работы с БД. В конце работы, после закрытия БД этот файл автоматически уничтожается программой ДИСПТ.

2.5.5. Поиск записей. Эта функция является основной в программе ДИСПТ. Формат соответствующей команды:

ПСК <процедура поиска> КНЦ ; .

Здесь процедура поиска - это часть команды, в которой в виде соотношений именами элементов и их значениями заданы условия поиска.

Ф о р м а т п р о ц е д у р ы п о и с к а :

<соотношение> <логическая операция> <соотношение>

Ф о р м а т с о т н о ш е н и я :

<имя элемента> <оператор поиска> "значение элемента"

О п е р а т о р ы п о и с к а :

РВ - равно                      НР - не равно

МН - меньше                    НМ - не меньше

БД - больше                    НБ - не больше

В процедуре поиска логические операции используют двух видов: логическое И; логическое ИЛИ. Логическое И означает, что должны быть удовлетворены оба условия, которые связывает эта логическая операция. Если одно из этих условий не выполняется для какого-либо элемента данных, то он не включается в файл найденных записей. При наличии в процедуре поиска логического ИЛИ в БД отыскивают все элементы, для которых выполняется хотя бы одно из условий, связанных данной логической операцией.

При составлении команд поиска необходимо знать имена элементов БД и учитывать, что в одном запросе допускается использовать только элементы, принадлежащие одному и тому же массиву.

Пример. Пусть в базе данных СТУДЕНТ (пп. 4.1) необходимо найти информацию о студентах I4I группы, имеющих рост выше 180 см или голубые

глаза. Команда на поиск этой информации будет иметь следующий вид:

```
ИСК  ГРУП  РВ "I4I" и РОСТ БД"180" ИЛИ  
ГРУП  РВ    "I4I" И ГЛАЗА РВ "ТОЛ" КНЦ; .
```

После выполнения команды поиска на экран дисплея выводится сообщение о количестве найденных записей.

2.5.6. В ы д а ч а н а й д е н н ы х з а п и с е й. После выполнения команды поиска найденные в БД записи переписываются в файл записей на МД. Необходимая пользователю информация из этих записей может быть выдана на экран дисплея или на печать с помощью следующей команды:

```
ВЫД  Д,  <имя элемента>, <имя элемента> , . . .
```

По этой команде печатаются значения указанных элементов первой из найденных записей. Для печати остальных достаточно последовательно передавать в ЭВМ команду ВЫД. После распечатки последней записи ДИСПИТ ответит только символом вопроса "?".

Формат команды выдачи может быть и иным:

```
ВЫД  <имя элемента>, <имя элемента>, . . .
```

В этом случае распечатываются подряд значения указанных элементов всех найденных записей.

Пример. Пусть в результате выполнения команды поиска (см. предыдущий пример) необходимо выдать на экран дисплея из найденных записей фамилии и рост студентов. Соответствующая команда будет иметь вид:

```
ВЫД  ФАМ,  РОСТ
```

2.5.7. З а м е н а з н а ч е н и й э л е м е н т о в. При реализации этой функции программы ДИСПИТ заменяются значения элементов в записях, найденных предварительно с помощью процедуры поиска. При этом следует помнить, что замена значений указанных элементов производится во всех записях, найденных после выполнения предшествующей команды поиска. Формат команды:

```
ЭЛН  <имя массива>;<имя элемента> = " значение " ;  
<имя элемента> = "значение ";..... ; КНЦ;
```

Для выполнения команды необходимо, чтобы режим работы БД был равен 3 (см. 2.5.3) и уровень полномочий пользователя был не меньше уровня защиты элемента по записи.

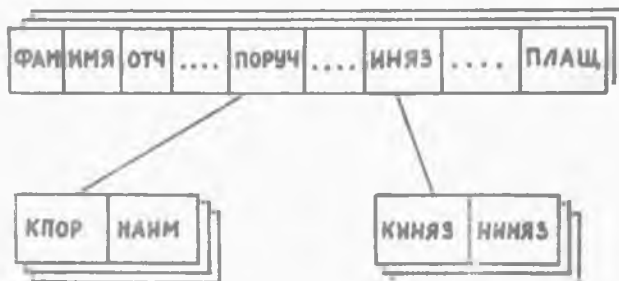
### 3. СТРУКТУРА БАЗЫ ДАННЫХ "СТУДЕНТ"

Лабораторное занятие предусматривает работу студентов с ПИМ БАНК-2 и базой данных СТУДЕНТ, которая создается следующим образом. Сначала в лабораторной работе К2 студенты всего потока готовят исходные данные на бланках, ИЛ и ПК. Затем в лабораторной работе К3 эти данные вводятся



с алфавитно-цифровых дисплеев с помощью редактирующей системы **SLIDE** и программы **НАСОС** в промежуточный файл на МД. Далее системный программист (администратор БД) переписывает созданные записи в БД **СТУДЕНТ**.

