

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ  
НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ЭКОНОМИКИ И УПРАВЛЕНИЯ – «НИНХ»**

Кафедра информационных технологий

**МЕТОДИЧЕСКОЕ РУКОВОДСТВО  
ПО ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ  
СТУДЕНТОВ ОЧНОЙ ФОРМЫ ОБУЧЕНИЯ**

Учебная дисциплина  
**ПРОГРАММИРОВАНИЕ**

по направлению подготовки

**38.03.05 Бизнес-информатика**

профили **Электронный бизнес, Архитектура предприятий**

**Новосибирск 2016**

## ОГЛАВЛЕНИЕ

<b>РАЗДЕЛ 1. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПОДГОТОВКЕ К ПРАКТИЧЕСКИМ ЗАНЯТИЯМ.....</b>	<b>4</b>
1.1. ОРГАНИЗАЦИЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ ПО ПОДГОТОВКЕ К ПРАКТИЧЕСКИМ ЗАНЯТИЯМ.....	4
1.2. СОДЕРЖАНИЕ ПРАКТИЧЕСКИХ ЗАНЯТИЙ.....	4
<b>РАЗДЕЛ 2. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ИЗУЧЕНИЮ ДИСЦИПЛИНЫ.....</b>	<b>9</b>
2.1. ТЕМЫ, ПОДЛЕЖАЩИЕ САМОСТОЯТЕЛЬНОМУ ИЗУЧЕНИЮ СТУДЕНТАМИ .....	9
2.2. ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ, НЕОБХОДИМЫЕ СТУДЕНТУ ДЛЯ УСВОЕНИЯ ТЕМ, ПОДЛЕЖАЩИХ САМОСТОЯТЕЛЬНОМУ ИЗУЧЕНИЮ.....	9
2.3. МАТЕРИАЛЫ ДЛЯ САМОПРОВЕРКИ ЗНАНИЙ .....	9
<b>РАЗДЕЛ 3. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПОДГОТОВКЕ К ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ .....</b>	<b>10</b>
3.1. РЕКОМЕНДАЦИИ ПО ПОДГОТОВКЕ К ЭКЗАМЕНАМ В 1-ОМ И 2-ОМ СЕМЕСТРАХ .....	10
3.2. СПИСОК ВОПРОСОВ ДЛЯ ПОДГОТОВКИ К ЭКЗАМЕНУ В 1 СЕМЕСТРЕ.....	10
3.3. ПРИМЕР ЭКЗАМЕНАЦИОННОГО БИЛЕТА ПО ИТОГАМ 1 СЕМЕСТРА .....	12
3.4. СПИСОК ВОПРОСОВ ДЛЯ ПОДГОТОВКИ К ЭКЗАМЕНУ ВО 2 СЕМЕСТРЕ .....	13
3.5. ПРИМЕР ЭКЗАМЕНАЦИОННОГО БИЛЕТА ПО ИТОГАМ 2 СЕМЕСТРА .....	14
3.6. ОБЩИЕ ПОЛОЖЕНИЯ ПРОВЕДЕНИЯ ЭКЗАМЕНА И КРИТЕРИИ ОЦЕНКИ .....	15
<b>РАЗДЕЛ 4. ЗАДАНИЯ ПО ВЫПОЛНЕНИЮ РГР .....</b>	<b>16</b>
4.1. ВЫБОР ВАРИАНТА РГР .....	16
4.2. ТЕМЫ РГР СОГЛАСНО ВАРИАНТАМ.....	16
<b>РАЗДЕЛ 5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ КУРСОВОГО ПРОЕКТА .....</b>	<b>17</b>
5.1. ЦЕЛЬ И ЗАДАЧИ КУРСОВОГО ПРОЕКТА.....	17
5.2. ЭТАПЫ ВЫПОЛНЕНИЯ КУРСОВОГО ПРОЕКТА.....	17
5.3. ВЫБОР ТЕМЫ КУРСОВОГО ПРОЕКТА .....	17
5.4. ВАРИАНТЫ ТЕМ КУРСОВОГО ПРОЕКТА .....	18
5.5. ПРАВИЛА ОФОРМЛЕНИЯ ТЕКСТА КУРСОВОГО ПРОЕКТА .....	23
5.6. ПРИМЕР ПРОГРАММНОГО КОДА, ПОЛУЧЕННОГО В РЕЗУЛЬТАТЕ РАЗРАБОТКИ ПРИЛОЖЕНИЯ СОГЛАСНО ТЕМЕ КУРСОВОГО ПРОЕКТА .....	24
5.7. ПРИМЕР ПРОГРАММНОГО КОДА, ПОЛУЧЕННОГО В РЕЗУЛЬТАТЕ РАЗРАБОТКИ ПРИЛОЖЕНИЯ СОГЛАСНО ТЕМЕ КУРСОВОГО ПРОЕКТА .....	24
5.8. ПРИМЕРЫ ОФОРМЛЕНИЯ СКРИНШОТОВ ПРОГРАММЫ .....	29
5.9. КРИТЕРИИ ОЦЕНИВАНИЯ .....	32
5.10. ОФОРМЛЕНИЕ ТИТУЛЬНОГО ЛИСТА.....	33

