

Министерство образования и науки Российской Федерации  
Федеральное агентство по образованию

Государственное образовательное учреждение  
высшего профессионального образования  
«Оренбургский государственный университет»

Факультет информационных технологий

Кафедра программного обеспечения вычислительной техники  
и автоматизированных систем

## **МЕТОДИЧЕСКИЕ УКАЗАНИЯ К УЧЕБНО-ВЫЧИСЛИТЕЛЬНОЙ ПРАКТИКЕ**

Рекомендовано к изданию Редакционно-издательским советом  
государственного образовательного учреждения  
высшего профессионального образования  
«Оренбургский государственный университет»

Оренбург 2008

## Содержание

|     |   |    |
|-----|---|----|
|     | Введение .....                                | 4  |
| 1   | Цель и задачи учебной практики .....          | 5  |
| 2   | Требования к разрабатываемым программам ..... | 6  |
| 2.1 | Общие требования к программе .....            | 6  |
| 2.2 | Примерные разрабатываемые функции .....       | 6  |
| 2.3 | Примерное содержание отчета .....             | 7  |
|     | Список использованных источников .....        | 8  |
|     | Приложение А Пример оформления отчета .....   | 9  |
|     | Приложение Б Варианты заданий .....           | 37 |

## **Введение**

Учебно-вычислительная практика студентов, обучающихся по специальности 230105.65 – Программное обеспечение вычислительной техники и автоматизированных систем является составной частью образовательной программы и представляет собой форму профессионально-практической подготовки студентов на базах практики.

Практика имеет целью закрепление полученных на первом курсе теоретических знаний и практических знаний.

В методических указаниях представлены цель, задачи практики, примерный план работы, требования к разрабатываемым приложениям, примерное оформление отчета по практике, варианты заданий.

## 1 Цель и задачи учебной практики

Целью учебной практики является приобретение первичных профессиональных умений и навыков путем самостоятельного решения задач алгоритмизации, конструирования и практической реализации программ на ЭВМ с использованием современных технологий программирования.

В процессе учебной практики студенты должны разработать обобщенную схему алгоритма по словесному описанию задачи, провести детализацию отделочных блоков обобщенной схемы, выделить необходимые процедуры и функции, определить наборы логически связанных между собой данных (потoki данных), ввести различные дополнительные средства для обеспечения наглядности и повышения уровня сервиса проектируемой программы, разработать и отладить программу, реализующую спроектированный алгоритм, выполнить на ЭВМ сконструированную программу.

Отчет по учебной практике оформляется в виде пояснительной записки согласно правилам ЕСПД. В качестве приложений разрабатывается «Руководства программиста» по каждой сконструированной программе. Практика завершается защитой отчета.

Для успешного выполнения учебной практики студент должен обладать подробными сведениями о работе в операционной системе, уметь пользоваться библиотеками Турбо Паскаля, владеть знаниями особенностей алгоритмического языка в области модульного программирования, обработки сложных структур данных (файлы, динамические списковые структуры).

Учебная практика базируется на курсах «Информатика», «Алгоритмические языки и программирование» и проводится во 2-м семестре 1-го курса по завершению указанных предметов.

В качестве базы практики используются лаборатории университета, кафедры.

В ходе учебной практики студент должен выполнить одно комплексное задание или три задания по темам:

- 1 Не рекурсивные алгоритмы повышенной сложности.
- 2 Рекурсивные алгоритмы (синтаксические анализаторы, алгоритмы поиска с возвратом и т. д.).
- 3 Начала объектно-ориентированного программирования.

В таблице 1 приведен примерный план работы студента на учебно-вычислительной практике.

Таблица 1 - Примерный план работы

| День недели | Дата     | Вид работы   | Роспись руководителя |
|-------------|----------|--|----------------------|
| ПН          | 03.07.06 | Вводная лекция. Получение задания на учебно-вычислительную практику. Анализ задания.                   |                      |
| ВТ          | 04.07.06 | Словесное описание алгоритма решения. Описание данных и основных подпрограмм в языке программирования. |                      |
| СР          | 05.07.06 | Детальная разработка основных подпрограмм и описание дополнительных подпрограмм.                       |                      |
| ЧТ          | 06.07.06 | Детальная разработка дополнительных подпрограмм.   |                      |
| ПТ          | 07.07.06 | Доработка процедур.  |                      |
| ПН          | 10.07.06 | Разработка интерфейса  |                      |
| ВТ          | 11.07.06 | Начало работы с основной программой.   |                      |
| СР          | 12.07.06 | Корректировка основной программы и завершение работы с ней.  |                      |
| ЧТ          | 13.07.06 | Отладка всей программы.  |                      |
| ПТ          | 14.07.06 | Разработка руководства программиста.   |                      |
| ПН          | 17.07.06 | Оформление отчета.   |                      |
| ВТ          | 18.07.06 | Оформление отчета.   |                      |
| СР          | 19.07.06 | Оформление отчета.   |                      |
| ЧТ          | 20.07.06 | Оформление отчета. Защита работы.  |                      |
| ПТ          | 21.07.06 | Защита работы.   |                      |

## **2 Требования к разрабатываемым программам**

### **2.1 Общие требования к программе**

При разработке программы применить технологию нисходящего проектирования. Логически законченные фрагменты оформить в виде подпрограмм, которым все необходимые данные передаются через список параметров. Использование глобальных переменных следует избегать. Все подпрограммы описываются в отдельных модулях.

В программе предусмотреть меню, содержащее команды для проверки всех возможных действий с заданными в конкретном варианте данными. Само меню также оформляется в виде подпрограммы.

Необходимо предусмотреть защиту от некорректного ввода данных.

Разработать удобный пользовательский интерфейс.

### **2.2 Примерные разрабатываемые функции**

1. Наборы данных (согласно заданному варианту) оформить в виде записей (тип record). Создать одномерный массив указателей на запись

2. Написать функцию для формирования одномерного массива записей, значения которых вводятся с клавиатуры. При вводе записей можно реализовать один из следующих механизмов:

- ввод заранее заданного количества записей;
- ввод до появления записи с заданным признаком;
- диалог с пользователем о необходимости продолжать ввод

3. Написать функцию для записи в файл массива записей

4. Написать функцию чтения в массив записей из файла

5. Написать функцию дополнения уже существующего массива записей новыми записями

6. Написать функцию поиска записей с заданным значением выбранного элемента;

7. Написать функцию постраничного вывода на экран дисплея содержимого массива структур;

8. Написать функцию поиска записей с заданными признаками (например, выбор записей по заданному диапазону значений элемента);

9. Написать функцию упорядочивания массива записей по заданному полю (элементу).

10. Написать функцию полного обновления файла, например, массив записей переписывается в файл после упорядочения.

11. Сохранения данных в файле после завершения программы

### 2.3 Примерное содержание отчета

Первый лист – титульный, второй - задание на учебно-вычислительную практику, третий – содержание, далее по пунктам:

1. Словесное описание алгоритма.
2. Спецификация глобальных констант и переменных.
3. Спецификация процедур и функций.
4. Руководство пользователя.

Список использованных источников.

Приложения: контрольный пример, схемы алгоритмов, исходные тексты программы.

## Список использованных источников

- 1 **Подбельский В.В.** Программирование на Си Учебн. пос./ В.В. Подбельский, С.С. Фомин.-2-е изд., доп.-М.:Финансы и статистика, 2002.-600с.:ил.
- 2 **Березин Б.И.** Начальный курс С и С++.-М.: ДИАЛОГ-МИФИ, 2004.-288с.
- 3 **Иванова Г.С.** Основы программирования Учеб. для вузов/ Г.С.Иванова .-М.МГТУ им Н.Э.Баумана, 2001 – 392с
- 4 **Павловская Т.А.** С/С++. Программирование на языке высокого уровня – СПб.: Питер, 2006.- 461с.: ил.
- 5 **Подбельский В.В.** Язык Си++ Учебн. пос./ -5-е изд., доп.-М.: Финансы и статистика, 2003.-560с.:ил.
- 6 **Подбельский В.В.** Практикум по программированию на языке Си (+CD).: Учеб. пособие.-М.: Финансы и статистика, 2004.-576с.:ил.
- 7 **С/С++. Структурное программирование: Практикум / Т.А. Павловская, Ю.А. Щупак.- СПб.: Питер, 2007.- 239с.: ил.**
- 8 **Прата, Стивен** Язык программирования С++. Лекции и упражнения. Учебник: Пер. с англ./Стивен Прата – СПб.: ООО «ДиаСофтЮП», 2003. – 1104 с.



