

№	Доля в тексте	Доля в отчете	Источник	Актуален на	Модуль поиска	Комментарии
[01]	89,45%	0%	Разработка базы данных, отражающей учет успева... https://kazedu.kz	20 Ноя 2023	Интернет Плюс*	
[02]	87,94%	6,92%	Разработка базы данных, отражающей учет успева... http://myref.ru	15 Июн 2014	Интернет Плюс*	
[03]	87,84%	0%	8_117_18_0_0.600_89950334 Разработка базы данных... http://referatcollection.ru	07 Дек 2020	Интернет Плюс*	
[04]	87,84%	0%	Разработка базы данных, отражающей учет успева... http://ref.by	20 Ноя 2023	Интернет Плюс*	
[05]	87,83%	0%	С этого сайта http://p6.ru	20 Ноя 2023	Интернет Плюс*	
[06]	86,83%	36,21%	Разработка базы данных, отражающей учет успева... https://kazedu.kz	20 Ноя 2023	Интернет Плюс*	
[07]	86,05%	0%	181_99_39_0_0.1_8788518 http://p6.ru	05 Дек 2020	Интернет Плюс*	
[08]	64,55%	0%	Курсовая базы данных на pascal - реферат на сайте г... http://refwin.ru	19 Окт 2021	Интернет Плюс*	
						ПЕРЕФРАЗИРОВАННЫЕ

[09]	45,3%	28,72%	Разработка базы данных, отражающей учет успева... http://myref.ru	15 Июн 2014	заимствования по коллекции Интернет в русском сегменте
[10]	45,23%	11,71%	181_99_39_0_0.1_8788518 http://p6.ru	05 Дек 2020	Перефразированные заимствования по коллекции Интернет в русском сегменте
[11]	44,06%	0%	8_117_18_0_0.600_89950334 Разработка базы данных... http://referatcollection.ru	07 Дек 2020	Перефразированные заимствования по коллекции Интернет в русском сегменте
[12]	43,46%	2,95%	Курсовая: Разработка базы данных, отражающей у... http://referatbank.ru	03 Авг 2016	Перефразированные заимствования по коллекции Интернет в русском сегменте
[13]	43,28%	5,85%	Разработка базы данных, отражающей учет успева... http://referatcollection.ru	29 Янв 2017	Перефразирования по Интернету
[14]	43,28%	0%	не указано http://p6.ru	29 Янв 2017	Перефразирования по Интернету
[15]	42,32%	0%	Розробка бази даних, що відбиває облік успішності ... http://reff.net.ua	20 Ноя 2023	Интернет Плюс*
[16]	34,24%	0%	Реферат - Разработка базы данных, отражающей уч... http://ronl.ru	04 Авг 2016	Перефразированные заимствования по коллекции Интернет в русском сегменте
[17]	27,2%	0%	Курсовая базы данных на pascal - реферат на сайте r... http://refwin.ru	19 Окт 2021	Перефразированные заимствования по коллекции Интернет в русском сегменте
[18]	24,95%	0%	Курсовая базы данных на pascal - реферат на сайте r... http://refwin.ru	29 Янв 2017	Перефразирования по Интернету
[19]	19,24%	0%	Скачать База данных Успеваемость студентов. Разр... http://za4et.net.ru	05 Янв 2017	Перефразирования по Интернету
[20]	14,32%	0%	Разработка базы данных, отражающей учет успева... http://referat.star-info.ru	05 Янв 2017	Перефразирования по Интернету
[21]	10,51%	0%	Курсовая Базы данных на pascal - Реферат , страниц... http://works.doklad.ru	30 Янв 2017	Перефразирования по Интернету
[22]	6,81%	0%	СКАЧАТЬ РЕФЕРАТ Разработка базы данных, отража... http://zadachi.org.ru	05 Янв 2017	Перефразирования по Интернету
[23]	5,74%	0%	181_99_39_0_0.1_8788518 http://p6.ru	05 Дек 2020	Переводные заимствования по коллекции Интернет в русском сегменте
[24]	5,74%	0%	Разработка базы данных, отражающей учет успева... http://myref.ru	15 Июн 2014	Переводные заимствования по коллекции Интернет в русском сегменте
[25]	5,74%	0%	8_117_18_0_0.600_89950334 Разработка базы данных... http://referatcollection.ru	07 Дек 2020	Переводные заимствования по коллекции Интернет в русском сегменте
[26]	5,45%	0%	Создание базы данных аэропорта https://stud.wiki	21 Дек 2021	Интернет Плюс*
[27]	5,39%	0%	База данных 'Аэропорт'. Дипломная (ВКР). Информац... https://bibliofond.ru	15 Янв 2021	Интернет Плюс*
[28]	5,39%	0%	База данных "Аэропорт" https://knowledge.allbest.ru	09 Янв 2020	Интернет Плюс*
[29]	5,39%	0%	Курсовая работа: Создание базы данных аэропорта ... https://bestreferat.ru	10 Июн 2021	Интернет Плюс*
[30]	4,92%	0%	Курсовая "Базы данных" на pascal - курсовая работа ... http://dipland.ru	30 Янв 2017	Перефразирования по Интернету
[31]	4,72%	4,72%	Реферат: База данных в СУБД ACCESS - BestReferat.ru https://bestreferat.ru	08 Дек 2020	Интернет Плюс*
[32]	4,41%	0%	Курсовая работа: Создание базы данных аэропорта ... https://bestreferat.ru	06 Июн 2019	Интернет Плюс*
[33]	4,33%	0%	СКАЧАТЬ РЕФЕРАТ Сортировка Программирование,... http://zadachi.org.ru	30 Янв 2017	Перефразирования по Интернету
[34]	4,27%	0%	Курсовая базы данных на pascal - реферат на сайте r... http://refwin.ru	19 Окт 2021	Переводные заимствования по коллекции Интернет в русском сегменте
[35]	3,9%	0%	42910_744_00bc37df.docx	02 Ноя 2022	Кольцо вузов
[36]	3,9%	0%	Разработка алгоритма определения справочных су... http://elibrary.ru	06 Мая 2020	eLIBRARY.RU
[37]	3,9%	0%	О. В. Власова ; М-во образования и науки Российско... http://dlib.rsl.ru	01 Дек 2014	Сводная коллекция РГБ
[38]	3,87%	0%	Еркынбек-Диана ИП-18-Зр	07 Мая 2021	Кольцо вузов
[39]	3,87%	0%	Ахылбеков Ұ.А. ИП-18-Зтр Оперативный учет матер...	12 Дек 2021	Кольцо вузов
[40]	3,87%	0%	Пожидаев, Алексей Викторович диссертация ... канд... http://dlib.rsl.ru	07 Мар 2012	Сводная коллекция РГБ
[41]	3,86%	0%	Андриевский, Александр Леонидович На примере А... http://dlib.rsl.ru	раньше 2011	Сводная коллекция РГБ
[42]	3,86%	0,17%	СКАЧАТЬ РЕФЕРАТ Успеваемость студентов Програ... http://zadachi.org.ru	12 Апр 2022	Интернет Плюс*

