

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное 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 2012

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 Примерные задания.....	34

Введение

Учебно-вычислительная практика студентов, обучающихся по специальности 230105.65 – программное обеспечение вычислительной техники и автоматизированных систем, 230100 – Информатика и вычислительная техника, 231000 – Программная инженерия является составной частью образовательной программы и представляет собой форму профессионально-практической подготовки студентов на базе практики.

Практика имеет целью закрепление полученных на первом курсе теоретических знаний и практических умений. Во время практики студенты разрабатывают приложения на языках высокого уровня, таких как C/C++, Pascal.

Для успешного выполнения учебной практики студент должен уметь работать в операционной системе, уметь пользоваться библиотеками Турбо Паскаля, C/C++, владеть знаниями особенностей алгоритмического языка в области модульного программирования, обработки сложных структур данных (файлы, структуры).

Учебная практика базируется на курсах «Информатика», «Программирование на языке высокого уровня» и проводится во 2-м семестре 1-го курса по завершению указанных предметов.

В качестве базы практики используются лаборатории университета, кафедры.

В методических указаниях представлены цель и 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 закрепление приобретенных 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 (*потоки 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 “Руководство оператора” по каждой сконструированной программе. Практика завершается защитой отчета.

В ходе учебной практики студент должен выполнить одно комплексное задание или несколько отдельных задач по темам:

- 1 Нерекурсивные алгоритмы повышенной сложности.
- 2 Рекурсивные алгоритмы (синтаксические анализаторы, алгоритмы поиска с возвратом и т. д.).
- 3 Обработка данных структурного типа, работа с файлами.
- 4 Реализация математических алгоритмов.
- 5 Начала объектно-ориентированного программирования.

В таблице 1 приведен примерный план работы по учебно – вычислительной практике.

Таблица 1 - Примерный план работы

День недели	Дата	Вид работы	Отчет
ПН	--.---.---	Вводное занятие. Анализ задания. Словесное описание алгоритма решения	
ВТ	--.---.---	Описание данных и основных подпрограмм в языке программирования	
СР	--.---.---	Детальная разработка основных	

		подпрограмм и описание дополнительных подпрограмм	
ЧТ	--.---.---	Детальная разработка дополнительных подпрограмм	
ПТ	--.---.---	Доработка процедур	Отчет о проделанной работе за неделю
ПН	--.---.---	Разработка интерфейса	
ВТ	--.---.---	Начало работы с основной программой	
СР	--.---.---	Корректировка основной программы и завершение работы с ней	
ЧТ	--.---.---	Отладка всей программы	
ПТ	--.---.---	Разработка руководства оператора	Отчет о проделанной работе за неделю
ПН	--.---.---	Составление отчета	
ВТ	--.---.---	Составление отчета	
СР	--.---.---	Составление отчета	
ЧТ	--.---.---	Составление отчета	
ПТ	--.---.---	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. Программа должна быть разбита на модули.

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

Приведен примерный перечень разрабатываемых подпрограмм для комплексного задания по обработке и хранению структурных данных.

1 Наборы данных (согласно заданному варианту) оформить в виде записей (структур в Си). 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 практическую работу, третий – содержание, далее по пунктам:

- 1 Описание разработки программы
- 2 Словесное описание алгоритма
- 3 Спецификация глобальных констант и переменных
- 4 Спецификация функций
- 5 Руководство оператора

Список использованных источников

Приложения: тестирование программы, схемы алгоритмов, исходные тексты программ

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 государственное бюджетное образовательное учреждение
высшего профессионального образования

«Оренбургский государственный университет»

Факультет информационных технологий
Кафедра программного обеспечения вычислительной техники и
автоматизированных систем

ОТЧЁТ

по учебно-вычислительной практике
на базе кафедры ПОВТАС

ФГБОУОГУ 231000.62.9012. __ П

Руководитель от кафедры

Тишина Н.А.

Исполнитель

студент гр. 11ИВТ(б)-1

Лежнев С.В.

Оренбург 2012

Содержание

1	Обоснование выбора языка программирования.....	3
2	Описание разработки структуры программы.....	4
3	Словесный алгоритм	5
4	Спецификация переменных	6
5	Спецификация функций	8
6	Руководство оператора	9
6.1	Назначение и условия применения программы.....	9
6.2	Характеристика программы.....	9
6.3	Выполнение программы.....	10
6.4	Сообщения оператору.....	12
	Список использованных источников	13
	Приложение А Схемы алгоритмов	14
	Приложение Б Тестирование программы	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 обработка данных проводится 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 используется структурированный тип, поэтому он описывается 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

Таблица 1 – Спецификация переменных

Идентификатор	Тип	Описание
1	2	3
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

Таблица 2 – Спецификация функций

Название	Описание
1	2
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: m , f char Вспомогательная переменная Key int Номер клавиши Cr Int Номер текущего пункта меню Old Int Номер предыдущего пункта меню</p>
flush(void)	<p>Функция записи всей информации в файлы</p> <p>Локальные переменные: SKLAD FILE* Указатель на Файл SKLAD RASXOD FILE* Указатель на Файл RASXOD 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: x,y int Координаты курсора</p>
window1(int x1,int y1,int x2,int y2)	<p>Рисует на экране окно.</p> <p>Локальные переменные: I int Счетчик</p>
vvod(void)	<p>Функция ввода целых значений, которая возвращает только цифры.</p> <p>Локальные переменные: M,c,cif,d Int Вспомогательные переменные Key Int Номер введенной клавиши array[] Int Массив кодов клавиш цифр 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>