**Приложение А**  
*(обязательное)*  
**Пример оформления отчета**

Министерство образования и науки Российской Федерации  
Федеральное агентство по образованию

Государственное общеобразовательное учреждение  
высшего профессионального образования  
«Оренбургский государственный университет»

Факультет информационных технологий

Кафедра ПОВТАС

**ОТЧЁТ**  
**по учебно-вычислительной практике**  
на базе кафедры ПОВТАС

ОГУ 230105.65.9007.22 П

Руководитель

\_\_\_\_\_ А.Н. Калинин  
«\_\_» \_\_\_\_\_ 2008 г.

Исполнитель

студент гр. 03ПОВТ-1  
\_\_\_\_\_ С.В. Лежнев  
«\_\_» \_\_\_\_\_ 2008 г.

Оренбург 2008

Министерство образования и науки Российской Федерации  
Федеральное агентство по образованию

Государственное общеобразовательное учреждение  
высшего профессионального образования  
«Оренбургский государственный университет»

Факультет информационных технологий

Кафедра ПОВТАС

**Задание на учебно-вычислительную практику  
«Склад»**

Исходные данные: текст задания

а) Существует файл склад, в котором хранится информация об ассортименте продукции на складе:

- наименование продукции;
- цена единицы продукции;
- количество единиц наличия;
- сумма.

б) Имеется сеть магазинов, которые получают продукцию со склада. Информация хранится в файле расход:

- потребитель;
- наименование продукции;
- количество единиц отпущено;
- сумма;
- дата.

Программа следит за движением продукции со склада, обеспечивает ввод, удаление и просмотр информации, каждому потребителю выдаёт счёт-накладную.

Разработать: 1) Алгоритм и программу;  
2) Разработать интерфейс;

Дата выдачи задания «\_\_» \_\_\_\_\_ 2008 г.

Руководитель

\_\_\_\_\_ А.Н.Калинин

Исполнитель

студент группы 03ПОВТ-1

\_\_\_\_\_ С.В. Лежнев

Срок защиты работы «\_\_» \_\_\_\_\_ 2008 г.

## Содержание

|   |   |    |
|---|---|----|
| 1 | Постановка задачи .....                         | 3  |
| 2 | Обоснование выбора языка программирования ..... | 4  |
| 3 | Словесный алгоритм .....                        | 5  |
| 4 | Спецификация переменных .....                   | 6  |
| 5 | Спецификация функций .....                      | 8  |
| 6 | Руководство пользователя .....                  | 9  |
|   | Список использованных источников .....          | 7  |
|   | Приложение А Схемы алгоритмов .....             | 11 |
|   | Приложение Б Тестовый пример .....              | 18 |
|   | Приложение В Текст программы .....              | 21 |

## **1 Обоснование выбора языка программирования**

Для разработки программы использован язык C++, как наиболее популярный и имеющий достаточную функциональность для решения поставленной задачи. C++ постоянно находился в состоянии изменения, поскольку происходило его дополнение новыми функциональными возможностями. Этот язык совмещает несколько парадигм – объектно-ориентированное программирование, обобщенное программирование и традиционно-процедурное программирование.

Этот язык стал одним из основных языков программирования в 90-е годы и сохраняет свои позиции в первом десятилетии XXI века. Унаследовав от языка C такие свойства, как эффективность, компактность, быстроту выполнения и переносимость программ.

## 2 Описание разработки структуры программы

Для облегчения и наглядности будем проводить обработку данных, находящихся в динамической памяти. При разработке программного комплекса был использован принцип модульного программирования (создан специальный модуль SKLADUNIT.H, который содержит в себе некоторые вспомогательные функции), а также стандартные библиотеки языка C++. Графический режим не использовался.

Для хранения и обработки данных используем структурированный тип, поэтому описываем его до описания функций. Исходный код программы состоит из главного файла (SKLAD.CPP), вспомогательного модулей (PRODUCT.H, POTREB.H, GRAPH.H). Для успешной компиляции необходимы стандартные библиотеки языка C++. В главном файле программы содержится функция main(). В модуле содержатся необходимые для работы программы функции. Программа начинается с вызова функции main().

### 3 Словесный алгоритм

Программа начинается с вызова функции `main()`. Первоначально на экране появляется меню, состоящее из 2 пунктов:

1. просмотр товаров на складе
2. просмотр потребителей.

Выбор какого-либо пункта осуществляется путём передвижения курсора вверх или вниз с помощью функции `up()` и `down()` соответственно.

При выборе первого пункта на экран выводится информация о товарах на складе. При нажатии клавиши «Del» происходит удаление выбранного товара с помощью функции `del()`, при нажатии «1» происходит добавление количества выбранного товара со склада с помощью функции `add_prod()`, при нажатии «2» - происходит редактирования выбранного товара на складе, программа просит ввести новое имя и количество, осуществляемое с помощью функции `correct()`, при нажатии «3» - запускается функция `buy()`, которая осуществляет покупку магазином, выбранным из предоставленного списка, нажатие клавиши «4» дает возможность добавить новый товар на склад с помощью функции `new_prod()`, при нажатии клавиши «5» происходит поиск товара на складе с помощью функции `seach()`, при нажатии клавиши Esc программа возвращается к предыдущему пункту.

При выборе второго пункта на экране появляется список магазинов-потребителей. При нажатии клавиши Enter происходит выбор одного из них и появляется вся информация о выбранном магазине. Нажатие клавиши «1» дает возможность добавить нового потребителя с помощью функции `new_potr()`, при нажатии клавиши «2» происходит удаление выбранного потребителя с помощью функции `del_potr()`.

## 4 Спецификация переменных

| Идентификатор    | Тип     | Описание                     |
|------------------|---------|------------------------------|
| Главный файл     |         |                              |
| Potreb[10]       | Potr    | Информация о потребителей    |
| ntov             | Int     | Количество товаров на складе |
| npotr            | int     | Количество потребителей      |
| strmenu[30][30]  | Char    | Список имен для меню         |
| Color            | Int     | Текущий цвет фона            |
| Da               | date    | Текущая дата                 |
| Модуль Product.h |         |                              |
| Product[30]      | product | Информация о товарах         |

## 5 Спецификация функций

| Название     | Описание   |
|--------------|--|
| Главный файл |  |
| main()       | <p>Главная функция программы. В зависимости от выбранного пункта меню вызывает соответствующую функцию: see_tov(), see_potr) . Параметры не получает. Значений не возвращает.</p> <p>Локальные переменные:</p> <p>m , f char Вспомогательная переменная</p> <p>Key int Номер клавиши</p> <p>Cr Int Номер текущего пункта меню</p> <p>Old Int Номер предыдущего пункта меню</p> |
| flush(void)  | <p>Функция записи всей информации в файлы</p> <p>Локальные переменные:</p> <p>SKLAD FILE* Указатель на Файл SKLAD</p> <p>RASXOD FILE* Указатель на Файл RASXOD</p> <p>i , j int Счетчик</p>  |
| Файл Graph.h |  |



|                                      |  |
|--------------------------------------|--|
| new_menu(int cr,int old)             | <p>Меняет цвет экрана при выборе какого-либо пункта из меню.</p> <p>Локальные переменные:</p> <p>x,y int Координаты курсора</p>  |
| window1(int x1,int y1,int x2,int y2) | <p>Рисует на экране окно.</p> <p>Локальные переменные:</p> <p>I int Счетчик</p>  |
| vvod(void)                           | <p>Функция ввода целых значений, которая возвращает только цифры.</p> <p>Локальные переменные:</p> <p>M,c,cif,d Int Вспомогательные переменные</p> <p>Key Int Номер введенной клавиши</p> <p>array[] Int Массив кодов клавиш цифр</p> <p>kol Int Введенное число</p> |
| up(int *old,int *cr)                 | <p>Функция изменения позиции текущего указателя, т.е. перевод его вверх (смена цвета текущей позиции и верхней).</p>   |
| Файл Product.h                       |  |