## **РАЗДЕЛ 1. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПОДГОТОВКЕ К ПРАКТИЧЕСКИМ ЗАНЯТИЯМ**

### **1.1. Организация самостоятельной работы студентов по подготовке к практическим занятиям**

Для подготовки к практическим занятиям студенту рекомендуется использовать следующие материалы:

- тексты лекций (файл приложен);
- задачи для самоподготовки (файл приложен);
- источники из списка литературы.

Студенту рекомендуется следующая схема подготовки к практическому занятию по учебной дисциплине «Программирование»:

- прочитать очередную тему в текстах лекций;
- решить 5-10 задач по соответствующей теме из файла с задачами для самоподготовки;
- ознакомиться с дополнительной информацией из рекомендуемой литературы.

### **1.2. Содержание практических занятий**

#### **Раздел 1. Основы программирования**

##### **Тема 1.1. Введение в программирование**

Понятие алгоритма, блок-схемы, псевдокода, программного кода. Компиляция и отладка программы. Синтаксические и алгоритмические (логические) ошибки, ошибки времени исполнения. Структура программы. Функция `main()`. Средства ввода/вывода. Переменные и типы данных. Объявление переменной. Ключевые слова и идентификаторы. Арифметические операции. Остаток от деления. Деление нацело. Подключение библиотек (заголовочных файлов).

##### **Тема 1.2. Математические вычисления**

Использование математических функций. Приведение типов. Разработка программ с использованием тригонометрических функций. Квадратный корень. Возведение в степень. Вычисление логарифмов. Вычисление характеристик

геометрических объектов. Сокращённые формы арифметических операций. Особенности работы оператора деления с целыми и вещественными числами. Обращение к отдельным цифрам числа с помощью операций деления нацело и остатка от деления.

### **Тема 1.3. Условный оператор if-else**

Понятие условного оператора. Синтаксис оператора if-else. Операции сравнения. Логические операции и выражения. Таблицы истинности логических операций. Примеры использования оператора if-else. Использование оператора if без ветки else. Генерация псевдослучайных чисел и игра «Угадай число!». Вложенные операторы if-else и «лесенка» if-else-if. Интеграция математических вычислений и условных операторов. Глобальные и локальные переменные относительно тела оператора if-else. Сокращённая форма оператора if-else. Решение квадратного уравнения.

### **Тема 1.4. Основы работы со строками**

Объявление строки и символа. Нумерация символов строки. Обращение к отдельным символам строки. Использование кавычек и апострофов применительно к строкам и символам. Определение длины строки. Конкатенация строк. Обработка строк с использованием условного оператора if-else. Ввод строк, состоящих из нескольких слов, при помощи функции getline(). Использование функций setlocale(), SetConsoleCP() и SetConsoleOutputCP () для отображения русских символов.