База данных **СТУДЕНТ** состоит из трех массивов: одного рабочего и двух управляющих (рис.4). Подробная расшифровка рабочего массива дана в таблице. Здесь управляющий массив для кодировки общественных поручений содержит следующие элементы: **КЦОР** – код общественного поручения, записываемый в виде целого числа; **НАИМ** – наименование поручения, записываемое в символьном виде (до 30 символов). Запись управляющего массива для кодировки иностранного языка также состоит из двух элементов: **КНИЯЗ** – код иностранного языка (целое число); **НИНЯЗ** – наименование языка, состоящее не более чем из 20 символов.



Р и с.4. Структура БД **СТУДЕНТ**

Для иллюстрации уровня полномочий при доступе к БД **СТУДЕНТ** выделено три кодовых слова: **СТУДТ**, **ПРЕП** и кодовое слово администратора БД. Кодовое слово **СТУДТ** позволяет читать всю информацию, кроме номера телефона и домашнего адреса; кодовое слово **ПРЕП** позволяет читать всю информацию; кодовое слово администратора БД позволяет не только читать, но и модифицировать всю информацию.

Т а б л и ц а

## Рабочий массив базы данных СТУДЕНТ

Имя элемента	Значение элемента	Тип данных	Максимальное количество символов
ФАМ	Фамилия студента	Символы	30
ИМЯ	Имя студента	Символы	20
ОТЧ	Отчество	Символы	30
ПОЛ	Пол(М или Ж)	Символы	1
ГОДР	Год рождения	Целое число	-
МРОЖД	Место рождения(город, село)	Символы	20
СЕМЬЯ	Семейное положение(ДА или НЕ)	Символы	2
КОЛДТ	Количество детей	Целое число	-

## Д а н н ы е о м е с т е ж и т е л ь с т в а

ГОР	Город	Символы	20
УЛ	Улица	Символы	30
ДОМ	Номер дома	Целое число	-
КВ	Номер квартиры(комнаты в общежитии)	Целое число	-
ТВЛ	Номер телефона	Символы	6

## Д а н н ы е о б и н с т и т у т е

ГОДП	Год поступления	Целое число	-
ГРУП	Номер группы	"	-
ПОРУЧ	Код общественного поручения	"	-
СТИП	Размер стипендии	"	-
СТОТР	Участие в стройотряде(ДА или НЕ)	Символы	2
ВЛКСМ	Год вступления в комсомол	Целое число	-
ИНЯЗ	Код иностранного языка	Целое число	-

Имя элемента	! Значение элемента !	! Тип данных !	! Максимальное количество символов !
О с о б н е п р и м е т ы			
РОСТ	Рост, в см	Целое число	-
ГЛАЗА	Цвет глаз (ГОЛ,КОР,ЧЕРН и т.д.)	Символы	4
ОБУВЬ	Размер обуви	Целое число	-
ШАПКА	Размер головного убора	Целое число	-
ПЛАЩ	Размер одежды	Целое число	-

#### 4. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

##### 4.1. Индивидуальное задание

1. Составить запросы к базе данных СТУДЕНТ на поиск и выдачу необходимой информации в соответствии с вопросами, приведенными в прил.2. Номера вопросов сообщаются каждому студенту на предыдущем занятии.

2. Выписать вопросы и составленные команды в отчет по лабораторной работе ( по форме прил.3).

3. На I-м этапе аудиторного занятия показать преподавателю подготовленные команды.

4. Во время диалога с программой ДИСПТ выполнить запланированные операции поиска и выдачи данных.

5. Дополнить отчет результатами работы на ЭВМ и сдать его преподавателю.

##### 4.2. Последовательность диалога с ЭВМ

1\* Запустить ДИСПТ. Форма диалога приведена в прил.1.

2. Инициализировать БД:

◀БДН СТУДТ , I2345, СТУДТ, I;▶

\* Выполняет системный программист

3. Назначить файл для запоминания найденных записей:

◀ФЗП имя файла, 6, 1, КНЦ ;▶

4. Осуществить поиск и выдачу информации, используя заранее выписанные тексты команд.

5. Закрывать БД командой ◀ОС;▶

### КОНТРОЛЬНЫЕ ВОПРОСЫ

1. В связи с какими проблемами появилась необходимость в создании БД?

