# Файлы и базы данных

*текстовые и бинарные файлы, разработка простейших СУБД*

**Контрольные вопросы**

1. Что такое файл? В каких целях используются файлы?

2. Верны ли следующие утверждения?

1) Функцию **fscanf()** можно использовать для чтения данных с

клавиатуры.

2) С помощью **fprintf()** можно вывести информацию на экран.

3) Программа должна явно вызывать функцию **fclose()** для того, чтобы

закрыть файл.

4) В файле произвольного доступа нет необходимости просматривать

все записи для того, чтобы найти определенную запись.

5) Записи в файлах произвольного доступа могут иметь разную длину.

6) Функция **fseek()** может производить поиск только от начала файла.

3. В чем различие между текстовыми и бинарными файлами? В чем

заключаются преимущества тех и других?

4. Какие вы знаете режимы открытия текстовых и бинарных файлов?

5. Режимы “**a+**”, “**r+**” и “**w+**” открывают файл и для записи и для чтения.

Какой из них является наиболее подходящим для изменения уже

существующей информации в файле?

6. Объясните, что такое посимвольный и построчный режимы обработки

файлов, последовательный и произвольный доступ. К каким типам

файлов относятся эти понятия?

7. Какие действия выполняют функции **fopen()**, **fclose()**, **fgetc()**, **fgets()**,

**fputc()**, **fputs()**, **fprintf()**, **fscanf()**?

8. Что делают следующие функции?

void F1(FILE \*ifp, FILE \*ofp)

{

int c;

while ((c = getc(ifp)) != EOF)

putc(c, ofp);

}

void F2(char \*str, FILE \*iop)

{

int c;

while (c = \*str++)

putc(c, iop);

}

9. Как можно очистить уже существующий файл?

10. Как можно организовать быстрый постраничный просмотр текстового

файла, не храня его полностью в памяти, но и не загружая каждый раз

заново?

11. Как можно узнать позицию указателя в файле без использования функции

**ftell()**?

12. Как можно сохранить в файле дерево?

13. Что неправильно в этой программе?

void main(void)

{

int k, \*fp;

fp = fopen(″counting″);

for (k = 0; k < 30; k++)

fputs(fp, ″One, two, three...″);

fclose(″counting″);

}

**Задачи**

Выполнение данной лабораторной работы состоит из двух этапов. На

первом этапе необходимо разработать текстовый документ *Техническое*

*задание* (ТЗ). Второй этап – разработка на основе утвержденного

технического задания *программного продукта*.

**Общие требования**

1. Работа программы должна точно удовлетворять условиям

технического задания.

2. Программа должна иметь удобный и дружественный пользователю

интерфейс.

3. Программа не должна содержать предупреждений (Warnings)

компилятора.

4. Программа должна быть хорошо протестирована: при любом наборе

корректных входных данных она должна работать правильно.

5. Код программы должен быть выдержан в хорошем стиле. Необходимы

комментарии в ключевых местах.

**Специальные требования**

1. Программа должна представлять собой проект, состоящий из

нескольких модулей.

2. Каждый модуль должен быть логично разбит на функции.

Независимо от предметной области, выделяются требования, на

которых защищается квалификация разработчика.

**Требования к качеству**

1. **Устойчивость программы.** Программа не должна терять

работоспособности ни при каких, даже некорректных, действиях

пользователя. Любые действия, грозящие потерей информации,

выполняются только после повторного подтверждения. Вводимая

информация там, где возможно, подвергается логическому контролю.

2. **Полнота информации.** Вся информация поддается просмотру и

редактированию. Удаляемую информацию полезно переносить в

архивные базы для последующего просмотра и восстановления.

Записи, содержащие много полей, можно просматривать как в

табличном, так и в постраничном виде (в виде карточек).

3. **Поиск данных**. Программа должна позволять проводить поиск или

выборку информации по любому полю в базе данных.