### **Тема 1.5. Условный оператор switch-case**

Синтаксис оператора switch-case. Отличия операторов if-else и switch-case. Роль оператора break в конструкции switch-case. Примеры осознанного исключения операторов break из конструкции switch-case. Реализация консольного меню при помощи оператора switch-case. Метка default.

## **Раздел 2. Циклы и массивы**

### **Тема 2.1. Цикл на базе операторов if и goto**

Понятие цикла и итерации. Задачи, решаемые при помощи циклов. Основные причины использования циклов: гибкость и исключение дублирования

программного кода. Оператор goto. Цикл на базе операторов if и goto. Переменная-счётчик.

### **Тема 2.2. Циклы while и do-while**

Циклы while и do-while. Их связь с циклом на базе операторов if и goto. Решение вычислительных задач с использованием циклов. Использование цикла do-while, когда должна быть выполнена хотя бы одна итерация. Бесконечный цикл. Прерывание цикла при помощи оператора break. Неправильное условие как причина заикливания. Глобальные и локальные переменные относительно тела цикла.

### **Тема 2.3. Цикл for**

Цикл for. Использование цикла for, когда число итераций известно. Правила выбора цикла в зависимости от решаемой задачи. Вычисление сумм рядов с использованием циклов. Вывод последовательностей. Генерация псевдослучайных чисел. Линейный конгруэнтный генератор псевдослучайных чисел. Числа Фибоначчи. Факториал. Генерация арифметической и геометрической прогрессий. Вложение условных операторов в циклы.

### **Тема 2.4. Массивы**

Понятие массива. Объявление массива. Нумерация элементов массива. Индекс элемента массива. Заполнение массива. Сортировка массива. Поиск максимального и минимального элемента. Вычисление суммы элементов массива. Поиск элемента в массиве. Последовательный поиск. Интеграция массивов, циклов и условий. Подсчёт числа элементов с нужным свойством. Работа с массивом заранее не известного размера. Пузырьковая сортировка. Сортировка методом прямого выбора.

### **Тема 2.5. Обработка строк с использованием циклов**

Интеграция строк, циклов и условий. Подсчет числа символов строки с нужным свойством. Примеры решения задач. Задача поиска подстроки в строке. Наивный алгоритм.

## **Раздел 3. Процедуры и функции**

### **Тема 3.1. Процедуры**

Понятие подпрограммы, процедуры и функции. Объявление, определение и вызов процедуры (функции). Отличие процедуры и функции. Сигнатура (интерфейс) процедуры. Пустой тип – void. Выбор имени процедуры. Формальные и фактические аргументы. Тело процедуры. Правила выбора имён для процедур. Глобальные и локальные переменные относительно некоторой процедуры.

### **Тема 3.2. Функции**

Объявление, определение и вызов функции. Тип возвращаемого функцией значения. Оператор return. Сигнатура (интерфейс) функции. Создание и подключение заголовочного файла. Создание и вызов функций для решения различных вычислительных задач. Правила выбора имён для функций. Передача значений различных типов в качестве аргументов функций.

### **Тема 3.3. Рекурсивные функции и процедуры**

Примеры рекурсивных объектов. Рекуррентное соотношение. Понятие рекурсивной функции (процедуры). Структура рекурсивной функции (процедуры). Ошибочное проектирование рекурсивных функций как причина заикливание. Факториал. Числа Фибоначчи. Итеративные и рекурсивные версии функций (процедур). Связь рекурсии и цикла. Примеры известных рекурсивных алгоритмов. Вычисление числа Фибоначчи как пример неэффективного использования рекурсии.

### **Тема 3.4. Логические функции**

Логические функции как особый подвид функций. Логический тип bool. Объявление, определение и вызов логических функций. Особенности вызова логических функций. Использование логических функций при обработке массивов и строк. Правила выбора имён для логических функций. Флаг.

### **Тема 3.5. Вложенные циклы и понятие API**

Примеры задач, подразумевающих использование вложенных циклов. Вынос вложенного цикла в отдельную функцию (процедуру) как способ упрощения структуры программы. Понятие прикладного программного интерфейса

(Application programming interface - API). Пример создания и использования прикладного программного интерфейса.