Продолжение таблицы 2

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. Локальные переменные: RASXOD FILE* Указатель на Файл RASXOD I int Счетчик 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. Локальные переменные: j int Счетчик
corect(int number)	Функция изменяет имя какой-либо товар на складе.
schet(int p1,int p2)	Выводит на экран счёт-накладную. Локальные переменные: d date Текущая дата
int buy(int number)	Покупка товара на складе. Количество продукции вводится с клавиатуры. Локальные переменные: m,f,kol Int Вспомогательные переменные Key int Номер нажатой клавиши cr Int Номер текущего пункта меню Old Int Номер предыдущего пункта меню
int seach(void)	Функция поиска товара Локальные переменные: Name[20] Char Имя искомого товара m Int Вспомогательная переменная j Int Счетчик
int see_tov(void)	Функция просмотра продукции на складе. Локальные переменные: m,f,kol,u,d,l.p int Вспомогательная переменная Cr int Номер текущего пункта меню old int Номер предыдущего пункта меню key Int Номер клавиши

Продолжение таблицы 2

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

6 Руководство оператора

6.1 Назначение и условия применения программы

Программа предназначена для слежения за движением продукции со склада, обеспечения ввода, удаления и просмотра информации, выдачи каждому потребителю счёта-накладной.

Для нормального функционирования прототипа необходимо следующее минимальное аппаратное обеспечение:

- микропроцессор – 500 МГц;
- оперативная память – 128 Мб;
- свободное место на диске для программного комплекса 4 Мб и около 30 Мб свободного места для возможности наполнения базы данными;
- тип монитора не ниже VGA;
- клавиатура;
- мышь;
- принтер.

Минимальные требования к программному обеспечению:

- операционная система – не ниже Windows 98;
- среда программирования Borland C++ 3.1.

Для работы базы данных необходимо запустить prod.exe. Для этого необходимо зайти в папку C:\pract\prod.exe. После этого сервер запущен, и можно приступать к работе с прототипом.

6.2 Характеристика программы

Программа работает в консольном режиме. Для эксплуатации необходимо не менее 128 Мб оперативной памяти. Для проверки работоспособности программы необходимо запустить ее и проверить все режимы работы. Проверка программы должна начинаться с ее запуска. При запуске должно появиться главное окно с меню. Далее необходимо проверить правильно ли функционируют функции обработки данных. Для этого их поочередно выбирают в главном меню программы.

6.3 Выполнение программы

Программа использует для хранения информации динамическую память компьютера. При запуске программы появляется главное меню программы состоящее из 2 пунктов:

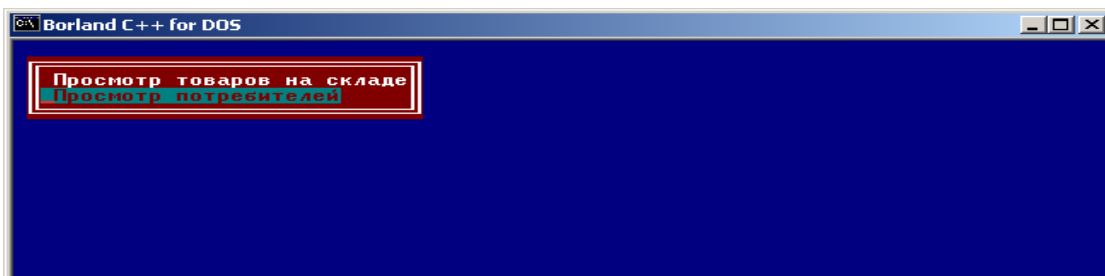


Рисунок 1 – Главное меню

- 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
moloko	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