2. Что представляет собой элемент данных?

3. Дайте определение записи.

4. Укажите назначение и разновидности ключей.

5. Что принято называть файлом?

6. Каковы отличия между логическим и физическим описаниями данных?

7. В чем суть понятий физической и логической независимости данных?

8. Дайте определение БД.

9. Можно ли на основании нескольких подсистем пользователей составить схему БД?

10. Каковы основные требования к СУБД?

11. Расскажите (по рис.1), как работает СУБД?

12. Какие структуры данных используются при разработке БД?

13. Чем отличается древовидная структура данных от сетевой?

14. Какие БД принято называть реляционными?

15. Что понимают под нормализацией структур данных?

16. Какие массивы используются в ШММ БАНК-2?

17. Какие программы входят в ШММ БАНК-2 и каково их назначение?

18. Напишите формат команды поиска информации программы ДИСПТ.

19. Напишите форматы команды выдачи информации.

### ЛИТЕРАТУРА

М а р т и н Дж. Организация баз данных в вычислительных системах. М.:Мир,1978.

О в ч а р о в Л.А., С е л е т к о в С.Н. Автоматизированные банки данных. М.:Финансы и статистика,1982.

ЗАГРУЗКА ПРОГРАММЫ ДИСПТ

- ◀: ЛН,7,пядн▶ , где пядн - порядковый номер дисплея
- ◀: СТ,ДИСПТ▶ ∅ ;
- ЧИСЛО БАЗ ДАННЫХ НА ОБСЛУЖИВАНИИ (запрос программы ДИСПТ)
- ◀1;▶
- ИМЯ СХЕМЫ БАЗЫ ДАННЫХ,ПАРОЛЬ,ПНД,АРХИВ
- ◀СТУД,12345,6,∅▶
- ПНД БАЗЫ ДАННЫХ
- ◀5;▶
- ЧИСЛО ОДНОВРЕМЕННО ОБСЛУЖИВАЕМЫХ АБОНЕНТОВ
- ◀3;▶
- ПНД РАСПОЛОЖЕНИЯ ТАБЛИЦ АБОНЕНТОВ
- ◀6;▶
- ЛН МЕЖЗАДАЧНОГО ОБМЕНА
- ◀∅;▶
- ПНД ФАЙЛА АВАРИИ
- ◀6;▶ ?
- ПНД ФАЙЛОВ ЗАПИСЕЙ
- ◀6;▶
- ◀ ПНД ФАЙЛОВ ПРОЦЕДУР
- ◀6;▶
- ПОДСИСТЕМА БАНК ГОТОВА К РАБОТЕ
- ◀БАНК▶
- ?

Символ "?" указывает на готовность к приему команд, описанных в пп.2.5.

ЗАДАНИЯ НА ПОИСК ИНФОРМАЦИИ В БД "СТУДЕНТ"

1. Сколько комсомольцев на потоке изучали английский или немецкий язык?
2. Какие иностранные языки изучали студенты вашей группы?
3. Сколько на вашем потоке старост ростом выше среднего (170 см)?
4. Сколько в вашей группе профсоюзных активистов?
5. Сколько в вашей группе комсомольских активистов?
6. У кого из студентов номер дома такой же, как и ваш?
7. Есть ли на потоке ваши тезки?
8. Сколько студенток знают немецкий язык и ведут активную общественную работу?
9. Сколько на потоке комсомольцев?
10. Все ли студенты имеют общественные поручения?
11. Определите цвета глаз ваших однокурсников.
12. Сколько студентов на потоке изучали французский язык?
13. Сколько студентов изучали немецкий?
14. Сколько студентов изучали английский язык?
15. Сколько студентов живет в общежитии №3?
16. Число студентов, у которых место рождения г.Куйбышев.
17. Сколько человек, кроме стройотрядовцев, никаких поручений не имеют?
18. Все ли студенты правильно задали свой рост в базе данных?
19. Все ли студенты правильно задали размеры обуви в базе данных?
20. Все ли студенты правильно задали размеры головного убора в базе данных?
21. У кого на потоке самый малый размер обуви?
22. Кто из студентов живет на вашей улице? (Свой адрес найти по базе данных).
23. У кого на потоке есть домашний телефон?
24. Сколько человек не получает стипендию и живет в общежитии?
25. Сколько человек получает стипендию?
26. Учитывая стипендию как оценку учебы студенты, определите, кто лучше учится: местные или иногородние студенты?
27. Сколько человек работало в стройотряде?
28. Сколько семейных студентов влезало в стройотряд?
29. Сколько детей у студентов вашего потока?