## **Раздел 4. Введение в структуры данных**

### **Тема 4.1. Матрицы**

Примеры задач, при решении которых используются матрицы. Объявление матрицы. Заполнение матрицы. Способы заполнения матриц. Вывод матрицы на экран в виде таблицы. Столбцы и строки матрицы. Обработка матриц с использованием вложенных циклов.

### **Тема 4.2. Массивы строк**

Объявление и заполнение массива строк. Особенности работы с массивами строк. Сортировка массива строк по алфавиту (по длине строки). Поиск конкретных строк в массиве строк. Передача строк в качестве аргументов функций. Возвращение строки в качестве значения функции.

### **Тема 4.3. Файлы**

О необходимости использования файлов при решении некоторых задач. Файл как структура данных последовательного доступа. Запись данных в файл. Чтение данных из файла. Признак конца файла. Функция eof() и чтение файла целиком. Определение размера файла. Обработка файлов. Копирование файлов. Режимы открытия файлов. Работа с файлами в стиле языка C и языка C++.

### **Тема 4.4. Структуры**

Понятие структуры. Структура как новый составной тип данных. Переменные типа «структура». Поле структуры. Оператор «точка». Массив структур. Вложенные структуры. Моделирование объектов реального мира при помощи структур. Моделирование геометрических объектов при помощи структур. Структура как класс без методов. Передача структур в качестве аргументов функций.

### **Тема 4.5. Динамические структуры данных на основе массива.**

Понятие динамической структуры данных. Вставка и удаление элементов. Стек, очередь и список. Использование вспомогательной переменной для

хранения текущего количества элементов. Имитация удаления и добавления элементов.

## **РАЗДЕЛ 2. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ИЗУЧЕНИЮ ДИСЦИПЛИНЫ**

### **2.1. Темы, подлежащие самостоятельному изучению студентами**

Тема 1.1. Введение в программирование; Тема 1.2. Математические вычисления; Тема 1.3. Условный оператор if-else; Тема 1.4. Основы работы со строками; Тема 1.5. Условный оператор switch-case;

Тема 2.1. Цикл на базе операторов if и goto; Тема 2.2. Циклы while и do-while; Тема 2.3. Цикл for; Тема 2.4. Массивы; Тема 2.5. Обработка строк с использованием циклов;

Тема 3.1. Процедуры; Тема 3.2. Функции; Тема 3.3. Рекурсивные функции и процедуры; Тема 3.4. Логические функции; Тема 3.5. Вложенные циклы и понятие API;

Тема 4.1. Матрицы; Тема 4.2. Массивы строк; Тема 4.3. Файлы; Тема 4.4. Структуры; Тема 4.5. Динамические структуры данных на основе массива.

### **2.2. Основные понятия и определения, необходимые студенту для усвоения тем, подлежащих самостоятельному изучению**

Список основных терминов и определений к ним приведен в файле с текстами лекций в разделе «Глоссарий» (с. 160).

### **2.3. Материалы для самопроверки знаний**

Для самопроверки знаний рекомендуется использовать:

- вопросы для самоконтроля (в конце каждой лекции в файле с текстами лекций);
- случайно выбранные задачи для самоподготовки;
- раздел «Глоссарий» в файле с текстами лекций.



### РАЗДЕЛ 3. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПОДГОТОВКЕ К ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Промежуточная аттестация в обоих семестрах проходит в виде экзамена. Экзамен в 1-ом семестре проверяет материал из первых 10 тем (№№ 1.1 – 2.5), а во 2-ом – из заключительных 10 тем (№№ 3.1 – 4.5).

#### 3.1. Рекомендации по подготовке к экзаменам в 1-ом и 2-ом семестрах

Для подготовки к экзаменам рекомендуется использовать:

- тексты лекций (приложенный файл);
- задачи для самоподготовки (приложенный файл).