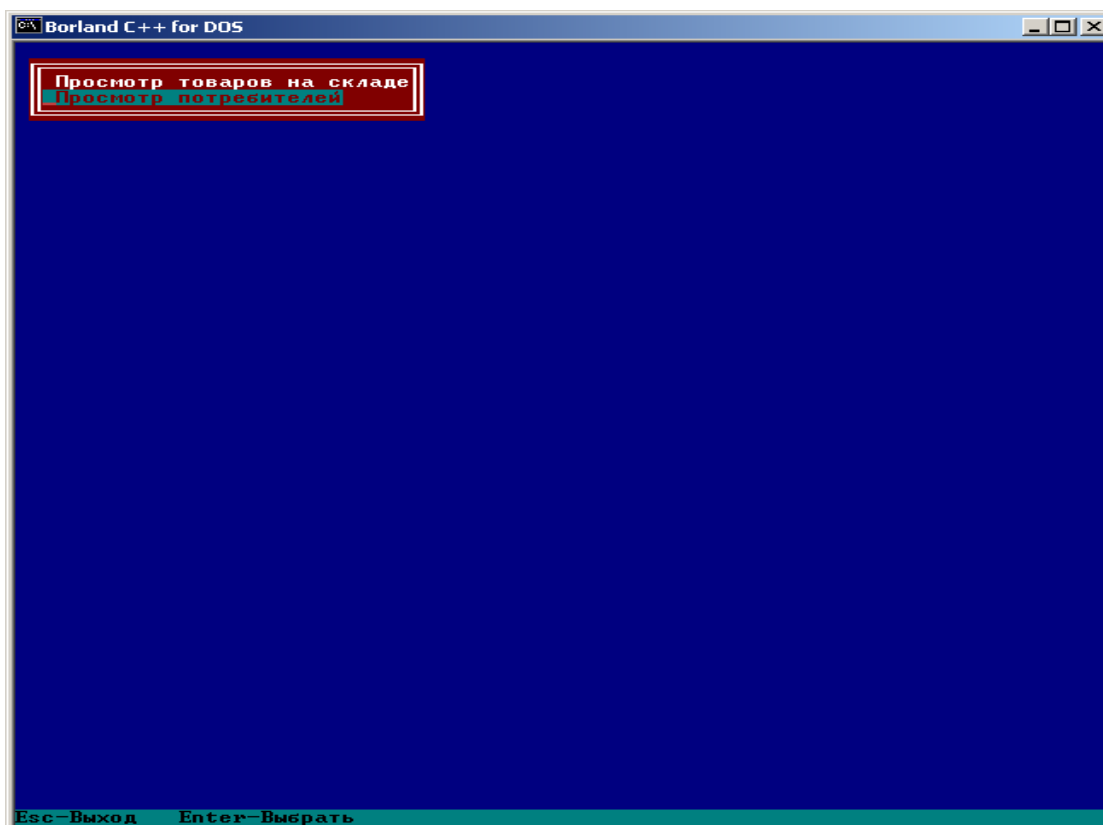
|                              |   |
|------------------------------|---|
| down(int *old,int *cr,int n) | Функция изменения позиции текущего указателя, т.е. перевод его вверх (смена цвета текущей позиции и нижней).  |
| int read_prod(void)          | Функция считывает из файла СКЛАД информацию о продукции, хранящейся на складе, и выводит её на экран.<br>Локальные переменные:<br>RASXOD FILE* Указатель на Файл RASXOD<br>I int Счетчик<br>j int Счетчик   |
| new_prod(void)               | Функция просит ввести с клавиатуры имя нового товара на складе, а также высчитывает его цену, кол-во и сумму.   |
| del_prod(char number)        | Функция удаляет какой-либо товар на складе.<br>Локальные переменные:<br>j int Счетчик   |
| corect(int number)           | Функция изменяет имя какой-либо товар на складе.  |
| schet(int p1,int p2)         | Выводит на экран счёт-накладную.<br>Локальные переменные:<br>d date Текущая дата  |
| int buy(int number)          | Покупка товара на складе. Количество продукции вводится с клавиатуры.<br>Локальные переменные:<br>m,f,kol Int Вспомогательные переменные<br>Key int Номер нажатой клавиши<br>cr Int Номер текущего пункта меню<br>Old Int Номер предыдущего пункта меню |
| int seach(void)              | Функция поиска товара<br>Локальные переменные:<br>Name[20] Char Имя искомого товара<br>m Int Вспомогательная переменная<br>j Int Счетчик  |

|                   |  |
|-------------------|--|
| int see_tov(void) | Функция просмотра продукции на складе.<br>Локальные переменные:<br>m,f,kol,u,d,l,p int Вспомогательная переменная<br>Cr int Номер текущего пункта меню<br>old int Номер предыдущего пункта меню<br>key Int Номер клавиши |
| Файл Potreb.h     |  |

|                           |   |
|---------------------------|---|
| el_potr(int number)       | <p>Функция удаления потребителя.<br/>Локальные переменные:</p> <p>i int Счетчик</p> <p>j int Счетчик</p> <p>j Int Счетчик</p>   |
| int read_potr(void)       | <p>Считывает из файла РАСХОД информацию о потребителях, и выводит её на экран. Значений не возвращает.<br/>Локальные переменные:</p> <p>RASXOD FILE* Указатель на Файл RASXOD</p> <p>i int Счетчик</p> <p>j int Счетчик</p>   |
| del_pok(int p,int number) | <p>Функция удаления текущей покупки</p>   |
| see_potreb(int number)    | <p>Функция просмотра информации о потребителе (информация о купленном товаре, дата покупки).<br/>Локальные переменные:</p> <p>m,f,kol,u,d,l,p int Вспомогательная переменные</p> <p>Cr int Номер текущего пункта меню</p> <p>old int Номер предыдущего пункта меню</p> <p>key Int Номер клавиши</p> |
| new_potr(void)            | <p>Функция просит ввести с клавиатуры имя нового потребителя.</p>   |
| see_potr(void)            | <p>Функция просмотра списка потребителей (возможность добавления и удаления потребителя)<br/>Локальные переменные:</p> <p>m,f int Вспомогательная переменные</p> <p>Cr int Номер текущего пункта меню</p> <p>old int Номер предыдущего пункта меню</p> <p>key Int Номер клавиши</p>                 |

## 6 Руководство пользователя

Программа - простейшая база данных, использующая для хранения информации динамическую память компьютера. При запуске программы появится главное меню программы состоящее из 2 пунктов:



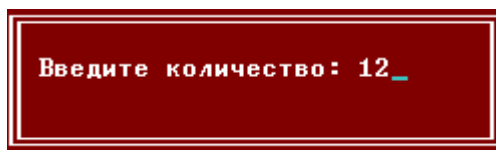
1) Просмотр товаров на складе

2) Просмотр потребителей

Выбор какого-либо пункта осуществляется с помощью клавиш вверх и вниз.