Рисунок 2 – Просмотр товара

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

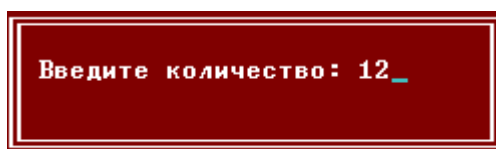


Рисунок 3 – Добавление количества товара на складе

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

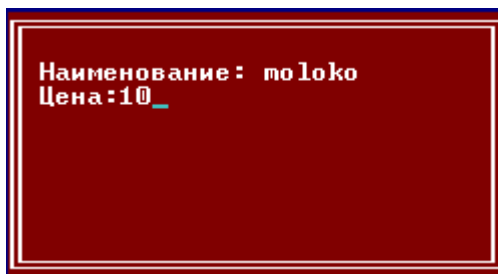


Рисунок 4 – Редактирование выбранного товара

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

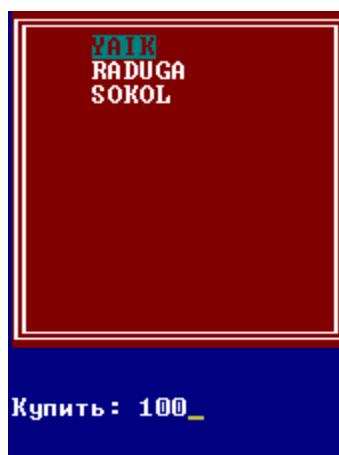


Рисунок 5 – Покупка товара

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

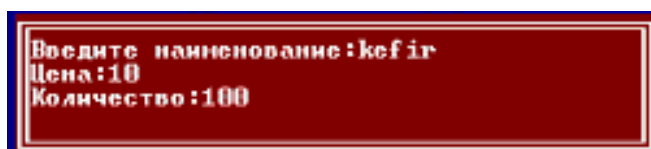


Рисунок 6 – Добавление товара на склад

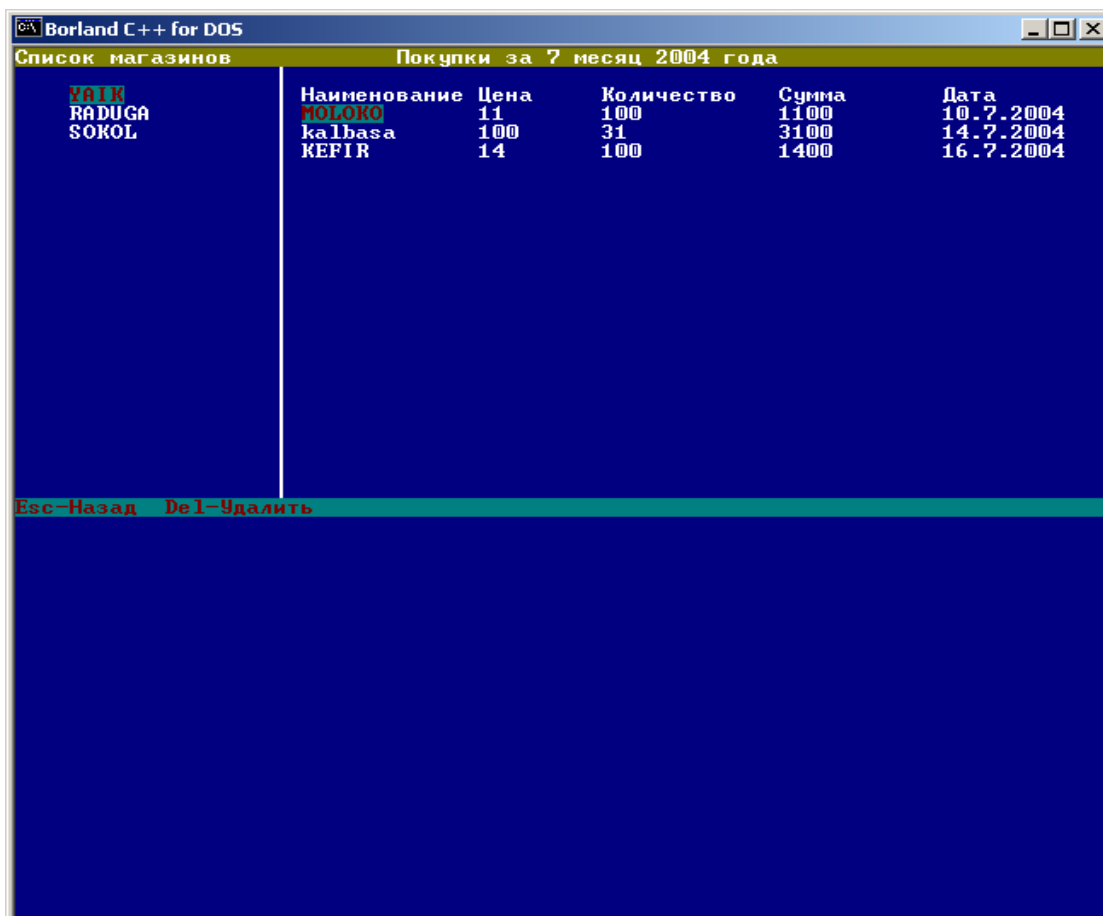
При нажатии клавиши «5» происходит поиск товара на складе.



Рисунок 7 – Поиск товара

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				
YAIK	Наименование	Цена	Количество	Сумма	Дата
RADUGA	MOLOKO	11	100	1100	10.7.2004
SOKOL	kalbasa	100	31	3100	14.7.2004
	KEPIR	14	100	1400	16.7.2004

Esc-Назад De1-Удалить

Рисунок 8 – 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, при нажатии клавиши «2» происходит удаление выбранного потребителя.