[43]	3,82%	0%	Дубровский, Алексей Викторович диссертация ... ка... http://dlib.rsl.ru	20 Янв 2010	Сводная коллекция РГБ	
[44]	3,61%	0%	Прохоров, Виктор Николаевич диссертация ... докто... http://dlib.rsl.ru	раньше 2011	Сводная коллекция РГБ	
[45]	3,29%	0%	РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ "СИС... http://elibrary.ru	24 Янв 2020	Перефразирования по eLIBRARY.RU	
[46]	2,32%	0%	Курсовая: Разработка базы данных, отражающей у... http://referatbank.ru	03 Авг 2016	Переводные заимствования по коллекции Интернет в русском сегменте	
[47]	2,3%	0%	Разработка базы данных, отражающей учет успева... http://referats.name	09 Апр 2016	Перефразированные заимствования по коллекции Интернет в русском сегменте	
[48]	2,24%	0%	Разработка программного прототипа медицинско...	09 Мая 2020	Кольцо вузов	
[49]	1,87%	0%	Разработка базы данных, отражающей учет успева... http://referats.name	09 Апр 2016	Перефразированные заимствования по коллекции Интернет в русском сегменте	
[50]	1,8%	0%	Введение в реляционные базы данных http://ibooks.ru	09 Дек 2016	Сводная коллекция ЭБС	
[51]	1,58%	0%	Контрольная: "Построение арифметико-логическ... https://referat.bookap.info	20 Ноя 2023	Интернет Плюс*	
[52]	1,51%	0%	ОСОБЕННОСТИ ЯЗЫКА SQL И ЕГО ИСПОЛЬЗОВАНИ... http://elibrary.ru	01 Янв 2023	eLIBRARY.RU	
[53]	1,5%	0%	РАЗРАБОТКА БАЗЫ ДАННЫХ ПО СОРТАМ РАСТЕНИЯ... http://elibrary.ru	01 Янв 2020	eLIBRARY.RU	
[54]	1,5%	0%	РЕЕСТР СОРТОВ РАСТЕНИЯ ВИДА «МОРКОВЬ». http://elibrary.ru	01 Янв 2020	eLIBRARY.RU	
[55]	1,5%	0%	РАЗРАБОТКА БАЗЫ ДАННЫХ ПО СОРТАМ РАСТЕНИЯ... http://elibrary.ru	01 Янв 2020	eLIBRARY.RU	
[56]	1,47%	0%	База данных "Успеваемость" - Информатика, прогр... https://kazedu.kz	20 Ноя 2023	Интернет Плюс*	
[57]	1,37%	0%	62851 http://e.lanbook.com	09 Мар 2016	Сводная коллекция ЭБС	
[58]	1,04%	1,04%	не указано	13 Янв 2022	Библиография	
[59]	1,03%	0%	2022_СКФ_09.03.01_БР_Хачатурян_Сергей_Арташесо...	08 Июн 2022	Кольцо вузов	
[60]	1,02%	0%	Базы данных http://ibooks.ru	09 Дек 2016	Сводная коллекция ЭБС	
[61]	1,02%	0%	Базы данных. Учебное пособие http://bibliorossica.com	26 Мая 2016	Сводная коллекция ЭБС	
[62]	0,99%	0%	Реферат.docx	18 Ноя 2022	Кольцо вузов	
[63]	0,93%	0,12%	Курсовая работа_БД_фармацевтика.docx	17 Дек 2021	tuiv-vuz	
[64]	0,92%	0%	Выборка, или чтение данных - Максимов Е.М. Струк... http://nashaucheba.ru	01 Янв 2017	Перефразирования по Интернету	
[65]	0,87%	0%	Проектирование и разработка базы данных страхо... https://revolution.allbest.ru	01 Дек 2020	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[66]	0,85%	0%	214532 http://biblioclub.ru	18 Апр 2016	Сводная коллекция ЭБС	Источник исключен. Причина: Маленький процент пересечения.
[67]	0,8%	0%	Реферат - Разработка базы данных, отражающей уч... http://ronl.ru	04 Авг 2016	Переводные заимствования по коллекции Интернет в русском сегменте	Источник исключен. Причина: Маленький процент пересечения.
[68]	0,74%	0%	РАЗРАБОТКА КЛИЕНТ-СЕРВЕРНОГО ПРИЛОЖЕНИЯ «... http://elibrary.ru	01 Янв 2020	eLIBRARY.RU	Источник исключен. Причина: Маленький процент пересечения.
[69]	0,54%	0%	Курсовая БД — исправ.docx	17 Янв 2023	Кольцо вузов	Источник исключен. Причина: Маленький процент пересечения.
[70]	0,54%	0%	Методика решения учебных задач средствами прог... http://studentlibrary.ru	20 Дек 2016	Медицина	Источник исключен. Причина: Маленький процент пересечения.
[71]	0,45%	0%	Базы данных и СУБД https://otherreferats.allbest.ru	19 Сен 2019	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[72]	0,35%	0%	Алгоритмы обработки строк http://studentlibrary.ru	20 Дек 2016	Медицина	Источник исключен. Причина: Маленький процент пересечения.
[73]	0,24%	0%	Некоторые вопросы вычислительной сложности и ... http://dep.nlb.by	06 Дек 2018	Диссертации НББ	Источник исключен. Причина: Маленький процент пересечения.
[74]	0,24%	0%	Оценка и учет частотных характеристик трансформ... http://dep.nlb.by	11 Ноя 2016	Диссертации НББ	Источник исключен. Причина: Маленький процент пересечения.
[75]	0,22%	0%	не указано	13 Янв 2022	Шаблонные фразы	Источник исключен. Причина: Маленький процент пересечения.



МОСКОВСКИЙ УНИВЕРСИТЕТ ИМ. С.Ю.ВИТТЕ

Факультет Информационных технологий

Кафедра Кафедра информационных систем

Направление подготовки/Специальность Прикладная информатика

ОТЧЕТ

о прохождении

Производственная практика

/вид практики/

Технологическая (проектно-технологическая) практика

/тип практики/

Студента Куркиной Натальи Владимировны
(фамилия, имя, отчество)

Место прохождения практики ЧОУВО "МУ им. С.Ю. Витте"

Период прохождения практики с 06.02.2023 г. по 11.06.2023 г.

г. Москва 2023 г.

Содержание

Введение.....	3
1. Постановка задачи.....	4
2. Описание базы данных.....	4
3. Первоначальные данные.....	6
4. Интерфейс программы.....	8
5. Описание используемых процедур.....	10
6. Возможности полученного программного продукта.....	15
Список использованной литературы.....	28

Введение

Основные идеи современной информационной технологии базируются на концепции, согласно которой данные должны быть организованы в базы данных с целью адекватного отображения изменяющегося реального мира и удовлетворения информационных потребностей пользователей. Эти базы данных создаются и функционируют под управлением специальных программных комплексов, называемых системами управления базами данных (СУБД).