4. **Порядок движения.** Движение по меню (вход в подменю)

сопровождается заголовками всех пройденных вершин; возврат

возможен только на предыдущий уровень с сохранением введенной

информации.

5. **Терминологическая среда и интерфейс.** В сообщениях используется

язык и термины, понятные пользователю, а не разработчику («запись»,

«индексация», «can’t open file» и т.д. – не допустимы). Язык диалога – с

соблюдением норм вежливости, цветовая гамма – по общепринятым

рекомендациям.

6. **Использование клавиатуры.** На любом этапе нажатие любой

клавиши (особенно функциональных) должно игнорироваться или

вызывать предусмотренные действия, описанные в средствах помощи.

Привязка действий к клавишам – общепринятая: F1 – помощь, Enter –

согласие, завершение ввода; Esc – отказ, возврат к родительскому

меню; Tab – переход к следующему полю, окну и т.д.; Shift-Tab –

возврат к предыдущем полю и т.д.

7. **Средства помощи.** В любой точке алгоритма в строке подсказки

высвечиваются все активные в данный момент горячие клавиши; в

любой момент при нажатии клавиши F1 выдается контекстно-

зависимый (зависящий от ситуации) текст помощи.

8. **Рекламная заставка.** При запуске программы появляется рекламная

заставка, отражающая суть и возможности программного средства, а

также сведения об авторе. Такую заставку можно разработать

средствами С либо сделать в других редакторах (PowerPoint, Word,

Flash и т.д.) и запускать отдельно.

**Задания**

**Задача 6**

Разработать техническое задание своего варианта.

**Пример разработки технического задания**

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ ПО ПРОЕКТУ**

**СУПЕРМАРКЕТ v1.0**

Колесов М.Ю.

гр. 552001

Обзор программы

Программа Супермаркет предназначена для автоматизации управления покупками

товаров в магазинах. Она является удобным инструментом в осуществлении контроля

над приобретаемым товаром, его количеством, методом оплаты товара и наличием

скидки при его покупке. Программа позволяет контролировать распределение скидок с

помощью системы дисконтных карт. Полезной возможностью программы является

способность хранить всю информацию о покупках в базе данных, из которой в любой

момент можно взять информацию о совершенной ранее сделке.

Программа рассчитана на пользователя, владеющего русским или английским

языком. Программа имеет дружественный интерфейс и может быть использована

пользователем, имеющим минимальный уровень компьютерной грамотности.

Структура данных

В программе использованы следующие структуры данных:

1) для хранения информации о товарах:

Типы:

//штрих-код фиксированной длинны

typedef char[10] tBarcode;

//цена может быть только положительной и целой (для BYB)

typedef unsigned long tPrice;

//наименование товара

typedef char[30] tName;

//количество данного товара (на складе либо купленного)

typedef unsigned long tGoodsNumber;

Структура:

//структура для хранения информации об одиночном товаре

typedef struct {

tBarcode Barcode;

tPrice Price;

tName Name;

tGoodsNumber GoodsNumber;

//для организации связного списка

tGoods \*Next, \*Prev;

} tGoods;

2) для хранения информации о дисконтных картах:

Типы:

//процент скидки

typedef unsigned char tDiscount;

//номер дисконтной карты фиксированной длинны

typedef char[15] tDiscoutCardCode;

Структура:

//структура для хранения информации о дисконтной карте

typedef struct {

tDiscount Discount;

tDiscoutCardCode Code;

//для организации связного списка

tDiscountCard \*Next, \*Prev;

} tDiscountCard;

3) для хранения информации о кредитных картах:

Типы:

//сумма на счете

typedef unsigned char tCardSumm;

//номер кредитной карты фиксированной длинны

typedef char[20] tCreditCardCode;

//пин-код кредитки

typedef char[4] tPinCode;

Структура:

//структура для хранения информации о дисконтной карте

typedef struct {

tCardSumm CardSumm;

tCreditCardCode Code;

tPinCode PinCode

//для организации связного списка

tCreditCard \*Next, \*Prev;