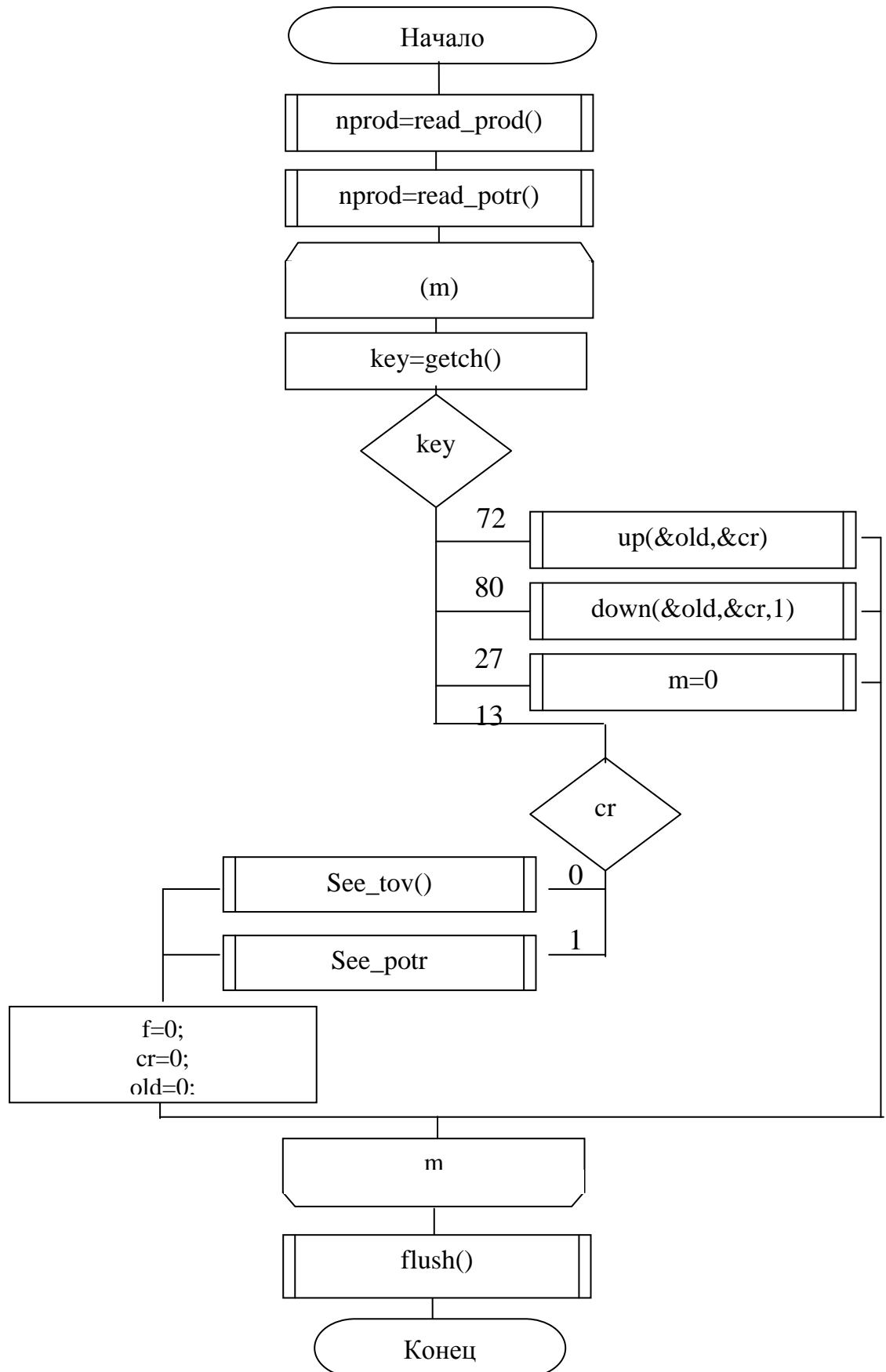
6.4 Сообщения оператору

Проверка возможности неправильного ввода данных...

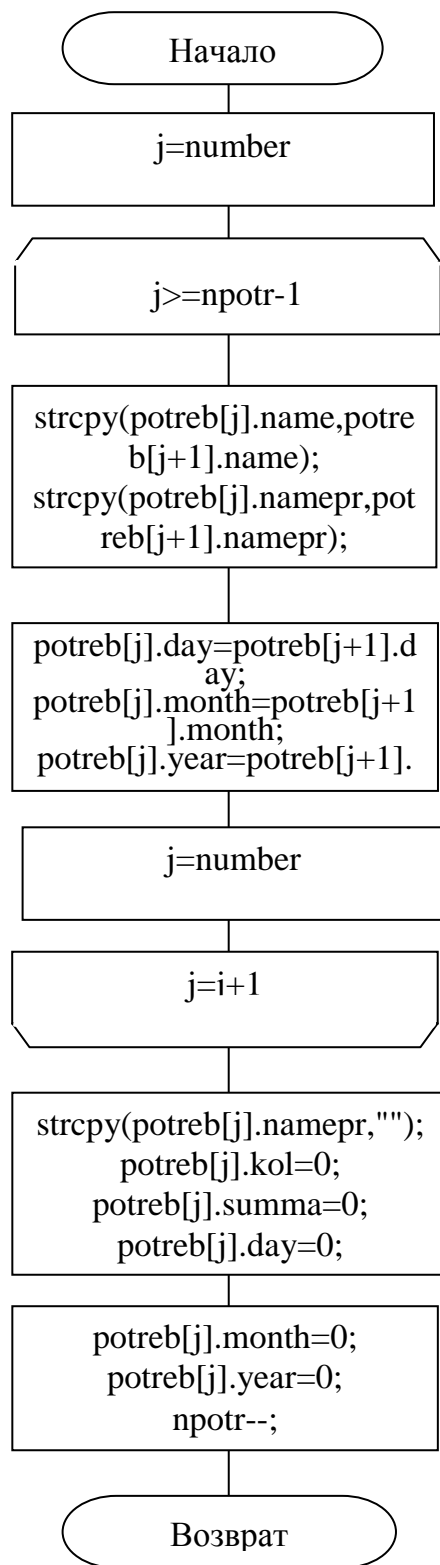
Список использованных источников

- 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.