Увеличение объема и структурной сложности хранимых данных, расширение круга пользователей информационных систем привели к широкому распространению наиболее удобных и сравнительно простых для понимания реляционных (табличных) СУБД. Для обеспечения одновременного доступа к данным множества пользователей, нередко расположенных достаточно далеко друг от друга и от места хранения баз данных, созданы сетевые мультипользовательские версии БД основанных на реляционной структуре. В них тем или иным путем решаются специфические проблемы параллельных процессов, целостности (правильности) и безопасности данных, а также санкционирования доступа.

Целью этой работы является создать базу данных в СУБД ACCESS. Которая должна будет автоматизировать работу ВУЗа.

1. Постановка задачи.

Построить систему управления базой данных (в дальнейшем-СУБД), отражающую учет успеваемости студентов вуза. При этом надо предусмотреть, что данные могут вноситься разными людьми. Построить базу данных, которая была бы максимально гибкой. В базе данных должны содержаться данные о студентах, оценках, преподавателях и предметах.

2. Описание базы данных.

База данных нормализована и состоит из четырех таблиц. Она разбита на четыре таблицы, дабы быть максимально гибкой. При этом разные таблицы могут заполняться разными людьми, а потом объединяться в одну общую базу данных. Например, таблицу о студентах заполняют на кафедре, на которой учиться студент, таблицу о преподавателе и предметах – на кафедре этих преподавателей, а оценки ставит преподаватель.

12

Таблица 1 - Структура таблиц

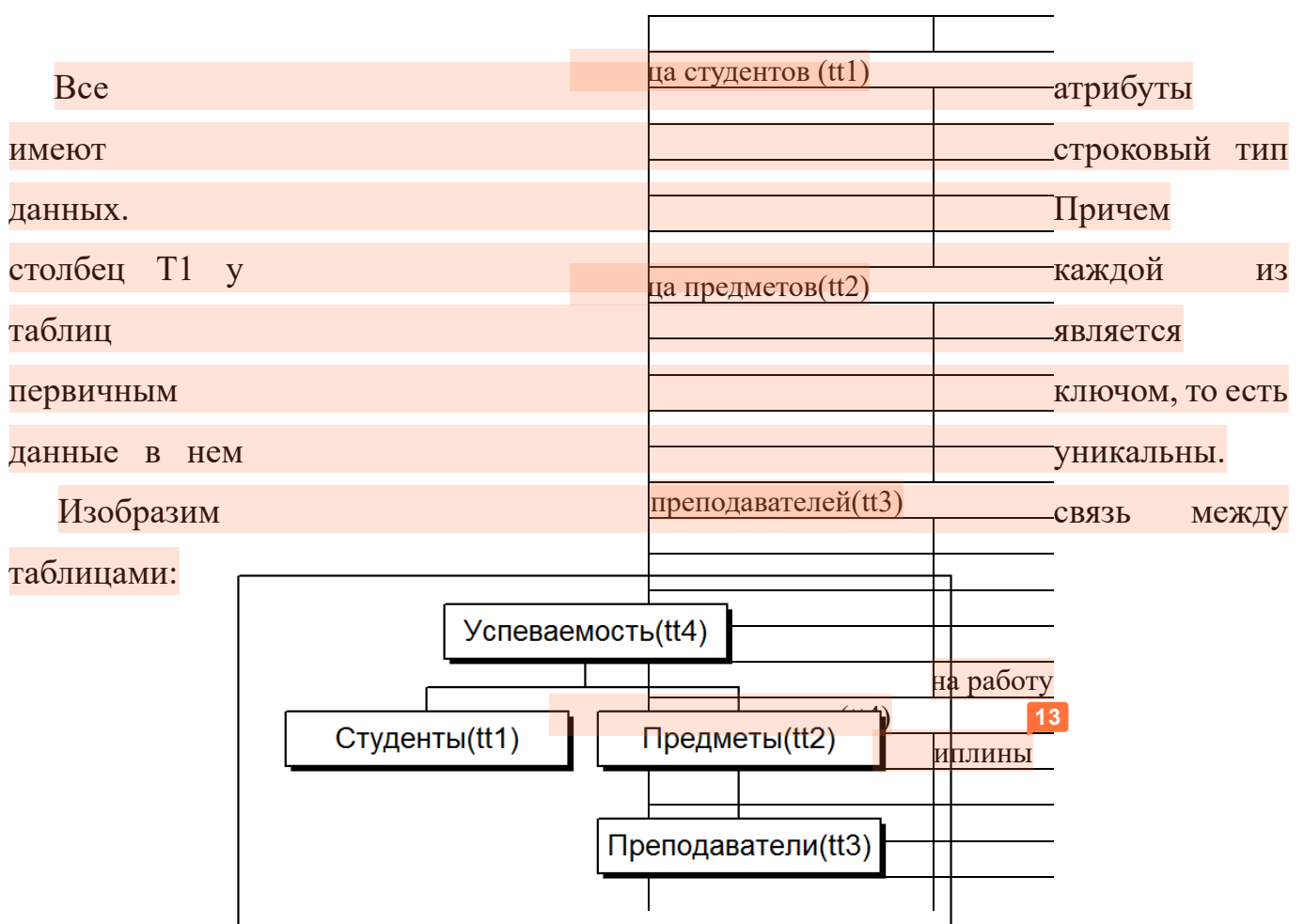


Рисунок 1 – Связь между таблицами

Связь эта осуществляется через уникальные поля каждой из таблиц. Из схемы видно, что связь между преподавателями и, например, студентами, осуществляется только через таблицу предметов. Такая организация базы данных позволяет избежать дублирования данных. Например, если бы у нас таблица предметов была объединена с таблицей преподавателей, то мог выползти такой случай – один преподаватель ведет два предмета, следовательно его фамилия, имя и отчество будут повторятся в таблице дважды. А это уже называется избыточностью данных.