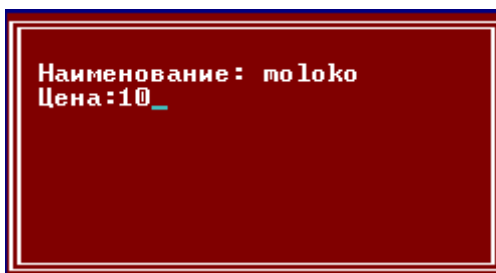
|         |    |     |      |
|---------|----|-----|------|
| KEFIR   | 14 | 200 | 2800 |
| smetana | 10 | 110 | 1100 |
| no loko | 20 | 31  | 620  |
| asd     | 21 | 41  | 861  |
| asdf    | 12 | 41  | 492  |
| sad     | 12 | 31  | 372  |
| ad      | 21 | 13  | 273  |
| adaf    | 12 | 31  | 372  |
| adjkb   | 12 | 31  | 372  |
| dlak    | 12 | 31  | 372  |
| akdl;n  | 12 | 31  | 372  |
| ad      | 12 | 231 | 2772 |
| a;d     | 12 | 31  | 372  |
| sa      | 12 | 31  | 372  |
| s       | 12 | 31  | 372  |
| l;sd    | 12 | 31  | 372  |
| l;a     | 12 | 31  | 372  |
| kalbasa | 1  | 12  | 12   |
| kjf     | 12 | 31  | 372  |
| ;kfn    | 12 | 11  | 132  |

При выборе первого пункта на экран выводится информация о товарах на складе. При нажатии клавиши «Del» происходит удаление выбранного товара, при нажатии «1» происходит добавление количества выбранного товара со склада



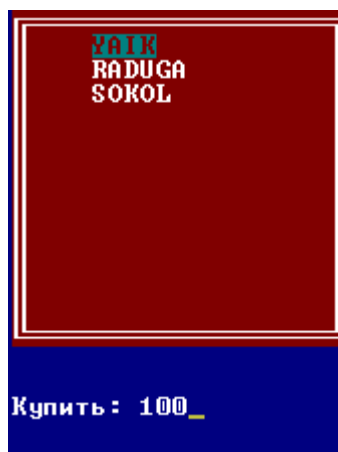
```
Введите количество: 12_
```

При нажатии «2» - происходит редактирования выбранного товара на складе, программа просит ввести новое имя и цену.



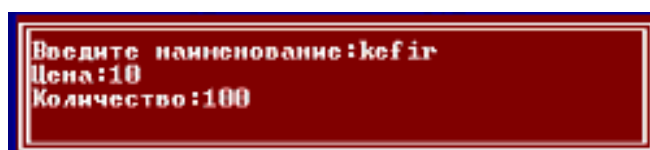
```
Наименование: молоко  
Цена:10_
```

При нажатии «3» - программа осуществляет покупку магазином, выбранным из предоставленного списка



```
МАГАЗИНЫ  
RADUGA  
SOKOL  
Купить: 100_
```

Нажатие клавиши «4» дает возможность добавить новый товар на склад



```
Введите наименование:kefir  
Цена:10  
Количество:100
```

При нажатии клавиши «5» происходит поиск товара на складе.

Введите имя товара: adad\_

При нажатии клавиши Esc программа возвращается к предыдущему пункту.

При выборе второго пункта на экране появляется список магазинов-потребителей.

| Список магазинов |              | Покупки за 7 месяц 2004 года |            |       |           |
|------------------|--------------|------------------------------|------------|-------|-----------|
| УДИК             | Наименование | Цена                         | Количество | Сумма | Дата      |
| RADUGA           | МОЛОКО       | 11                           | 100        | 1100  | 10.7.2004 |
| СОКОЛ            | kalbasa      | 100                          | 31         | 3100  | 14.7.2004 |
|                  | KEFIR        | 14                           | 100        | 1400  | 16.7.2004 |

Esc-Назад Del-Удалить

При нажатии клавиши Enter происходит выбор одного из них и появляется вся информация о выбранном магазине. Нажатие клавиши «1» дает возможность добавить нового потребителя, при нажатии клавиши «2» происходит удаление выбранного потребителя.

## Список использованных источников

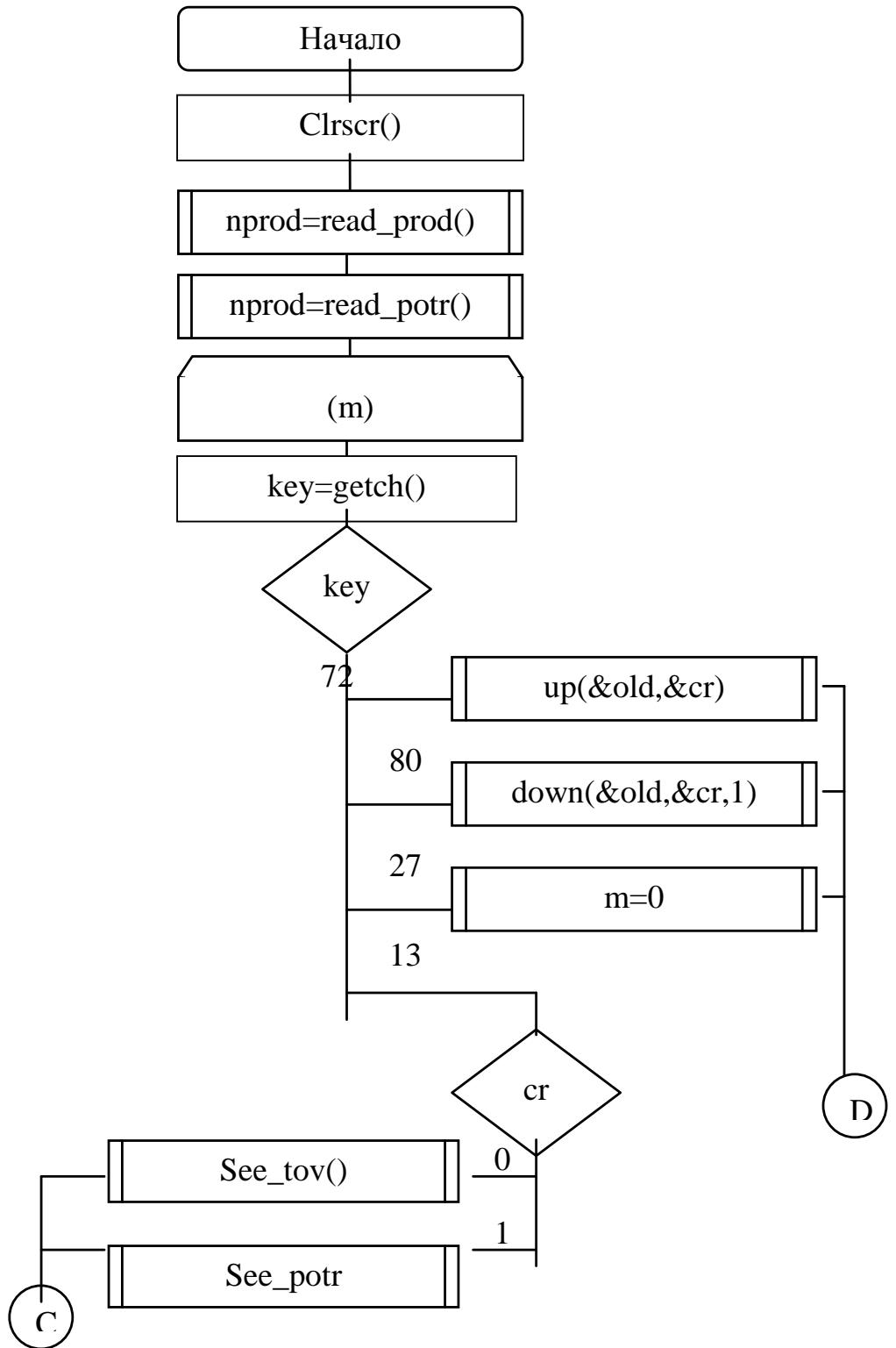
- 1 **Франка, П.** С++ Учебный курс, 2003.
- 2 **Нейбауэр, А.** Моя первая программа на С/С++, 2000.
- 3 **Прата, Стивен** Язык программирования С++. Лекции и упражнения. Учебник: Пер. с англ./Стивен Прата – СПб.: ООО «ДиаСофтЮП», 2003. – 1104 с.
- 4 **Подбельский, В.В.** Язык СИ++ : Учеб. пособие. – 3-е изд., дораб. – М.: Финансы и статистика, 1998. – 560 с.: ил.
- 5 **Страуструп, Б.** Язык программирования С++.: Москва, 1999.