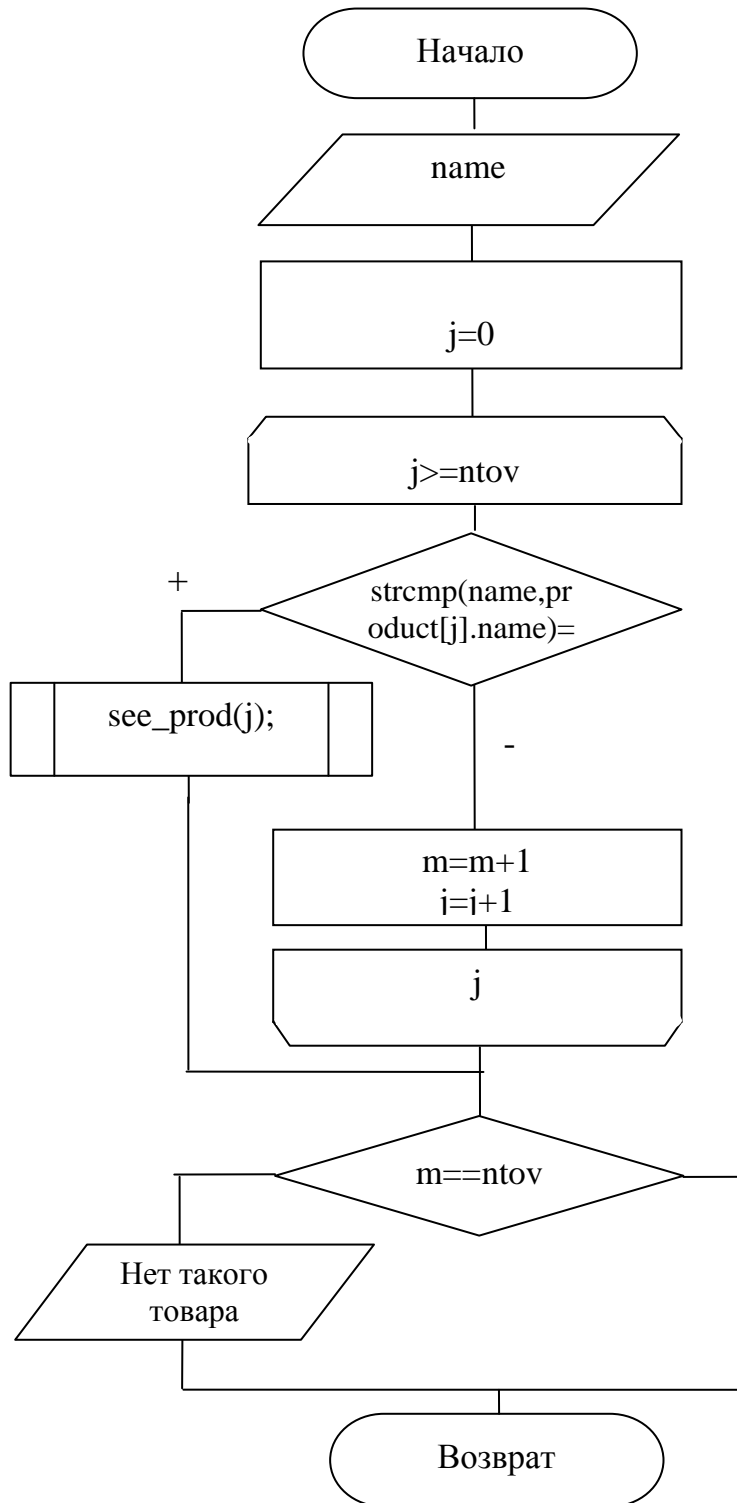
Приложение А
Схемы алгоритмов
Функция main()



Функция del_potr(int number)



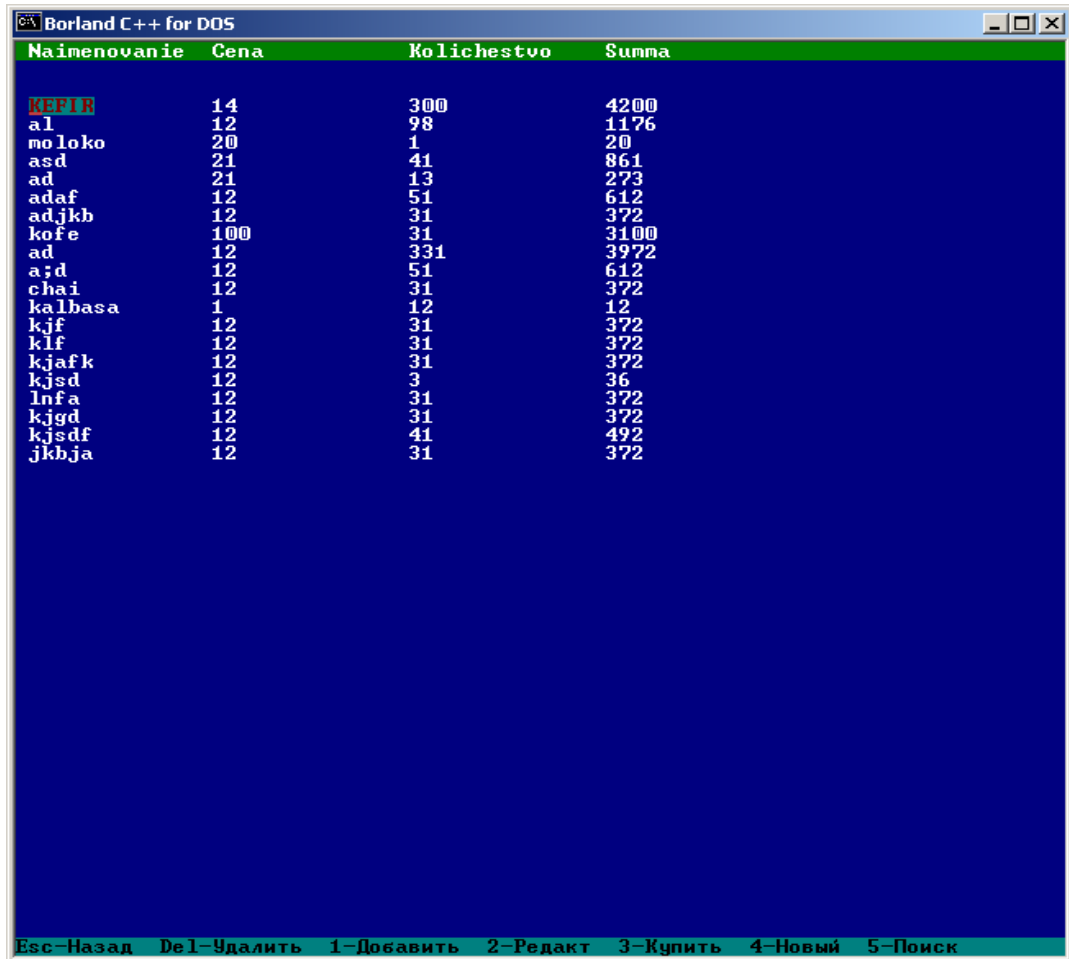
Функция seach(void)