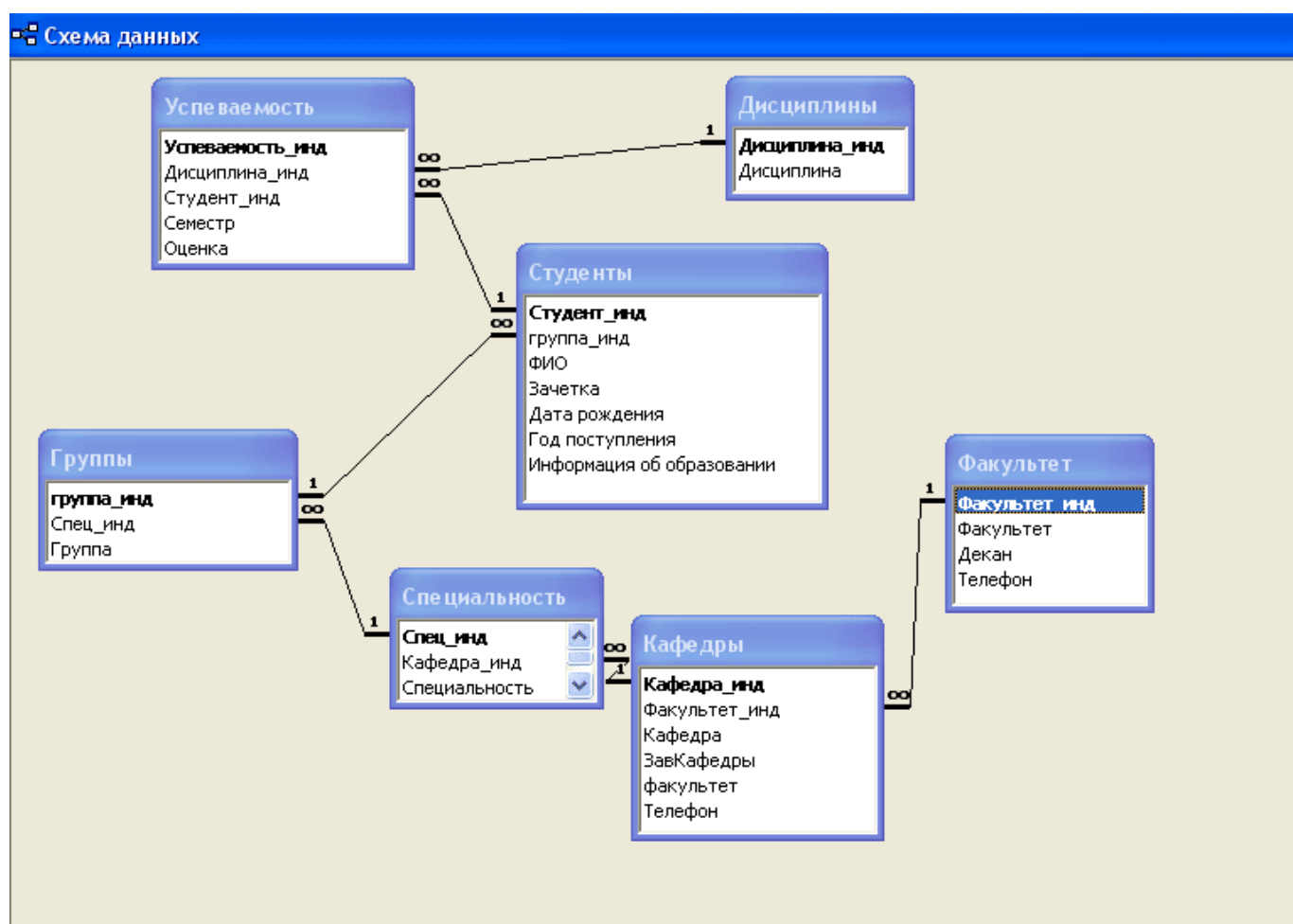


Рисунок 2 – Реляционная модель

3. Первоначальные данные

Для примера введем некоторые данные. Вот эти данные, выведенные с помощью команды – «Сохранить в текст. файл»:

Таблица 2 - Студенты

Студенческий код	Фамилия	Имя	Отчество	Стипендия
3412	Поляков	Анатолий	Алексеевич	54.50
3413	Старова	Любовь	Михайловна	17.00
3414	Гриценко	Владимир	Николаевич	0.00
3415	Котенко	Анатолий	Николаевич	0.00
3416	Нагорный	Евгений	Васильевич	25.50

13

Таблица 3 - Предметы

Код предмета	Название	Код преподавателя	Время учебы	Курс
2001	Физика	4001	34	1
2002	Химия	4002	68	1
2003	Математика	4003	68	1
2004	Философия	4005	17	2
2005	Экономика	4004	17	3

Таблица 4 – Преподаватели

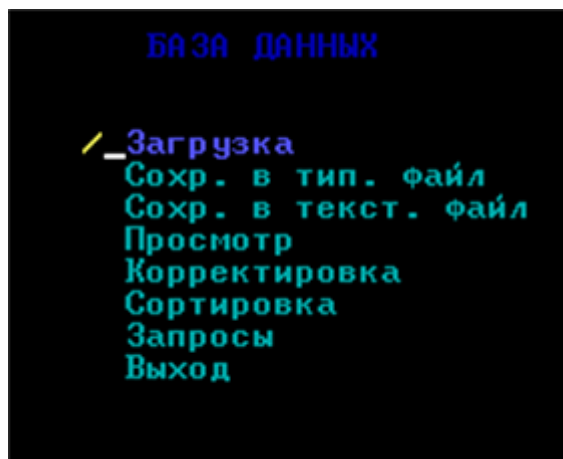
Код преподавателя	Фамилия	Имя	Отечество	Начало работы
4001	Викулина	Валентина	Ивановна	01.04.1984
4002	Костыркин	Олег	Владимирович	01.09.1997
4003	Казанко	Виталий	Владимирович	01.09.1988
4004	Позднякова	Любовь	Алексеевна	01.09.1998
4005	Загарийчук	Игорь	Дмитриевич	10.05.1989

Таблица 5 – Оценки

Код сдачи	Оценка	Дата сдачи	Студенческий	Код предмета
1001	5	10.06.1999	3412	2001
1002	4	10.06.1999	3413	2003
1003	3	11.06.1999	3414	2005
1004	4	12.06.1999	3412	2003
1005	5	12.06.1999	3416	2004

4. Интерфейс программы

Для того, чтоб понять, что делает программа рассмотрим ее меню:



9

Рис. 3

На рисунке 3 представлено главное меню программы, которое появляется при ее запуске. Тут:

«**Загрузка**» - ввод базы данных из типизированного файла.

«**Сохранение в типичный файл**» - сохранение базы данных в типизированном файле.

«**Сохранение в текстовый файл**» - сохранение выбранной таблицы в текстовый файл.

«**Просмотр**» - просмотр данных таблиц.

«**Корректировка**» - изменение записей таблиц.

«**Сортировка**» - сортировка записей по определенному полю.

«**Запросы**» - поиск данных в четырех таблицах, соответствующие заданному условию.

«**Выход**» - выход из программы.

Главное меню содержит вложенные. Например на 4 рисунке показано меню, служащее для выбора таблицы (одной из четырех). Вход в него может быть осуществлен по разному. То есть это меню вы будете проходить при просмотре данных, при корректировке и при сохранении в текстовый файл. Это меню в комментариях не нуждается.

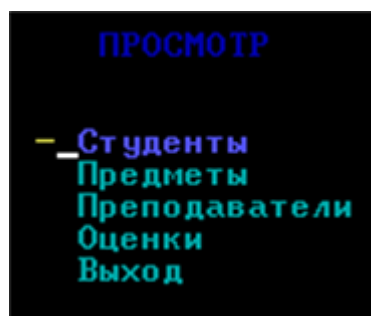


Рис. 4

При выборе «Корректировка» вы перейдете к меню, показанному на рисунке 5.

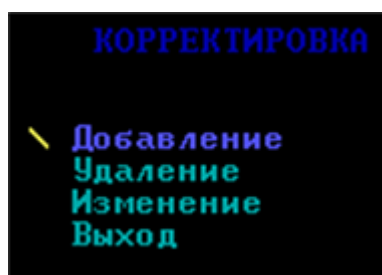


Рис. 5

«Добавление» - добавление нового кортежа в заданную с помощью предыдущего меню таблицу.

«Удаление» - удаление строки из таблицы, причем запись в таблице ищется по первому столбцу.