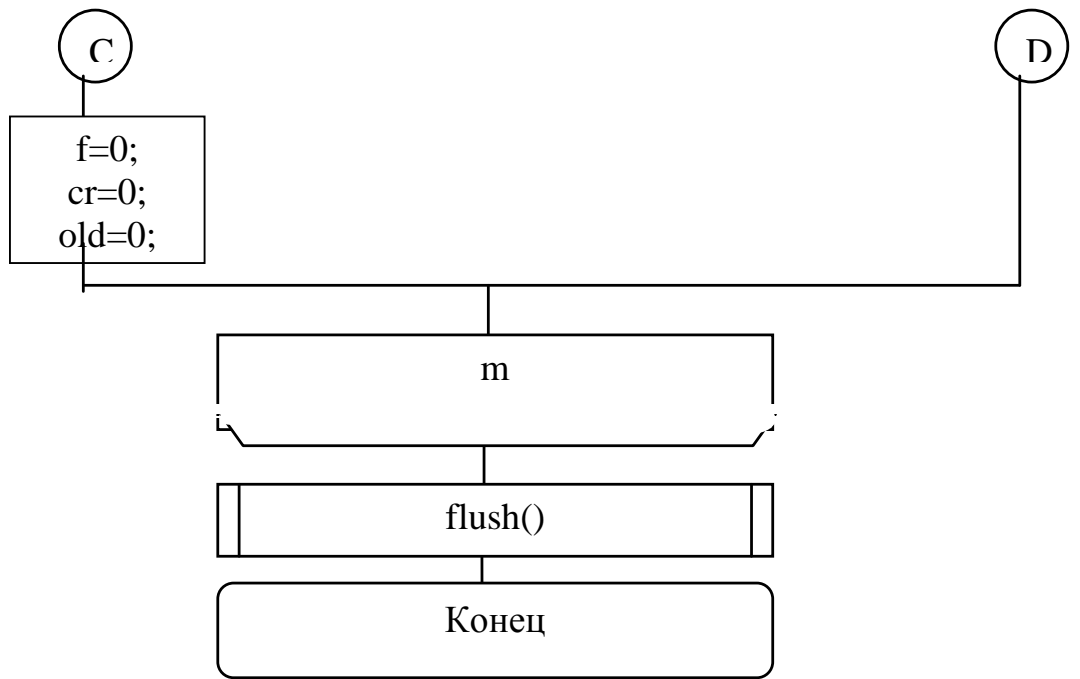


# Приложение А

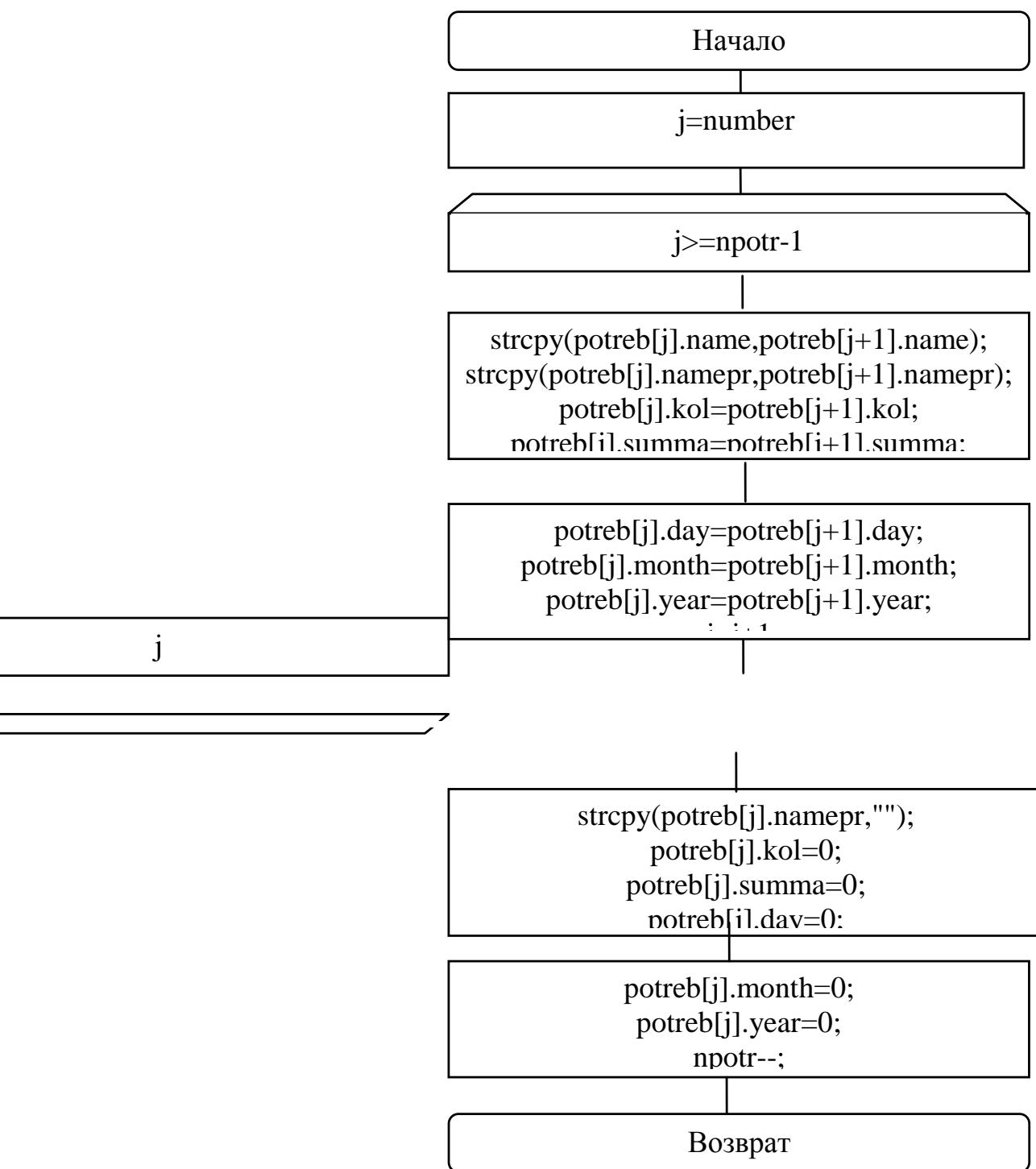
## Схемы алгоритмов

Функция main()





Функция del\_potr(int number)





## Приложение Б Текст программы

### Основная программа

```
#include <stdio.h>
#include <math.h>
#include <conio.h>
#include <iomanip.h>
#include <string.h>
#include <stdlib.h>
#include <alloc.h>
#include <dos.h>
#include "graph.h"
#include "product.h"
#include "potreb.h"
////////////////////////////////////
potr potreb[10];
int ntov;
int npotr;
char strmenu[30][30];
int color;
struct date da;
////////////////////////////////////
void flush(void) {
    FILE*SKLAD,*RASXOD;
    SKLAD=fopen("SKLAD.txt","wt");
    for(int i=0;i<ntov;i++) {
        fprintf(SKLAD,"\n%s %i
%i",product[i].name,product[i].price,product[i].kol);
    }
    fclose(SKLAD);
    RASXOD=fopen("RASXOD.txt","wt");
    for(i=0;i<npotr;i++) {
        fprintf(RASXOD,"\n%s %i",potreb[i].name,potreb[i].npok);
        for(int j=0;j<potreb[i].npok;j++) {
            fprintf(RASXOD,"\n%s %i %i %i %i
%i",potreb[i].prod[j].name,potreb[i].prod[j].price,potreb[i].prod[j].kol,potreb[i].p
rod[j].day,potreb[i].prod[j].month,potreb[i].prod[j].year);
        }
    }
    fclose(RASXOD);
}
```

```

void main()
{
clrscr();
char m=1,f=0;
int cr=0,old=0;
char key=0;

ntov=read_prod();
npotr=read_potr();
while (m)
{
if (f==0)
{
window(1,1,80,50);
textbackground(1);
textcolor(0);
clrscr();
window(1,50,80,50);
textbackground(3);
textcolor(0);
clrscr();
printf("Esc-,леⓂ Enter-,лŸа вМ");
color=4;
window1(2,2,30,5);
textbackground(color);
textcolor(15);
strcpy(strmenu[0],"ЦаⓂб-Ⓜва вŸ аŸ бС« ⓂГ");
strcpy(strmenu[1],"ЦаⓂб-Ⓜва ĨⓂвГŸĚвГ«Г©");
gotoxy(2,2);
for(int j=0;j<2;j++)
{
if (j==0)
{
textbackground(3);
textcolor(4);
gotoxy(2,2+j);
printf("%s",strmenu[j]);
}
else
{
gotoxy(2,2+j);
textbackground(color);
textcolor(15);
printf("%s",strmenu[j]);
}
}
}
}