Приложение Б

Тестирование программы

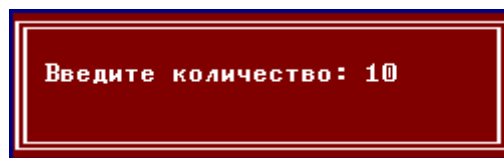
Пример просмотра товаров



Naimenovanie	Cena	Kolichestvo	Summa
KEPIR	14	300	4200
al	12	98	1176
molo ko	20	1	20
asd	21	41	861
ad	21	13	273
adaf	12	51	612
ad,jkb	12	31	372
kofe	100	31	3100
ad	12	331	3972
a;d	12	51	612
chai	12	31	372
kalbasa	1	12	12
kjf	12	31	372
klf	12	31	372
kjafk	12	31	372
kjsd	12	3	36
lnfa	12	31	372
kjgd	12	31	372
kjsdf	12	41	492
jkbja	12	31	372

Рисунок 9 – Просмотр товаров

Окно добавления количества товара



Введите количество: 10

Рисунок 10 – Добавление товара

Окно редактирования товара

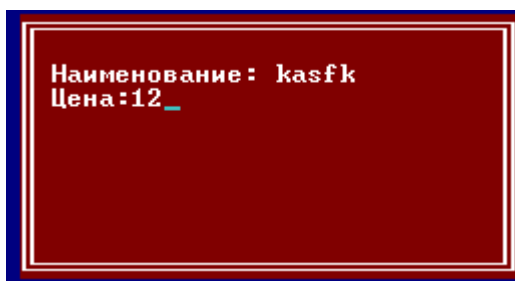


Рисунок 11 – Редактирование товара

Окно нового товара

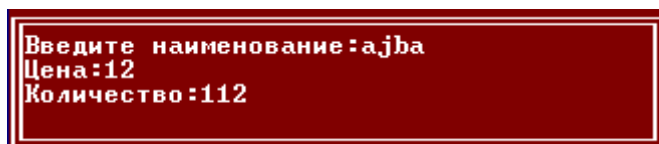


Рисунок 12 – Добавление нового товара

Покупка товара

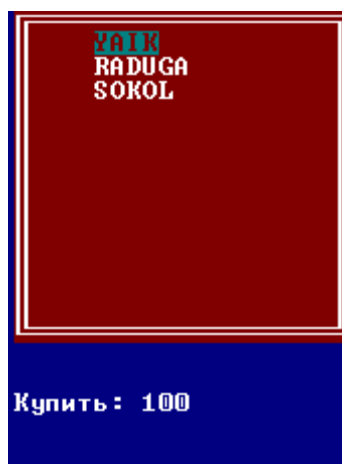


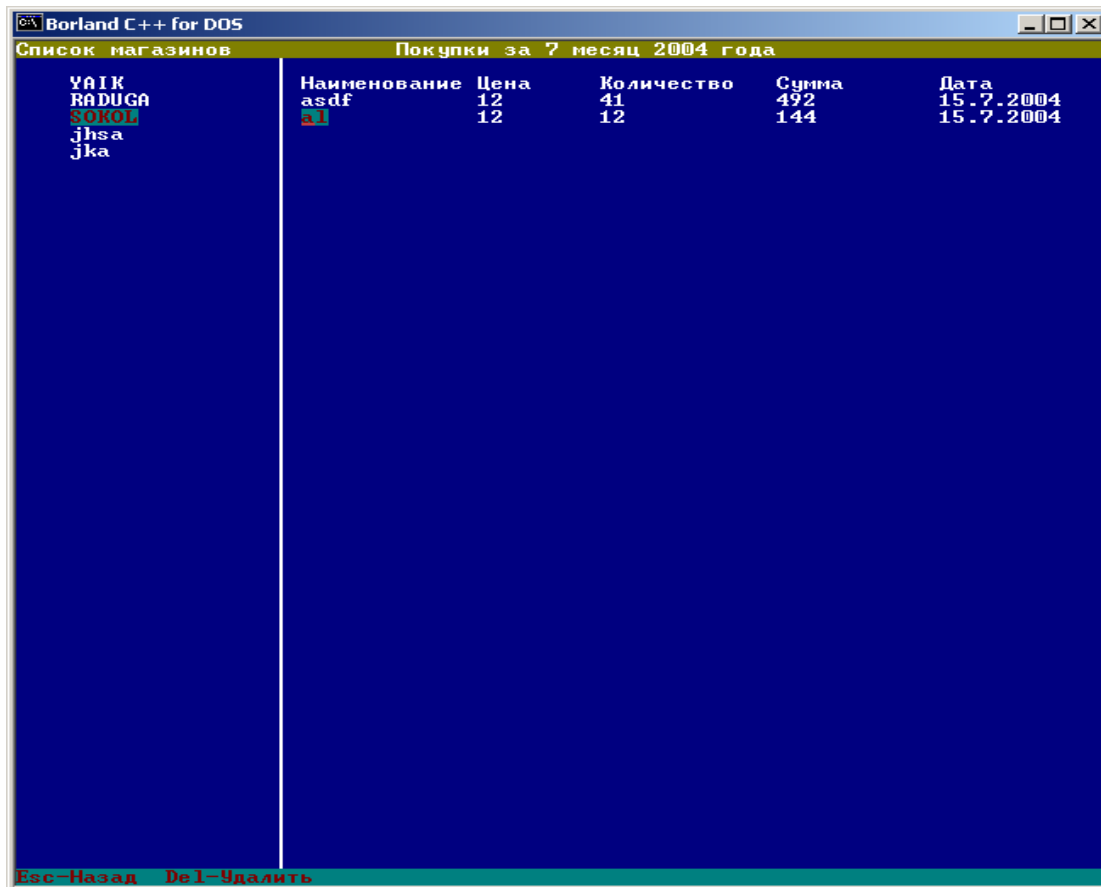
Рисунок 13 – Покупка товара

Окно поиска товара



Рисунок 14 – Поиск товара

Пример просмотра потребителей



Список магазинов

Покупки за 7 месяц 2004 года

Магазин	Наименование	Цена	Количество	Сумма	Дата
YA IK	asdf	12	41	492	15.7.2004
RADUGA	sd	12	12	144	15.7.2004
SOIKOI					
jhsa					
jka					