«Изменение» - изменение свойств существующей в базе данных строки.

Последнее подменю – это подменю запросов, на котором надо остановиться поподробнее. Рассмотрим рисунок:

«Найти оценку» - найти оценку студента по его фамилии и предмету, который он

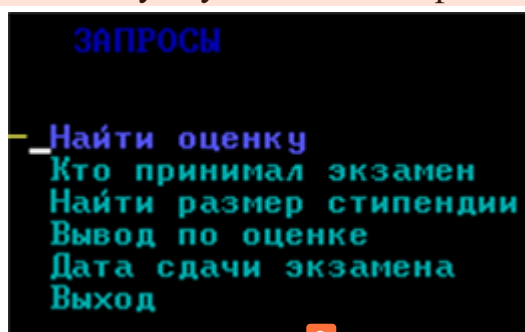


Рис. 6

сдавал.

«Кто принимал экзамен» - фамилия преподавателя, принимавшего данный предмет.

«Найти размер стипендии» - размер стипендии выбранного студента.

«Вывод по оценке» - показать всех студентов, получивших данную оценку.

«Дата сдачи экзамена» - найти даты, когда принимался данный предмет.

В запросах используются реальные названия предметов, фамилии студентов, а не их номера (для студентов например номер соответствует номеру студенческого билета).

5. Описание используемых процедур.

В приложении используются как глобальные, так и локальные переменные. К глобальным переменным относятся:

1) `names, namer:string[10]` – строковые переменные, которые вынесены в глобальные, чтобы сделать программу более компактной. Иначе бы в разных процедурах надо было вводить аналогичные типы данных, а так можно обойтись лишь парой глобальных переменных.

2) `key, kr:char` – символьные переменные. В переменной `key` хранится код нажатой клавиши. А в переменной `kr` находится текущее положение звездочки. Звездочка будет рассмотрена ниже.

3) `tek, i, j, izm:integer` – целочисленные переменные. Из них `i` и `j` – циклические, временные переменные. `Tek` и `izm` показывают, какое меню активно в текущий момент. Они помогают понять программе от какого меню к какому перешли, то есть помогает восстановить последовательность действий.

4) `exist, vfile, issor:boolean` – Логические переменные. `Exist` – служит для выхода из программы. `Vfile` – позволяет пускать одну процедуру по двум путям – либо выводить данные на экран, либо в файл. `Issor` – показывает меню выбора таблиц, что следующим шагом должна быть сортировка данной таблицы.

5) `nast:pered` – Типизированная переменная, в которой хранятся строки текущего меню. Описание типа `pered` имеется в модуле:

```
pered=record
```

```
st:array[1..12] of string;
```

```
m:byte; {количество строк в меню}
```

```
end;
```

`temr, tt1, tt2, tt3, tt4:cc` – Таблицы базы данных. Тут `tt1` – таблица с данными о студентах, `tt2` – предметы, `tt3` – преподаватели, `tt4` – оценки (успеваемость). `Temr` –

временная таблица. Все эти переменные являются динамическими списками. Они описаны в файле tips.pas:

```
tabl2=record {Сама таблица}
  t1,t2,t3,t4,t5:string[12]; {Состоят из строк}
end;
cc=^tab;
tab=record {Создание динамического списка}
  tabl:tabl2;
  sled:cc;
end;
```

Программа отвечает на то, какую клавишу нажал пользователь и передает управление процедуре main. Процедура readkey используется в основном блоке программы. Для работы с управляющими клавишами используются их коды в таблице ASCII. Тут:

80 – курсор вверх

72 – курсор вниз

27 – ESC

13 – Enter

При нажатии на клавишу «Ввод» управление передается процедуре main. При этом используются глобальные переменные.

В процедуре main управление переходит к подпрограмме, согласно текущему меню, но порой надо знать откуда был осуществлен вход. Как я уже говорил, зайти в меню выбора таблиц можно по разному. Все меню имеют разные заголовки, которые и используются для их идентификации. При нажатии на кнопку «Выход», текущее становится первоначальное меню.

Интерфейсная часть программы состоит из трех процедур:

```
procedure menus(m:pered;max:byte); {Вывод меню}
begin
  clrscr;
  For i:=1 to max do begin
    if i=1 then begin
```

```

    textcolor(1); gotoxy(9,2); write(m.st[i]);
end
else begin
    if i=2 then textcolor(9)
    else textcolor(3);
    gotoxy(8,i+3);write(m.st[i]);
end;
end;
end;
end;

```

С помощью этой процедуры на экране меняются меню. Нужно учесть, что мы первоначально знаем, из скольки строк состоит каждое меню, что упрощает задачу. Заголовков и содержание меню пишутся разными цветами. Изменение цвета выводимых символов идет с помощью команды `textcolor`, параметром которой является предопределенная константа.

Следующая процедура:

```

procedure krutis; {Звездочка рядом с активным элементом}
begin
    textcolor(14);
    if kr="/" then kr:="-"
    else if kr="-" then kr:="\
    else if kr="\ then kr:="|
    else kr:="/";
    gotoxy(6,tek+3);write(kr);
    textcolor(3);
end;

```

Эта процедура работает таким образом – обращение к ней идет из основного блока программы, в котором стоит бесконечный цикл. На экране отображается что то похожее на бенгальский огонь. Это достигается путем последовательного вывода на одном и том же месте символов: \, |, /, —. В переменной `tek` у нас содержится номер активной строки, а относительно нее мы и должны поставить звездочку.

Следующая процедура:

```

procedure ramka(ch:char); {перемещение указателя}
begin
  gotoxy(6,tek+3);Writeln(' ');
  textcolor(3);gotoxy(8,tek+3);write(nast.st[tek]);
  if ch='+' then tek:=tek+1
  else tek:=tek-1;
  if tek=1 then tek:=nast.m
  else if tek=nast.m+1 then tek:=2;
  key:=#0;
  textcolor(9);gotoxy(8,tek+3);write(nast.st[tek]);
  krutis;
end;

```

Данная процедура имеет параметр символьного типа. По содержанию *ch* определяется – вверх или вниз сдвинулась активная строка. Обращение к этой процедуре идет из основного блока программы. Больше процедур такого рода нет (ко всем другим процедурам обращение идет из процедуры *main*).

Эта процедура затирает элемент, который был активен перед этим цветом по умолчанию (голубым) и стирает предыдущее положение звездочки. После этого исходя из значения *ch* изменяется значение *tek* и выделяется элемент, на который был осуществлен переход. В подпрограмме вставлены два условия:

```

if tek=1 then tek:=nast.m
else if tek=nast.m+1 then tek:=2;

```

Они срабатывают когда активным является верхний элемент меню и пользователь нажал кнопку вверх, тогда текущей становится последняя строка, и наоборот – текущая строка является последней, а пользователь нажал кнопку вниз, тогда активной станет первая строка активного меню.

Процедуры следующего типа осуществляют загрузку и выгрузку данных из файла. Чтение осуществляется с помощью *readetip*, а запись – *writetip*. В этих процедурах используются глобальные переменные. Загрузка идет посредством таблиц *tt1*, *tt2*, *tt3*, *tt4*. При работе пользователь работает как бы с одним файлом, хотя реально каждая из таблиц сохраняется в свой файл. При запросе пользователь вводит название файла, а потом к этому названию добавляется слева цифра – 1, 2, 3 или 4.