} tCreditCard;

4) для хранения истории о покупках:

Типы:

//представление даты покупки

typedef struct {

char Year;

char Month;

char Day;

char Hour;

char Min;

char Sec;

} tDealDate;

//количество различных типов купленного товара

typedef unsigned int tDifferentGoodsNumber;

//сумма покупки

typedef unsigned long tSumm;

typedef enum {FALSE=0, TRUE=1} tBoolean;

Структура:

//структура, содержащая информацию о покупке

typedef struct {

tDealDate DealDate;

tDifferentGoodsNumber DifferentGoodsNumber;

tSumm Summ;

//использовалась ли дисконтная карта

tBoolean IfUsedDiscount;

//если использовалась дисконтная карта – здесь ее скидка

tDiscount Discount;

//если использовалась дисконтная карта – здесь ее код

tDiscoutCardCode UsedDiscountCardCode;

//использовалась ли кредитная карта

tBoolean IfUsedCreditCard;

//если использовалась кредитная карта – здесь ее код

tCreditCardCode UsedCreditCardCode;

//указатель на список купленных товаров

tGoods \*ListOfBuyedGoods;

//для организации связного списка

tDeal \*Next, \*Prev;

} tDeal;

Архитектура системы

Работа с программой организована в виде навигации по различным пунктам меню.

Карта меню:

1

1) Режим обслуживания покупателей

2) Работа с историей покупок

1) Отображение всей истории покупок

2) Поиск в истории покупок по разным критериям

3) Редактирование истории покупок (ограниченный доступ)

4) Сортировка истории по разным критериям

5) Возврат в предыдущее меню

6) Выход из программы

3) Работа со списком товаров на продажу

1) Отображение всего списка товаров

2) Поиск товара по разным критериям

3) Редактирование списка товаров (ограниченный доступ)

4) Сортировка списка товаров по разным критериям

5) Возврат в предыдущее меню

6) Выход из программы

4) Работа с информацией о дисконтных картах

1) Отображение всего списка дисконтных карт

2) Поиск дисконтной карты по разным критериям

3) Редактирование списка дисконтных карт (ограниченный доступ)

4) Сортировка

5) Возврат в предыдущее меню

6) Выход из программы

5) Смена языка интерфейса

6) Помощь

7) Информация о разработчике

8) Выход из программы

Доступ к пунктам меню осуществляется нажатием на клавиатуре клавиш цифр,

соответствующих позиции в меню. При выборе пункта меню, программа ведет

пользователя далее с помощью интуитивно понятного интерфейса, который

предусматривает обработку корректных и некорректных действий пользователя.

На каждой странице, после выбора пункта меню, внизу экрана находится список

«горячих» клавиш, осуществляющих быстрый вызов часто используемых действий на

данной странице. В нижней части окна находится краткое описание функциональности

данной страницы.

В программе имеется список общих «горячих» клавиш, с помощью которых

возможен доступ к общим свойствам и действиям различных пунктов меню. Список

«горячих» клавиш доступен из пункта меню Помощь.

**Задача 7**

Разработать программное обеспечение по техническому заданию

своего варианта.

**6.1. Книжный магазин**. Прием партии книг на склад; по каждому наименованию указывается: автор, название, реквизиты, количество экземпляров, цена приемки, отпускная цена. Оформление заказа: выбор и продажа нескольких наименований в заданном количестве экземпляров. Отчетность: количество имеющихся на складе книг; их общая стоимость; количество сделанных заказов на книги; количество проданных книг; общая прибыль магазина за период с момента внедрения программы. Списки имеющихся книг, история продаж; функция просмотра. Сохранение введенных данных на диск.

**6.2. Ломбард**. База хранимых товаров и недвижимости: наименование товара, идентификатор клиента, оценочная стоимость, сумма, выданная под залог, дата сдачи, срок хранения.