Esc-Назад Del-Удалить

Рисунок 15 – Просмотр потребителей

Приложение В

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",
                %i",potreb[i].prod[j].name,potreb[i].prod[j].price,potreb[i].prod[j].kol,potreb[i].prod[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;
window(2,2,30,5);
textbackground(color);
textcolor(15);
strcpy(strmenu[0],"ЦаⓂба вŸа - Ⓜ« Ⓜ");
strcpy(strmenu[1],"ЦаⓂба ĩⓂаŸĔĔĔ«Ĕ");
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);
printf("%s",strmenu[old]);
gotoxy(x,y);
textbackground(3);
textcolor(4);
printf("%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);
*****далее следует продолжение программы*****

```


Приложение Б *(обязательное)* **Примерные задания**

Задания на обработку и хранение структурных данных

Задача №1

Постановка задачи.

Составить программу простейшего автоматизированного англо - немецко-русского словаря, данные словаря хранятся в файле. Программа должна выполнять следующие задачи:

- 1 Добавление новых слов и редактирование имеющихся слов в словаре.
- 2 По запросу пользователя словарем выдавать перевод слова с любого языка на два других.
- 3 Производить поиск нужного слова (по полному и частичному совпадению).
- 4 Сортировку слов.

Задача №2

Постановка задачи.

Придумать контрольно - обучающую программу по изучению массивов. Программа должна содержать теоретическую информацию и контрольный блок, представленный тестами. Выводить оценку студента по завершению работы. Составлять ведомость студентов, Ф.И.О. и оценку, сохранять результаты в файле, осуществлять поиск.

Задача №3

Постановка задачи.

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.

Задача №4

Постановка задачи.

Написать программу, которая обеспечивает автоматизированную продажу авиабилетов на месяц вперед. Программа проверяет правильность каждой даты, делает возможным менять расписание. В файлах (raspisan , client) 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.

Задача №5

Постановка задачи.

Составить программу слежения за прививками в детском саду.

Программа должна уметь:

1) отображать список детей младшей и старшей групп детского сада.

2) Должна присутствовать возможность зачисления и отчисления детей из детского сада.

3) 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).

4) 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

5) программа должна учитывать возможность отвода от прививок.

Задача №6

Постановка задачи

Составить программу, 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 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.

Задача №7

Создать приложение «Картотека в бюро обмена квартир» организована как линейный список.

Сведения о каждой квартире содержат:

1. количество комнат;
2. этаж;
3. площадь;
4. адрес;

Составить программу, которая обеспечивает:

1. начальное формирование картотеки;
2. ввод заявки на обмен;
3. поиск в картотеке подходящего варианта: при равенстве количества комнат и этажа и различии площадей в пределах 10 % выводится соответствующая карточка, которая затем удаляется из списка; в противном случае поступившая заявка включается в список;
4. вывод всего списка.
5. Сохранение данных в файле по окончании работы программы

Определить тенденцию роста размена (зависимость количества обмена от года) и сделать прогноз на ближайшие два года.

Задача №8

Создать приложение «Администратор гостиницы». Список номеров: класс, число мест. Список гостей: паспортные данные, даты приезда и отъезда, номер. Поселение гостей: выбор подходящего номера (при наличии свободных мест), регистрация, оформление квитанции. Отъезд: выбор всех постояльцев, отъезжающих сегодня, возможность досрочного отъезда с перерасчетом. Поиск гостя по произвольному признаку.

Выяснить количество гостей иностранцев, определить десятку самых популярных номеров, определить самый доходный год и т. д. Дать

графическую интерпретацию полученным результатам в виде графиков, круговых диаграмм и гистограмм.

Определить тенденцию роста дохода (зависимость суммарной оплаты от года) и сделать прогноз на ближайшие два года.

Задача №9

Разработать программу с удобным пользовательским интерфейсом, реализующую следующие функции:

1. Синтаксический и семантический анализ понятия список геометрических фигур

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

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

прямоугольник ::= {вершина; вершина; вершина; вершина}

трапеция ::= {вершина; вершина; вершина; вершина}