Например при удалении записей из таблицы, пользователю будет удобнее, если он будет содержание таблицы перед глазами. Например покажем содержание таблицы студентов, которые будут выведены после нажатии на кнопку «Просмотр»:

Следующая процедура - `obrabotka(iz,t:integer; var rab:cc)`. Эта процедура вызывается при корректировке записей. Через параметр `iz` процедура выбирает путь дальнейшего хода. То есть надо ли добавить запись, изменить или удалить. Параметр `rab` – это таблица, которая передается в процедуру, и параметр `var` показывает, что данную таблицу можно изменять непосредственно из процедуры.

В ней используются локальные переменные –

`dlud:string;`

`bis:boolean;`

`tems,temr,tem:cc;`

Здесь все переменные типа `cc` – временные, `dlud` служит для ввода данных, а `bis` показывает, выполнимо ли выбранное действие или нет.

Процедура `sort(iz,t:integer; var rab:cc)` осуществляет сортировку записей в выбранной таблице по выбранному полю. Ее работа и параметры с переменными аналогичны предыдущей процедуре.

При запросах выполняется процедура `zaproz(num:integer)`, где через параметр `num` передается, какой именно запрос должен выполняться. Потом через условие `case` идет обработка запросов. Как показано на схеме взаимодействия таблиц, чтобы перейти от студентам к преподавателям надо пройти через таблицу оценок. Связь между таблицами осуществляется по уникальным полям. Например для нахождения оценки студента надо сначала из таблицы студентов найти номер его студенческого, а потом найти номер сдаваемого предмета из таблицы, а уже после этого, используя полученные номера, найти из таблицы успеваемости полученную студентом оценку. То есть связь идет по трем таблицам.

6. Возможности полученного программного продукта

Таким образом, мы построили гибкую модель базы данных, в которой легко создать нужный запрос, данные представлены в удобном для пользователя виде. Интерфейс программы построен без излишков и настроен на максимальное удобство пользователя. Программа позволяет заполнять базу данных одновременно несколькими пользователями, каждый из которых будет заполнять свою таблицу. Так как программа работает с динамическими списками, то она быстра и позволяет избежать избыточности данных в таблицах.

10

```

program Delphins;
uses crt,tips;
var names,namer:string[10];
    key,kr:char;
    tek,i,j,izm:integer;
    exist,vfile,issor:boolean;
    nast:pered;
    temr,tt1,tt2,tt3,tt4:cc;
    outf:file of tabl2;

procedure menus(m:pered;max:byte); {Вывод меню}
begin
    clrscr;
    For i:=1 to max do begin
        if i=1 then begin
            textcolor(1); gotoxy(9,2); write(m.st[i]);
        end
        else begin
            if i=2 then textcolor(9)
            else textcolor(3);
            gotoxy(8,i+3);write(m.st[i]);
        end;
    end;
end;

procedure krutis; {Звездочка рядом с активным элементом}
begin
    textcolor(14);
    if kr='/' then kr:='- '
    else if kr='- ' then kr:='\ '
    else if kr='\ ' then kr:='| '
    else kr:='/';
    gotoxy(6,tek+3);write(kr);
    textcolor(3);
end;

procedure ramka(ch:char); {перемещение указателя}
begin
    gotoxy(6,tek+3);Writeln(' ');
    textcolor(3);gotoxy(8,tek+3);write(nast.st[tek]);
    if ch='+' then tek:=tek+1
    else tek:=tek-1;
    if tek=1 then tek:=nast.m
    else if tek=nast.m+1 then tek:=2;
    key:=#0;
    textcolor(9);gotoxy(8,tek+3);write(nast.st[tek]);
    krutis;
end;

procedure tab 6 (t:integer;rab:cc); {Вывод таблицы в файл}
var ooutf:text;

```

```

tem:cc;
begin
clrscr;
writeln('Введите имя файла');
readln(names);
assign(ooutf,names);
rewrite(ooutf);
writeln(ooutf,menu2.st[t]);
writeln(ooutf,'+-----+');
writeln(ooutf,'|,mm[t-1,1]:14,|,mm[t-1,2]:14,|,mm[t-1,3]:14,|,mm[t-1,4]:14,|,mm[t-1,5]:14,|');
writeln(ooutf,'+-----+-----+-----+-----+');
tem:=rab;
while tem<>nil do
begin
writeln(ooutf,'|,tem^.tabl.t1:14,|,tem^.tabl.t2:14,|,tem^.tabl.t3:14,|,tem^.tabl.t4:14,|,tem^.tabl.t5:14,|');
tem:=tem^.sled;
end;
writeln(ooutf,'+-----+');
close(ooutf);
nast:=menu1;
menus(nast,nast.m);
tek:=2;
end;

```

```

procedure tabl1(t:integer;rab:cc;y:boolean); {Вывод таблицы на экран}
var tem:cc;
begin
clrscr;
writeln(menu2.st[t]);
writeln('+-----+');
writeln('|,mm[t-1,1]:14,|,mm[t-1,2]:14,|,mm[t-1,3]:14,|,mm[t-1,4]:14,|,mm[t-1,5]:14,|');
writeln('+-----+-----+-----+-----+');
tem:=rab;
while tem<>nil do
begin
writeln('|,tem^.tabl.t1:14,|,tem^.tabl.t2:14,|,tem^.tabl.t3:14,|,tem^.tabl.t4:14,|,tem^.tabl.t5:14,|');
tem:=tem^.sled;
end;
writeln('+-----+');
if not y then begin
readln;
nast:=menu1;
menus(nast,nast.m);
tek:=2;
end;
y:=false;
end;

```

```

procedure sort(iz,t:integer; var rab:cc); {Сортировка по полю}
var po:integer;
te1,te2,tem:cc;

```

```

str1,str2:string;
ttrtt:tabl2;
begin
tabl1(tek,rab,true);
writeln('Введите номер столбца по которому надо отсортировать данные');
readln(po);
te1:=rab;
while te1<>nil do begin
te2:=te1^.sled;
while te2<>nil do begin
case po of
1:begin str1:=te1^.tabl.t1; str2:=te2^.tabl.t1; end;
2:begin str1:=te1^.tabl.t2; str2:=te2^.tabl.t2; end;
3:begin str1:=te1^.tabl.t3; str2:=te2^.tabl.t3; end;
4:begin str1:=te1^.tabl.t4; str2:=te2^.tabl.t4; end;
5:begin str1:=te1^.tabl.t5; str2:=te2^.tabl.t5; end;
end;
if str1>str2 then begin
ttrtt:=te1^.tabl;
te1^.tabl:=te2^.tabl;
te2^.tabl:=ttrtt;
end;
te2:=te2^.sled;
end;
te1:=te1^.sled;
end;
tabl1(tek,rab,false);
end;