```

```

    }
    }
    gotoxy(2,2);
    f=1;
    }
    key=getch();
    switch(key)
    {
    case 72: up(&old,&cr);
    break;
    case 80: down(&old,&cr,1);
    break;
    case 27: m=0;
    break;
    case 13:
    {

switch(cr)
{
    case 0: see_tov();
    break;
    case 1: see_potr();
    break;
    }
    f=0;
    cr=0;
    old=0;
    }
    break;
    }
}
flush();
}

```

### Модуль Graph.h

```

extern int color;
extern char strmenu[30][30];

void new_menu(int cr,int old) {
    int x,y;
    x=wherex();
    y=wherey();
    gotoxy(wherex(),wherey()-(cr-old));
}

```

```

textbackground(color);
textcolor(15);
cprintf("%s",strmenu[old]);
gotoxy(x,y);
textbackground(3);
textcolor(4);
cprintf("%s",strmenu[cr]);
gotoxy(x,y);
}
void window1(int x1,int y1,int x2,int y2) {
    window(x1,y1,x2,y2);
    textbackground(4);
    textcolor(15);
    clrscr();
    gotoxy(2,1);
    for(int i=2;i<=x2-x1;i++) printf("%c",205);
    gotoxy(2,y2-y1+1);
    for(i=2;i<=x2-x1;i++) printf("%c",205);
    for(i=2;i<=y2-y1;i++)
    {
        gotoxy(1,i);
        printf("%c",186);
    }
    for(i=2;i<=y2-y1;i++) {
        gotoxy(x2-x1+1,i);
        printf("%c",186);
    }
    gotoxy(1,1);
    printf("%c",201);
    gotoxy(x2-x1+1,1);
    printf("%c",187);
    gotoxy(x2-x1+1,y2-y1+1);
    printf("%c",188);
    gotoxy(1,y2-y1+1);
    printf("%c",200);
}

```



## **Приложение Б**

*(обязательное)*

### **Варианты заданий**

#### **Вариант 1**

Разработать программу для работы с БД «Книги почтой», включающую следующую информацию: код книги, автор, название, год издания, цена книги, ФИО читателя, год рождения читателя, образование, адрес и т.д. Разработать удобный пользовательский интерфейс и организовать добавление, удаление, поиск и корректировку записей в БД.

Выяснить самый читающий слой населения, самую продаваемую книгу, наиболее популярного автора и т.д. Дать графическую интерпретацию полученным результатам в виде графиков, круговых диаграмм и гистограмм.

Выяснить тенденцию развития рынка книг: определить зависимость общего объема продаж книг от года и сделать прогноз на ближайших три года.

#### **Вариант 2**

Разработать программу для работы с БД «Студент», включающую следующую информацию: ФИО студента, год рождения, домашний адрес, факультет, специальность, курс, какое ср. учебное заведение закончил и т.д. Разработать удобный пользовательский интерфейс и организовать добавление, удаление, поиск и корректировку записей в БД.

Определить факультет, курс, имеющий наибольшее количество неуспевающих, отличников и хорошистов. Определить десятку лучших студентов и т. д. Дать графическую интерпретацию полученным результатам в виде графиков, круговых диаграмм и гистограмм.

Выяснить тенденцию роста или уменьшения неуспевающих по годам и сделать прогноз на ближайших два года.

#### **Вариант 3**

Разработать программу для работы с БД «Наряд», включающую следующую информацию: шифр наряда, дата( год, месяц, число), номер цеха, табельный номер, код операции, норма времени, количество изготовленных деталей и др. Разработать удобный пользовательский интерфейс и организовать добавление, удаление, поиск и корректировку записей в БД.

Определить количество нарядов определенного вида по годам, месяцам, числам, самые производительные цеха, самые трудоемкие операции и т. д. Дать графическую интерпретацию полученным результатам в виде графиков, круговых диаграмм и гистограмм.

Определить рост производительности цехов по месяцам и сделать прогноз на ближайшие два месяца.

#### **Вариант 4**

Разработать программу для работы с БД «Продуктовый магазин», включающую следующую информацию: код продукта, название, вид упаковки, дата поступления, срок хранения, объем закупки, объем продажи, цена и т.д. Разработать удобный пользовательский интерфейс и организовать добавление, удаление, поиск и корректировку записей в БД.

Выяснить самый продаваемый продукт, продукты с истекшим сроком хранения, организовать уценку товаров, пролежавших более половины своего срока хранения и т. д. Дать графическую интерпретацию полученным результатам в виде графиков, круговых диаграмм и гистограмм.

Выяснить тенденцию развития дохода магазина (зависимость дохода от месяца) и сделать прогноз на ближайшие три месяца.

### **Вариант 5**

Разработать программу для работы с БД «Бухгалтерия», включающую следующую информацию: ФИО работника, образование, должность, дата поступления на работу, оклад и т.д. Разработать удобный пользовательский интерфейс и организовать добавление, удаление, поиск и корректировку записей в БД.

Выяснить количество работников с высшим, техническим и среднеспециальным образованием, определить десятку самых высокооплачиваемых сотрудников, определить старейших работников предприятия и т. д. Дать графическую интерпретацию полученным результатам в виде графиков, круговых диаграмм и гистограмм.

Определить тенденцию роста фонда заработной платы (зависимость суммарного оклада от года) и сделать прогноз на ближайшие два года.

### **Вариант 6**

Разработать программу для работы с БД «Сотрудники НИИ», включающую следующую информацию: № отдела, табельный номер, ФИО, № темы над которой работает сотрудник, продолжительность работы в месяцах, код должности, звания, размер заработной платы и т.д. Разработать удобный пользовательский интерфейс и организовать добавление, удаление, поиск и корректировку записей в БД.

Выяснить самую продолжительную работу, наиболее высокооплачиваемого сотрудника, распределение сотрудников по темам и званиям т. д. Дать графическую интерпретацию полученным результатам в виде графиков, круговых диаграмм и гистограмм.

Выяснить прирост сотрудников с определенным званием по годам и сделать прогноз о их количестве на ближайшие два года.

### **Вариант 7**

Разработать программу для работы с БД «Реализованный товар», включающую следующую информацию: номер магазина, номер секции, номер чека, наименование товара, артикул товара, цена товара, дата продажи

и т.д. Разработать удобный пользовательский интерфейс и организовать добавление, удаление, поиск и корректировку записей в БД.

Выяснить самый доходный магазин, товар, секцию, самый доходный месяц и т. д. Дать графическую интерпретацию полученным результатам в виде графиков, круговых диаграмм и гистограмм.

Выяснить развитие товарооборота в зависимости от месяца и сделать прогноз на ближайшие три месяца.

### **Вариант 8**

Разработать программу для работы с БД «Аэрофлот», включающую следующую информацию: номер рейса, пункт назначения, время вылета, время прибытия, количество свободных мест в салоне, цена билета и т.д. Разработать удобный пользовательский интерфейс и организовать добавление, удаление, поиск и корректировку записей в БД.

Определить число рейсов до определенного пункта назначения, самые продолжительные рейсы, наиболее дорогостоящие рейсы и т. д. Организовать оптимальный выбор рейса по заказу пассажира. Дать графическую интерпретацию полученным результатам в виде графиков, круговых диаграмм и гистограмм.

Выяснить динамику роста или снижения цен на билеты по годам и сделать прогноз на ближайшие два года.