Список клиентов (ФИО, паспортные данные, дата регистрации, история обслуживания). Редактирование, удаление и просмотр введенной информации. Функции приема/возврата товара, расчет прибыли ломбарда, оценочная стоимость всех хранимых вещей.

**6.4. Стипендия.** Разработать программу для расчетной группы БГУИР, начисляющей стипендию. Ввод, редактирование и удаление информации о студенте, его среднем балле. Расчет стипендии. Начисление именных стипендий при наличии ходатайства кафедры.

Печать ведомости: список студентов, которым была начислена стипендия, общее количество выплаченных денег. Список студентов, получающих повышенную стипендию. Социальная надбавка.

**6.5. Больница**. Больница состоит из нескольких отделений.

В каждом отделении находится определенное количество палат.

Программа формирует список отделений (название, месторасположение), список врачей (ФИО, должность, специальность), списки пациентов (ФИО, адрес, год рождения, диагноз, дата поступления, дата выписки номер палаты). Для каждого пациента назначается свой курс лечения, он вводится пользователем программы. Подготовка к печати истории болезни.

Расчет нагрузка на врача и отделение. Если отделение переполнено – вывод возможной даты принятия нового больного.

**6.6. Общежитие БГУИР.** База данных блоков и комнат в общежитии. Заселение формируется по принципу факультетов: за каждым факультетом закреплены «свои» этажи. Операции заселения и выселения. Формирование списка жильцов (ФИО, факультет, курс, группа, нарушения). При наличии более трех нарушений правил проживания, программа выводит напоминание о необходимости выселения. Список служащих (заведующая, паспортистка, психолог, воспитатели, комендант, старосты этажей), содержит их контактную информацию. Поиск комнаты по фамилии, фамилий по комнате. Отчетность: количество жильцов, количество и номера свободных мест (для всего общежития и для каждого факультета в отдельности).

**6.7. Абитуриент**. Программа для приемной комиссии БГУИР. Формирование списка факультетов и специальностей. Подача заявления на специальность (ФИО, оценки по трем предметам, средний балл аттестата, паспортные данные, место проживания и номер школы абитуриента). Абитуриент имеет право забрать заявление. Зачисление по специальностям согласно поданным заявлениям и выделенным бюджетным местам. Определение проходных и полупроходных баллов по специальностям и факультетам. Отчеты: рейтинг абитуриентов, список поступивших на данную специальность, рейтинг специальностей и факультетов.

**6.9. Администратор гостиницы.** Список всех номеров, которые подразделяются на классы (обычный класс, полулюкс, люкс, королевский номер), число мест. Списки занятых и свободных номеров. Поселение гостей: выбор подходящего номера (при наличии свободных мест), регистрация, оформление квитанции. Отъезд: выбор всех постояльцев, отъезжающих сегодня, освобождение места или оформление задержки с выпиской дополнительной квитанции. Возможность досрочного отъезда с перерасчетом. При заселении создается список постояльцев: ФИО, паспортные данные, номер комнаты, дата приезда и выселения. Поиск постояльца по любому полю. Расчет стоимости проживания.

**6.11. Биржа труда.** База претендентов: ФИО, год рождения, контакты, пол, специальность, образование, семейное положение, город проживания, полная или частичная занятость, желаемый уровень з/п. База вакансий: название компании, город, должность, специальность, образование, пол, семейное положение, возраст, частичная или полная занятость, уровень оплаты. Поиск и регистрация вариантов с той и другой стороны; удаление в архив после трудоустройства, полное удаление при отказе от услуг. Перечень всех возможных трудоустройств по имеющейся БД.

**Подробная информация:** на сайтах.

**6.14. B-Project.** Программа B-Project представляет собой упрощенную версию продукта MS Project, позволяющего управлять проектами. Каждый проект состоит из задач, которые могут выполняться последовательно или параллельно. Задача имеет дату начала и конца. На проект выделяются люди, которые будут работать над ним, и ресурсы (если это необходимо). Программа рассчитывает сроки проекта, ресурсы, определяет критический путь и задачи, которые могут выполняться параллельно, запас времени на их выполнение.