30. Кто самый молодой студент на потоке?
31. Есть ли на потоке студенты, ведущие активную общественную работу в ущерб учебе?
32. Найдите студентов вашей группы, знающих английский язык или не бывавших в стройотряде.
33. Кто из студентов потока живет в 6-м общежитии на 4-м этаже?
34. Есть ли семейные студенты, не получающие стипендии?
35. Кто на потоке раньше всех поступил в наш институт? Из группы?
36. Какие иностранные языки знают члены ДНД?
37. Кто из студентов ошибся при записи в БД следующих элементов: СЕМЬЯ, СТОТР, ПОЛ?
38. Все ли семейные студенты на потоке имеют общественные поручения?
39. Сколько человек с потока носит обувь и одежду вашего размера?
40. Сколько человек с потока выше вас? Ниже?
41. Чья голова на потоке самая большая? В группе?
42. Кто из треугольника вашей группы выше всех по росту?
43. В какой группе получают больше всех стипендий?
44. В какой группе больше всех комсомольских активистов?
45. Кто из студентов выезжал в стройотряды?
46. Проверьте, есть ли на потоке человек с вашим же ростом, цветом глаз, размером обуви, головного убора и одежды?
47. Чей стаж в ВЛКСМ больше всех?
48. Сколько в вашей группе голубоглазых? На потоке?
49. Сколько девушек вашего потока живет в общежитии?
50. Необходимо срочно составить баскетбольную команду для встречи с англичанами. Кто из потока подойдет для этой цели?
51. У кого из студентов вашей группы отца зовут так же, как и Вашего?

ОБРАЗЦЫ ОФОРМЛЕНИЯ ОТЧЕТА

Лабораторная работа №11

Выполнил студент 141 группы  
А.Н.ИВАНОВ

Задание. Определить количество студентов на потоке, изучающих английский язык.

Решение. 1. Определить код английского языка. Для этого прочитать все записи массива кодов иностранных языков:

⟨ПСК ИМЯ.Э БЛ Ø КНЦ ;⟩ - найти все коды.

⟨ВЫД П, КИМ.Э, ИМЯ.Э;⟩ - распечатать коды и определить код английского языка.

2. По найденному коду можно составить процедуру поиска:

⟨ПСК ИМЯ.Э РВ "код английского языка" КНЦ ;⟩

3. Выдать на печать найденные записи: фамилию и группу студента

⟨ВЫД ФАМ, ГРУП;⟩

Результаты работы (распечатка найденных данных).

ИВАНОВ 141  
ПЕТРОВ 149  
СИДОРОВ 144



## СОДЕРЖАНИЕ

	стр.
Порядок проведения работы.....	3
1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ РАБОТЫ.....	3
1.1. История появления баз данных.....	3
1.2. Основные понятия, термины и определения.....	4
1.3. Системы управления базами данных.....	5
1.1. Структуры данных.....	8
2. ПАКЕТ ПРОГРАММНЫХ МОДУЛЕЙ БАНК-2.....	10
2.1. Назначение и состав пакета.....	10
2.2. Программа проектирования БД.....	10
2.3. Программа установки БД.....	12
2.4. Программа аварийного запуска.....	13
2.5. Программа управления БД.....	13
3. СТРУКТУРА БАЗЫ ДАННЫХ "СТУДЕНТ".....	15
4. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ.....	18
4.1. Индивидуальное задание.....	18
4.2. Последовательность диалога с ЭВМ.....	18
Контрольные вопросы.....	19
ЛИТЕРАТУРА.....	19
Приложение 1. Загрузка программы ДИСПТ.....	20
Приложение 2. Задания на поиск информации в БД "СТУДЕНТ".....	21
Приложение 3. Образец оформления отчета.....	23

Составители: Ренат Наилевич И у р т а з и н,  
Александр Васильевич С о л о в о в

## ВВЕДЕНИЕ В БАЗЫ ДАННЫХ

Редактор в.д. А н т о н о в а  
Техн. редактор Н.М. К а л е н ю к

Подписано в печать 12.09.84.  
Формат 60x84 1/16. Бумага оберточная белая.

Оперативная печать.

Усл. п.л. 1,39. Уч.-изд.л. 1,3.

Тираж 500 экз. Заказ 6410 Бесплатно

Куйбышевский ордена Трудового Красного Знамени авиационный институт имени академика С.П.Королева,  
г. Куйбышев, ул. Молодогвардейская, 151.

Обл. типография им. В.П.Мяги, г. Куйбышев,  
ул. Венцека, 60.