### **Вариант 9**

Разработать программу для работы с БД «Аптека», включающую следующую информацию: код лекарства, название, дата выпуска, срок хранения, форма изготовления, объем партии и т.д. Разработать удобный пользовательский интерфейс и организовать добавление, удаление, поиск и корректировку записей в БД.

Выяснить самые крупные объемы партий лекарств, наиболее быстро портящиеся и т. д. Организовать удаление из БД лекарств с просроченным сроком годности. Дать графическую интерпретацию полученным результатам в виде графиков, круговых диаграмм и гистограмм.

Выяснить динамику роста или снижения объемов партий по месяцам и сделать прогноз на ближайшие два месяца.

### **Вариант 10**

Разработать программу для работы с БД «Абонент телефона», включающую следующую информацию: ФИО абонента, адрес, номер телефона, год установки телефона, время междугородних переговоров, оплата за переговоры и т.д. Разработать удобный пользовательский интерфейс и организовать добавление, удаление, поиск и корректировку записей в БД.

Выяснить абонентов, имеющих наибольшее время междугородних переговоров, год, в котором установлено наибольшее число телефонов,

абонентов, имеющих долг за переговоры и т. д. Дать графическую интерпретацию полученным результатам в виде графиков, круговых диаграмм и гистограмм.

Выяснить динамику прибыли от междугородних переговоров по годам и сделать прогноз на ближайшие два года.

### Вариант 11

Разработать программу для работы с БД «Подписка», включающую следующую информацию: индекс издания, наименование, цена подписки с доставкой, цена подписки без доставки, количество подписчиков, на какой срок подписался (1 месяц, ...1 год), наличие льгот на подписку и т.д. Разработать удобный пользовательский интерфейс и организовать добавление, удаление, поиск и корректировку записей в БД.

Определить самые популярные издания и виды услуг, организовать оптимальный поиск вида подписки по желанию подписчика и т. д. Дать графическую интерпретацию полученным результатам в виде графиков, круговых диаграмм и гистограмм.

Определить динамику развития рынка подписки по годам и сделать прогноз на ближайшие два года.

### Вариант 12

Разработать программу с удобным пользовательским интерфейсом, реализующую следующие функции:

Синтаксический и семантический анализ понятия список геометрических фигур

$$СГФ ::= \left\{ \begin{array}{l} \text{геометрическая фигура} \\ \text{геометрическая фигура } \{ ; \text{список геометрических фигур} \} \end{array} \right\}$$

$$\text{геометрическая фигура} ::= \left\{ \begin{array}{l} \text{треугольник} \\ \text{окружность} \end{array} \right\}$$

треугольник ::= {вершина; вершина; вершина}

окружность ::= {центр; радиус}

радиус ::= {число с точкой}

вершина ::= {абсцисса, ордината}

центр ::= {вершина}

абсцисса ::= {число с точкой}

ордината ::= {число с точкой}

$$\text{число с точкой} ::= \left\{ \begin{array}{l} \text{пробел} \\ \pm \end{array} \right\} \left\{ \text{целое без знака} . \text{целое без знака} \right\}$$

$$\text{целое без знака} ::= \left\{ \begin{array}{l} \text{цифра} \\ \text{цифра } \{ \text{целое без знака} \} \end{array} \right\}$$

В случае отсутствия синтаксических ошибок, вычисление площадей этих фигур и их покоординатное построение на экране.

### Вариант 13

Разработать программу с удобным пользовательским интерфейсом, реализующую следующие функции:

Синтаксический и семантический анализ понятия список геометрических фигур

$$СГФ ::= \left\{ \begin{array}{l} \text{геометрическая фигура} \\ \text{геометрическая фигура } \{ ; \text{список геометрических фигур} \} \end{array} \right\}$$

$$\text{геометрическая фигура} ::= \left\{ \begin{array}{l} \text{прямоугольник} \\ \text{трапеция} \end{array} \right\}$$

$$\text{прямоугольник} ::= \{ \text{вершина}; \text{вершина}; \text{вершина}; \text{вершина} \}$$

$$\text{трапеция} ::= \{ \text{вершина}; \text{вершина}; \text{вершина}; \text{вершина} \}$$

$$\text{вершина} ::= \{ \text{абсцисса}, \text{ордината} \}$$

$$\text{абсцисса} ::= \{ \text{число с точкой} \}$$

$$\text{ордината} ::= \{ \text{число с точкой} \}$$

$$\text{число с точкой} ::= \left\{ \begin{array}{l} \left\{ \begin{array}{l} \text{пробел} \\ \pm \end{array} \right\} \text{целое без знака} . \text{целое без знака} \end{array} \right\}$$

$$\text{целое без знака} ::= \left\{ \begin{array}{l} \text{цифра} \\ \text{цифра } \{ \text{целое без знака} \} \end{array} \right\}$$

В случае отсутствия синтаксических ошибок, вычисление площадей этих фигур и их покоординатное построение на экране.

### Вариант 14

Разработать программу с удобным пользовательским интерфейсом, реализующую следующие функции:

Синтаксический и семантический анализ понятия список геометрических фигур

$$СГФ ::= \left\{ \begin{array}{l} \text{геометрическая фигура} \\ \text{геометрическая фигура } \{ ; \text{список геометрических фигур} \} \end{array} \right\}$$

$$\text{геометрическая фигура} ::= \left\{ \begin{array}{l} \text{квадрат} \\ \text{ромб} \end{array} \right\}$$

$$\text{квадрат} ::= \{ \text{вершина}; \text{вершина}; \text{вершина}; \text{вершина} \}$$

$$\text{ромб} ::= \{ \text{вершина}; \text{вершина}; \text{вершина}; \text{вершина} \}$$

$$\text{вершина} ::= \{ \text{абсцисса}, \text{ордината} \}$$

$$\text{центр} ::= \{ \text{вершина} \}$$

$$\text{абсцисса} ::= \{ \text{число с точкой} \}$$

$$\text{ордината} ::= \{ \text{число с точкой} \}$$

$$\text{число с точкой} ::= \left\{ \left\{ \begin{array}{l} \text{пробел} \\ \pm \end{array} \right\} \text{целое без знака} . \text{целое без знака} \right\}$$

$$\text{целое без знака} ::= \left\{ \begin{array}{l} \text{цифра} \\ \text{цифра} \{ \text{целое без знака} \} \end{array} \right\}$$

В случае отсутствия синтаксических ошибок, вычисление площадей этих фигур и их покоординатное построение на экране.

### Вариант 15

Разработать программу с удобным пользовательским интерфейсом, реализующую следующие функции:

Синтаксический и семантический анализ понятия список геометрических фигур

$$СГФ ::= \left\{ \begin{array}{l} \text{геометрическая фигура} \\ \text{геометрическая фигура} \{ ; \text{список геометрических фигур} \} \end{array} \right\}$$

$$\text{геометрическая фигура} ::= \left\{ \begin{array}{l} \text{параллелограм} \\ \text{эллипс} \end{array} \right\}$$

параллелограмм ::= {вершина; вершина; вершина; вершина}

эллипс ::= {центр; радиус; радиус}

радиус ::= {число с точкой}

вершина ::= {абсцисса, ордината}

центр ::= {вершина}

абсцисса ::= {число с точкой}

ордината ::= {число с точкой}

$$\text{число с точкой} ::= \left\{ \left\{ \begin{array}{l} \text{пробел} \\ \pm \end{array} \right\} \text{целое без знака} . \text{целое без знака} \right\}$$

$$\text{целое без знака} ::= \left\{ \begin{array}{l} \text{цифра} \\ \text{цифра} \{ \text{целое без знака} \} \end{array} \right\}$$

В случае отсутствия синтаксических ошибок, вычисление площадей этих фигур и их покоординатное построение на экране.