вершина ::= {абсцисса, ордината}

абсцисса ::= {число с точкой}

ордината ::= {число с точкой}

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

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

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

Задания на обработку динамических структур данных

Задача №10

Написать программу, использующую стек для моделирования T – образного сортировочного узла на железной дороге. Программа должна разделять на два направления состав, состоящий из вагонов двух типов (на

каждое направление формируется состав из вагонов одного типа). Предусмотреть возможность формирования состава из файла и с клавиатуры.

Предусмотреть графическую интерпретацию.

Задача №11

Разработать приложение «Предметный указатель», в котором описан класс «Предметный указатель». Каждая компонента указателя содержит слово и номера страниц, на которых это слово встречается. Допустимое количество номеров страниц, относящихся к одному слову, от одного до десяти. Предусмотреть следующие возможности:

- 1 формирование указателя с клавиатуры и из файла,
- 2 вывод указателя,
- 3 вывод номеров страниц для заданного слова,
- 4 удаление элемента из указателя.
- 5 предусмотрена статистика слов, ее графическое изображение

Задача № 12

Написать игру «Пулемет».

Правила игры. Пулемет находится в нижней части экрана, может перемещаться влево и вправо с помощью клавиш управления курсором и стрелять. Слева на право по экрану перемещаются различные объекты. При попадании их пулемета объект взрывается.

Разработать *иерархию классов* для представления пулемета и различных (два-три) типов объектов. Предусмотреть хранение и редактирование данных о результатах игры.

Задача №13

Создать приложение «Генеалогическое дерево». Паспортные данные членов некоторого родового клана; ссылки на детей (или на родителей). Поиск всех потомков или всех предков для указанного лица, возможность редактирования, добавления, удаления. Графическое изображение дерева.

Задания на реализацию математических методов

Задача №14

Написать программу, позволяющую вычислять площади криволинейных трапеций несколькими способами. Предусмотреть защиту от некорректного ввода исходных данных, реализовать графическую интерпретацию решения задачи, сравнить результаты расчета различными методами.

Задача 15

Написать программу, позволяющую вычислять корни нелинейных уравнений несколькими способами. Предусмотреть защиту от некорректного ввода исходных данных, реализовать графическую интерпретацию решения задачи, сравнить результаты расчета различными методами.

Задача 16

Написать программу, реализующую решение системы линейных алгебраических уравнений несколькими способами. Организовать ввод данных с клавиатуры и из файла. Предусмотреть защиту от некорректного ввода исходных данных. Придумать тестовые примеры, сравнить результаты расчета различными методами.

Задача 17

Написать программу, имитирующую перевод чисел в различные системы счисления и выполнение арифметических операций над ними. Разработать удобную систему подсказок пользователю и графический интерфейс.

Задача 18

Программа должна генерировать непрерывный поток случайных чисел, т.е. задавать значения некоторой случайной величины X . Каждые t секунд определять подчиняется ли случайная величина равномерному или нормальному закону распределения.

Задания на обработку файлов

Задача 19

Написать программу, имитирующую работу файлового менеджера компьютера: создание, удаление файлов и каталогов и т.д. Реализовать графический пользовательский интерфейс и возможность расширения программы.

Задача 20

Дан текстовый файл большого размера произвольной структуры. Реализовать различные способы чтения файла: посимвольно, поблочно, построчно с использованием разных функций чтения: низкоуровневые, потоковые. Организовать поиск заданного слова в тексте при полном и частичном совпадении, посчитать количество встречаемости слова в тексте. Определить скорость работы каждого способа.

Задача №21

Написать программу обработки журнала файервола (Windows NT, iptables) или журнала программы типа сетевой сканер TCPdump. Программа должна считывать содержимое файла построчно, разбивать строку на

структурные элементы и записывать их в массив структур. Организовать поиск и сортировку по различным критериям. Посчитать статистику использования пакетов: TCP, UDP, ICMP. Посчитать среднее количество байт, пакетов на заданную единицу интервала. Дать графическую интерпретацию полученных результатов.