**6.15. e-Shop.by**. Интернет-магазин занимается реализацией компьютерной техники и комплектующих. В каждом разделе существуют подразделы и сами товары. Формирование списка товаров по запросу пользователя. Доступная информация о товаре: название, цена, наличие на складе, описание, гарантия, страна производства и т.д. После выбора продукта, он добавляется в корзину и вычисляется общая цена покупки. При подтверждении покупки, формируется список покупателей: ФИО, контакты, список покупок. Отчеты о продажах, общая сумма выручки, история покупок.

**6.18. Супермаркет**. Программа для кассового аппарата. Имеется база уникальных штрих-кодов (представлены в виде числовых значений). С каждым штрих-кодом связано название товара и его цена. При вводе задается количество нужного товара (ничего не введено – 1 штука, введен 0 – отмена ввода). Вести историю всех покупок. После окончания ввода всех товаров одного покупателя программа спрашивает, есть ли дисконтная карта, и если есть, то требует ее серийный номер. Сверяет по базе дисконтных карт, берет оттуда процент скидки, производит расчет и записывает эту информацию в историю покупки.

**6.20. HotelTracer.** Программа работает с базами данных туристических сайтов (по вашему выбору. Каждая база данных представляет собой двоичный файл, содержащий информацию о свободных номерах в отеле: город, название отеля, адрес отеля, класс номера (обычный класс, полулюкс, люкс, vip-номер), количество мест и стоимость номера. Информация в базах данных обновляется каждые 60 секунд. HotelTracer ищет наилучший отель, соответствующий запросу пользователя. Примеры запросов: “Венеция, престижный номер на двоих, цена не имеет значения”, “София, одиночный номер по минимальной цене”. Функции бронирования номера и снятия брони.

**6.21. Рейтинг студентов**. Программа рассчитывает рейтинг студентов специальности «Информатика». Рейтинг учитывает все оценки, полученные студентами во время сессий – как положительные, так и отрицательные. Можно просмотреть рейтинг по курсам, по группам. Отображается текущий средний балл, рост (положительный или отрицательный) относительно последней сессии, изменение места в рейтинге.

**6. 22. Банк**. Сведения о вкладчиках банка: номер лицевого счета, категория вклада, паспортные данные, текущая сумма вклада (остаток). Операции приема и выдачи сумм по вкладу (разрешенные категорией вклада), расчет процентов на заданную дату. Переводы денежных средств между клиентами, подготовка квитанций.

Отчеты: выписка счета (все операции, проведенные с данным счетом в текущем году); список всех клиентов банка и поиск по нему; сортировка по остаткам вкладов клиентов; сумма остатков.

**Справка:** информация по условиям вкладов – на сайтах банков

**6.23. Путеводитель.** Разработать программу для карманного электронного путеводителя. Международный поиск по городам (в каждом городе существует список мест отдыха (название, адрес, направление деятельности, время работы) – отели, кинотеатры, музеи, ночные клубы, театры и др.). Информация о памятниках и сооружениях, имеющих историческую ценность. Пользователь может выбрать город и получить подробную информацию о местах отдыха. Поиск оптимального отеля по запросу пользователя.

Сформировать записную книжку тех мест, где он побывал, с возможностью добавлять комментарии.

**6.27. Outlook.** Программа реализует ежедневник, содержащий контактный лист владельца (ФИО, контакты, место работы, группа (друзья, коллеги, родственники и т.д.), дата рождения) и данные о делах (звонки, встречи, задания, пометки). Каждое мероприятие хранит в себе дату, время, продолжительность, место проведения, ссылки на контакты, примечание. Программа автоматически генерирует напоминание о днях рождениях. Возможно заполнение ежедневника периодическими событиями на год. Анализ накладок. Возможность просмотра мероприятий на любую дату.