### Вариант 16

Сеть Петри представляет собой двудольный граф с двумя типами вершин: кружочки (соответствуют условия) и палочки (соответствуют переходам). Если условие выполнено, то внутри соответствующего кружочка помещается маркер (точки или число, равное потенциалу условия). Дуги, соединяющие кружочки и палочки, задают входные условия переходов, а дуги, соединяющие палочки и кружочки – выходные условия переходов. Переход может сработать, если количество маркеров во всех его входных условиях не меньше, чем вес дуги, соединяющей это условие с данным

переходом. При срабатывании перехода из каждого входного условия изымается количество маркеров, равное весу дуги, соединяющей это условие со срабатываемым переходом, а в каждое выходное условие добавляется количество маркеров, равное весу дуги, соединяющей срабатываемый переход с этим условием. Сеть Петри функционирует переходя от одной разметки к другой.

Написать программу, моделирующую функционирование произвольной сети Петри.

### **Вариант 17**

Разработать программу для работы с БД которая обеспечивает работу склада:

а) Существует файл СКЛАД, в котором хранится информация об ассортименте продукции на складе:

- наименование продукции
- цена единицы продукции
- количество единиц наличия
- сумма

б) Имеется сеть магазинов, которые получают продукцию со склада. Информация храниться в файле РАСХОД:

- потребитель
- наименование продукции
- количество единиц отпущено
- сумма
- дата

Программа следит за движением продукции со склада, обеспечивает ввод, просмотр и удаление информации, каждому потребителю выдает счет-накладную.

### **Вариант 18**

Разработать программу для работы с БД, которая обеспечивает автоматизированную продажу авиабилетов на месяц вперед. Программа проверяет правильность каждой даты, делает возможным менять расписание. В типизированных файлах хранятся входные данные: номера рейсов, маршруты, количество свободных мест, дата вылета, ФИО.

При введении нового пункта расписания не разрешается вводить прошедшую дату (дата на данный момент времени высвечивается).

При покупке билета также проверяется дата и номер рейса. Не разрешается покупать билеты более, чем на месяц вперед.

### **Вариант 19**

Разработать программу для работы с БД слежения за прививками в детском саду.

Программа должна уметь:

отображать список детей младшей и старшей групп детского сада.

Должна присутствовать возможность зачисления и отчисления детей из детского сада.

Контролировать перевод из младшей группы в старшую, а также отчисление из старшей группы в зависимости от даты зачисления ребенка в детсад. (ребенок должен пробыть год в первой группе и год во второй).

Отображать данные о необходимости той, или иной прививки в зависимости от даты:

- Коклюш - раз в год;
- Палиомелит - раз в 6 месяцев;
- Желтуха - раз в 4 месяца;
- Туберкулез - раз в 3 месяца

Программа должна учитывать возможность отвода от прививок.

### **Вариант 20**

Разработать программу для работы с БД, которая регистрирует лиц, работающих по контракту, начисляет ежедневную заработную плату. Программа должна для каждого служащего выводить:

- Ф.И.О.
- Личный номер
- Количество отработанных дней
- Сумму заработной платы

Программа проверяет правильность каждой даты, делает возможным изменения Ф.И.О., даты начала и окончания контракта, заработной платы при введении личного номера рабочего.

В типизированном файле хранятся входные данные:

- Личный номер
- Ф.И.О.
- Даты начала и конца контракта
- Дневная заработная плата

При введении нового сотрудника дата начала контракта вводится по текущей дате на сегодняшний день, которую можно изменить.

### **Вариант 21**

Составить справочную телефонную систему. База данных содержит следующую информацию: фамилии абонентов телефонной системы, их номера телефонов и адрес (номер дома и название улицы). База данных может пополняться.

Уточненная постановка задачи.

В данной задаче нужно учесть несколько пунктов кроме указанных в условии:

- фамилии абонентов при выведении списка должны формироваться в алфавитном порядке;



- надо предусмотреть возможность исправления неправильного ввода Ф.И.О. и телефона;
- необходимо создать возможность легкого поиска абонентов;
- в справочнике нужно учесть возможность не только пополнять его, но и удалять абонентов из списка (проверку на удаление - подтверждение).

Для удобства обращения со справочником необходимо также предусмотреть возможность выбора операции (создать меню).

- обеспечить вывод личной карточки абонента со всей информацией. Задавать для поиска - по фамилии абонента в любом формате.
- по номеру абонента.
- по адресу (улица дом).
- (при этом предусмотреть выбор названий улиц из готового списка).

### **Вариант 22**

Написать программу которая выводит оценку за контрольную работу, которая выполняет путем выбора номеров ответов из представленной совокупности вариантов возможных ответов.

Имеется несколько видов контрольных работ. Информация, необходимая для оценки каждой контрольной работы:

наименование предмета

число вопросов в контрольной работе

последовательность цифр от 1 до 5. Количество цифр равно числу вопросов

Цифры представляют собой номер правильных ответов.

Сведения об ответах студентов содержатся в файле.

- личный номер
- фамилия (она запрашивается просто так и нигде не учитывается, поэтому под одним личным номером могут быть несколько разных фамилий)
- наименование предмета
- ответы, выбранные студентом

Программа должна выводить количество правильных ответов для каждого студента, оценку по каждому предмету (в пяти бальной системе)

Отл.5 – если от 0 до 10% неправильных ответов

Хор. 4 – от 11 до 20%

Уд.3 – от 21 до 30%

- Предусмотреть поиск информации по фамилии студента.
- просмотр личной карточки.
- составить ведомость, содержащую информацию о всех неуспевающих студентах (средний бал ниже).

### **Вариант 23**

Составить программу контроля за своевременной оплатой электроэнергии жильцами одного дома. Оплата должна производиться ежемесячно до последнего числа месяца, в случае несвоевременной оплаты

начисляются пени в расчете 1,5% за каждый просроченный день от невыплаченной суммы. Кроме того программа должна иметь удобный интерфейс. Здесь имеется ввиду система меню для общения с пользователем, сочетающее в себе справочную информацию по каждому пункту.

#### **Вариант 24**

Составить программу для игры в карточную игру «21».

Игра заключается в том, что побеждает тот, кто набирает 21 очко. Если и у игрока, и у компьютера меньше 21, то выигрывает тот, у кого очков больше. Если набираешь больше 21, то сразу проигрываешь.

Очки:

6 - 6оч,

7 - 7оч,

8 - 8оч,

9 - 9оч,

10 - 10оч,

J - 2оч,

Q - 3оч,

K - 4оч,

A - 11оч.

#### **Вариант 25**

Составить игровую программу «Морской бой», чтобы в ней могли принимать участие два игрока (компьютер и человек).

Каждый должен поставить свои корабли на игровом поле (один четырех палубный корабль, два - трехпалубных, три - двухпалубных, четыре - однопалубных), причем корабли должны ставиться как в вертикальном положении, так и горизонтальном. Игрок должен сам выбить место для постановки своих кораблей с помощью клавиш, а компьютер определяет координаты с помощью случайных чисел. Случайные числа должны лежать в диапазоне от 1 до 10. После расстановки кораблей начинается сама игра. Правила игры: компьютер и игрок поочередно стреляют по полю противника, подбивая корабли. Победит тот, кто первым потопит все вражеские корабли. Игрок делает выстрелы так: с помощью клавиш ←, ↑, →, ↓, управляя прицелом (квадратом красного цвета), выбирает место для выстрела и нажимает клавишу ENTER. Если игрок попадает, он получает право на дополнительный выстрел. Если игрок промахивается, то ход переходит к компьютеру. Компьютер стреляет в случайное место, но если он уже стрелял в это место, то выстрел будет сделан по другой клетке. При попадании, компьютер начинает стрелять вокруг этого места до тех пор, пока не добьет корабль. Если компьютер промахнулся – ход переходит к игроку.

#### **Вариант 26**

Руководство отеля решило автоматизировать обслуживание предварительных заказов на места в своем отеле. Написать программу соответствующей автоматизации системы. Отель имеет 14 комнат, пронумерованных 1-14. Каждая комната индивидуальна. Система резервирования мест обслуживает заказы на один год. Заказ принимается к обслуживанию только тогда, когда плата за комнатку уже внесена. Система диалога позволяет заказчику в процессе диалога конкретизировать параметры заказа. Заказчику предлагается ввести дату определяющую когда ему понадобятся услуги отеля и номер комнаты в которую он хотел бы вселиться.