```

```

procedure obrabotka(iz,t:integer; var rab:cc); {Обработка записей}
var dlud:string;
bis:boolean;
tems,temr,tem:cc;
begin
clrscr;
if iz=1 then begin {добавление записи}
if rab<>nil then begin
tem:=rab;
while tem^.sled<>nil do tem:=tem^.sled;
new(tem^.sled);
tem:=tem^.sled;
end
else begin
new(rab);
tem:=rab;
end;
writeln(mm[t,1]);readln(tem^.tabl.t1);
writeln(mm[t,2]);readln(tem^.tabl.t2);
writeln(mm[t,3]);readln(tem^.tabl.t3);
writeln(mm[t,4]);readln(tem^.tabl.t4);
writeln(mm[t,5]);readln(tem^.tabl.t5);
tem^.sled:=nil;
tem:=rab;
izm:=0;

```

```

nast:=menu1;
menus(nast,nast.m);
tek:=2; iz:=0;
end
else if iz=2 then begin    {Удаление записи}
tems:=rab;
tabl1(tek,rab,true);
writeln('Введите уникальный номер'); readln(dlud);
bis:=true;
if rab^.tabl.t1 = dlud then begin
    rab:=rab^.sled;
    bis:=false;
end
else begin
    while tems<>nil do begin
        if tems^.sled^.tabl.t1=dlud then begin
            tem:=tems^.sled;
            tems^.sled:=tems^.sled^.sled;
            dispose(tem);
            bis:=false;
            break;
        end;
        tems:=tems^.sled;
    end;
    end;
if bis then writeln('Данной записи не обнаружено');
nast:=menu1;
menus(nast,nast.m);
tabl1(tek,rab,false);
izm:=0;
tek:=2;
end
else if iz=3 then begin    {изменение данных}
tems:=rab;
tabl1(tek,rab,true);
writeln('Введите уникальный номер'); readln(dlud);
bis:=true;
while tems<>nil do begin
    if tems^.tabl.t1=dlud then begin
        writeln(mm[t,1]);readln(tems^.tabl.t1);
        writeln(mm[t,2]);readln(tems^.tabl.t2);
        writeln(mm[t,3]);readln(tems^.tabl.t3);
        writeln(mm[t,4]);readln(tems^.tabl.t4);
        writeln(mm[t,5]);readln(tems^.tabl.t5);
        break;
    end;
    tems:=tems^.sled;
end;
if bis then writeln('Данной записи не обнаружено');
nast:=menu1;
menus(nast,nast.m);
tabl1(tek,rab,false);
iz 6 =0; tek:=2;
end;
end;

```

```

end;

procedure zapros(num:integer); {Запросы}
var str1,str2,str3:string;
tem1,tem2:cc;
nay:boolean;
zz:tabl2;
begin
  clrscr;
  nay:=false;
  case num of
    2:begin {Найти оценку}
      tem1:=tt1;
      writeln('Введите фамилию'); readln(str1);
      writeln('Введите название предмета');readln(str2);
      while tem1<>nil do begin
        if tem1^.tabl.t2=str1 then begin str1:=tem1^.tabl.t1; break; end;
        tem1:=tem1^.sled;
      end;
      tem1:=tt2;
      while tem1<>nil do begin
        if tem1^.tabl.t2=str2 then begin str2:=tem1^.tabl.t1; break; end;
        tem1:=tem1^.sled;
      end;
      tem1:=tt4;
      while tem1<>nil do begin
        if ((tem1^.tabl.t5=str2) and (tem1^.tabl.t4=str1)) then begin
          textcolor(red);
          writeln('Оценка этого студента-',tem1^.tabl.t2);
          nay:=true; break;
        end;
        tem1:=tem1^.sled;
      end;
    end;
    3:begin {Преподаватель}
      writeln('Введите название предмета');
      readln(str1);
      tem1:=tt2;
      while tem1<>nil do begin
        if tem1^.tabl.t2=str1 then begin str1:=tem1^.tabl.t3; break; end;
        tem1:=tem1^.sled;
      end;
      tem1:=tt3;
      while tem1<>nil do begin
        if tem1^.tabl.t1=str1 then begin
          textcolor(red);
          writeln('Преподаватель-' 6 2
          with tem1^.tabl do write(' ',t2,', ',t3,', ',t4);
          nay:=true; break;
        end;
        tem1:=tem1^.sled;
      end;
    end;
    4:begin {Найти размер стипендии}

```

```

writeln('Введите фамилию студента');
readln(str1);
tem1:=tt1;
while tem1<>nil do begin
  if tem1^.tabl.t2=str1 then begin
    textcolor(red);
    writeln('Стипендия-',tem1^.tabl.t5);
    nay:=true; break;
  end;
  tem1:=tem1^.sled;
end;
end;
5:begin      {Вывод всех студентов с избранной оценкой}
writeln('Введите оценку');
readln(str1);
tem1:=tt4; tem2:=tt1;
textcolor(red);
while tem1<>nil do begin
  if tem1^.tabl.t2=str1 then begin
    str2:=tem1^.tabl.t4;
    while tem2<>nil do begin
      if tem2^.tabl.t1=str2 then begin
        with tem2^.tabl do 6 2
          writeln('Студент-',t3,',',t4,',',t2);
        nay:=true;
      end;
      tem2:=tem2^.sled;
    end;
    end;
    tem2:=tt1; tem1:=tem1^.sled;
  end;
end;
6:begin      {Найти дату сдачи предмета}
writeln('Введите название предмета');
readln(str1);
tem1:=tt2;
while tem1<>nil do begin
  if tem1^.tabl.t2=str1 then begin str1:=tem1^.tabl.t1; break; end;
  tem1:=tem1^.sled;
end;
tem1:=tt4;
while tem1<>nil do begin
  if tem1^.tabl.t5=str1 then begin
    textcolor(red);
    writeln('Дата сдачи-',tem1^.tabl.t3);
    nay:=true;
  end;
  tem1:=tem1^.sled;
end;
end;
end;
textcolor(red);
if not nay then write 6 'Запрос невыполним');
textcolor(3); readln;

```

```

nast:=menu1; menus(nast,nast.m);
tek:=2;
end;

procedure writetip(temr:cc);
begin
clrscr;
write('Введите имя файла');
writeln('в котором хотите сохранить данные');
readln(names);
for i:=1 to 4 do begin
if temr<>nil then begin temr:=nil; end;
case i of
1:begin temr:=tt1; namer:='1'+names; end;
2:begin temr:=tt2; namer:='2'+names; end;
3:begin temr:=tt3; namer:='3'+names; end;
4:begin temr:=tt4; namer:='4'+names; end;
end;
assign(outf,namer); rewrite(outf);
while temr<>nil do begin
write(outf, temr^.tabl);
temr:=temr^.sled;
end;
CLOSE(outf);
end;
nast:=menu1; menus(nast,nast.m); tek:=2;
end;

procedure readtip(temr:cc);
var tems:cc;
begin
clrscr;
write('Введите имя файла');
writeln('из которого надо взять данные'); readln(names);
for i:=1 to 4 do begin
if temr<>nil then begin temr:=nil; end;
if tems<>nil then begin tems:=nil; end;
case i of
1:begin new(tt1); temr:=tt1; namer:='1'+names; end;
2:begin new(tt2); temr:=tt2; namer:='2'+names; end;
3:begin new(tt3); temr:=tt3; namer:='3'+names; end;
4:begin new(tt4); temr:=tt4; namer:='4'+names; end;
end;
assign(outf,namer); reset(outf);
if eof(outf) then begin
case i of
1:begin dispose(tt1);tt1:=nil;end;
2:begin dispose(tt2);tt2:=nil;end;
3:begin dispose(tt3);tt3:=nil;end;
4:begin dispose(tt4);tt4:=nil;end;
end;
end
else begin 6
tems:=temr;

```

```

while temr<>nil do begin
  if eof(outf) then break;
  read(outf,temr^.tabl);
  if eof(outf) then break;
  new(temr^.sled);
  temr:=temr^.sled;
end;
temr^.sled:=nil;
case i of
  1:tt1:=tems;
  2:tt2:=tems;
  3:tt3:=tems;
  4:tt4:=tems;
end;
end;
CLOSE(outf);
end;
nast:=menu1; menus(nast,nast.m); tek:=2;
end;

procedure main;
begin
key:=#0;
if nast.st[1]=menu1.st[1] then begin {Если меню - основное}
  case tek of
    2:readtip(temr);
    3:writetip(temr);
    4,5,7:begin
      nast:=menu2; menus(nast,nast.m);
      if tek=7 then issor:=true;
      if tek=4 then vfile:=true
      else if tek=5 then vfile:=false;
      tek:=2;
    end;
    6:begin
      nast:=menu3; menus(nast,nast.m); tek:=2;
    end;
    8:begin
      nast:=menu4; menus(nast,nast.m); tek:=2;
    end;
    9: begin
      exist:=true;
    end;
  end;
end;

else if nast.st[1]=menu3.st[1] then begin {Если текущее меню-menu3}
  case tek of
    2,3,4:begin
      izm:=tek-1;
      nast:=menu2; menus(nast,nast.m); tek:=2;
    end;
    5:begin
      nast:=menu1;

```

```

    menus(nast,nast.m); tek:=2;
end;
end;
end

else if nast.st[1]=menu4.st[1] then begin {Если текущее меню-menu4}
case tek of
2,3,4,5,6:zaproz(tek);
7:begin
    nast:=menu1;
    menus(nast,nast.m); tek:=2;
end;
end;
end

else if nast.st[1]=menu2.st[1] then begin {Если текущее меню-menu2}
if izm>0 then begin
case tek of
    2:obrabotka(izm, tek-1,tt1);
    3:obrabotka(izm, tek-1,tt2);
    4:obrabotka(izm, tek-1,tt3);
    5:obrabotka(izm, tek-1,tt4);
6:begin
    nast:=menu1; menus(nast,nast.m); izm:=0; tek:=2;
end;
end;
end
else if issor=true then begin
    issor:=false;
case tek of
    2:sort(izm, tek-1,tt1);
    3:sort(izm, tek-1,tt2);
    4:sort(izm, tek-1,tt3);
    5:sort(izm, tek-1,tt4);
6:begin
    nast:=menu1; menus(nast,nast.m); izm:=0; tek:=2;
end;
end;
end

else begin
case tek of
    2:if vfile then tabl11(tek,tt1)
        else tabl1(tek,tt1,false);
    3:if vfile then tabl11(tek,tt2)
        else tabl1(tek,tt2,false);
    4:if vfile then tabl11(tek,tt3)
        else tabl1(tek,tt3,false);
    5:if vfile then tabl11(tek,tt4)
        else tabl1(tek,tt4,false);
6:begin
    nast:=menu1; menus(nast,nast.m); izm:=0; tek:=2;
end 6
end;
end;

```

```

    end;
end;
end;

begin
clrscr;
textBackground(black);
tek:=2;   kr:='-';
    exist:=false;
nast:=menu1; menus(nast,nast.m);
while 1>0 do begin
    if keypressed then key:=readkey;
    case key of
        #80:ramka('+');
        #72:ramka('-');
        #27:exist:=true;
        #13:main;
    end;
    if exist then exit;
krutis;
en 6
end.

```

```

Unit tips;
interface
type
  pered=record
    st:array[1..12] of string;
    m:byte;
  end;

  tabl2=record
    t1,t2,t3,t4,t5:string[12];
  end;
  cc=^tab;
  tab=record
    tabl:tabl2;
    sled:cc;
  end;

var
  menu1,menu2,menu3,menu4:pered;
  mm:array[1..5,1..5] of string[50];
implementation
begin
  with menu1 do begin
    st[1]:='БАЗА ДАННЫХ';
    st[2]:='Загрузка';
    st[3]:='Сохран. в тип. файл';
    st[4]:='Сохран. в текст. файл';
    st[5]:='Просмотр';
    st[6]:='Корректировка';
    st[7]:='Сортировка';
    st[8]:='Запросы';
    st[9]:='Выход';
    m:=9;
  end;
  mm[1,1]:='Студенческий';
  mm[1,2]:='Фамилия';
  mm[1,3]:='Имя';
  mm[1,4]:='Отчество';
  mm[1,5]:='Стипендия';
  mm[2,1]:='Код предмета';
  mm[2,2]:='Название';
  mm[2,3]:='Код преподав.';
  mm[2,4]:='Время учебы';
  mm[2,5]:='Курс';
  mm[3,1]:='Код преподав.';
  mm[3,2]:='Фамилия';
  mm[3,3]:='Имя';
  mm[3,4]:='Отчество';
  mm[3,5]:='Начало работы';
  mm[4,1]:='Код сдачи';
  mm[4,2]:='Оценка';
  mm[4,3]:='Дата сдачи';
  mm[4,4]:='Студенческий';

```

```

mm[4,5]:='Код предмета';

with menu2 do begin
  st[1]:='ПРОСМОТР';
  st[2]:='Студенты';
  st[3]:='Предметы';
  st[4]:='Преподаватели';
  st[5]:='Оценки';
  st[6]:='Выход';
  m:=6;
end;
with menu3 do begin
  st[1]:='КОРРЕКТИРОВКА';
  st[2]:='Добавление';
  st[3]:='Удаление';
  st[4]:='Изменение';
  st[5]:='Выход';
  m:=5;
end;
with menu4 do begin
  st[1]:='ЗАПРОСЫ';
  st[2]:='Найти оценку';
  st[3]:='Кто принимал экзамен';
  st[4]:='Найти размер стипендии';
  st[5]:='Вывод по оценке';
  st[6]:='Дата сдачи экзамена';
  st[7]:='Выход';
  m:=7;
end 6
end.

```

Список использованной литературы

1. Информатика. Базовый курс /Симонович С.В. и др. - СПб: Издательство «Питер», 2006. – 640с.
2. Информатика. Учебное пособие /Под ред. В.Г. Кирия. – Иркутск: ИрГТУ, 2005, часть 2. – 382с.
3. Информатика. Учебное пособие /Ломтадзе В.В., Шишкина Л.П. – Иркутск: ИрГТУ, 2006. – 116с.