Рекомендуется ответить на вопросы для подготовки, пользуясь текстами лекций, решить по 5-10 задач из каждой темы и ознакомиться с примером экзаменационного билета по следующей схеме:

Номер семестра	Темы	Список вопросов	Пример билета
1	1.1, 1.2, 1.3, 1.4, 1.5, 2.1, 2.2, 2.3, 2.4., 2.5	Пункт 3.2	Пункт 3.3
2	3.1, 3.2, 3.3, 3.4, 3.5, 4.1, 4.2, 4.3, 4.4., 4.5	Пункт 3.4	Пункт 3.5

#### 3.2. Список вопросов для подготовки к экзамену в 1 семестре

1. Что такое интегрированная среда разработки, компилятор, блок-схема?
2. Как проявляются алгоритмические (логические), синтаксические и ошибки времени исполнения?
3. Как использование отступов в коде облегчает разработку программ?
4. Как выделяются однострочные и многострочные комментарии?
5. Перечислите основные типы данных языков C/C++.
6. Дайте определения ключевых слов и идентификаторов.
7. Как рекомендуется выбирать имена переменных?
8. Как объявлять целые, вещественные, символьные и строковые переменные?
9. Как вводить/выводить значения переменных?
10. Что такое операции и операнды? Что такое унарные и бинарные операции?

11. Приведите таблицы истинности логических операций.
12. Что такое приоритет операций?
13. Как выглядят сокращённые формы записи инкремента и декремента?
14. Перечислите существующие в языках C/C++ математические функции.
15. Приведите примеры использования условных операторов.
16. Как выглядит сокращённая запись оператора if-else?
17. Какие операции сравнения и логические операции существуют в языках C/C++?
18. Как можно реализовать консольное меню?
19. Приведите пример и изобразите блок-схему вложенных операторов if-else.
20. Приведите примеры использования «лесенки» if-else-if.
21. Изобразите блок-схему решения квадратного уравнения.
22. Объясните работу оператора goto.
23. Как реализовать цикл на базе операторов if и goto?
24. Дайте определение итерации.
25. Когда целесообразно использовать циклическую конструкцию?
26. Какие типы циклов существуют в языках C/C++?
27. Приведите правила выбора одного из циклов for, while и do-while.
28. Приведите примеры использования переменной-счётчика в циклах.
29. Как реализовать бесконечный цикл?
30. Объясните работу операторов break и continue. Приведите примеры.
31. Как вычислить сумму элементов последовательности, используя циклы?
32. Как объявить массив и заполнить его случайными числами?
33. Опишите алгоритм поиска максимального (минимального) элемента в массиве.
34. С какой целью обычно сортируются данные?
35. Опишите некоторые алгоритмы сортировки массива.
36. Опишите алгоритм последовательного поиска.
37. Как вывести на экран последовательность Фибоначчи?

38. По каким причинам цикл может зациклиться?
39. Сформулируйте постановку задачи поиска подстроки в строке.
40. Опишите алгоритм поиска подстроки в строке.
41. Как подсчитать число элементов с нужным свойством в массиве (строке)?
42. Изобразите блок-схему вычисления суммы цифр произвольного целого числа.

### 3.3. Пример экзаменационного билета по итогам 1 семестра

**Задача 1.** Некоторая научная организация выплачивает ученому премию в зависимости от числа публикаций: 20000 рублей (1-2 публикации); 40000 рублей (3-5 публикаций); 70000 рублей (6-10 публикаций); 100000 рублей (свыше 10 публикаций). Введите с клавиатуры число публикаций ученого и определите размер премии.

**Задача 2.** Напишите программу, которая принимает с клавиатуры число и проверяет, является ли оно числом из последовательности Фибоначчи. Например, для чисел 0, 13, 55 или 144 программа должна ответить «Да», а для чисел 6, 10, 80 или 200 – «Нет».

**Задача 3.** Напишите программу, которая принимает с клавиатуры число  $N$  и выводит на экран следующую последовательность (числа должны быть разделены запятой):

Возможные значения $N$	Соответствующие последовательности
0	0
1	0, 1, -1
2	0, 1, -1, 2, -2
3	0, 1, -1, 2, -2, 3, -3
4	0, 1, -1, 2, -2, 3, -3, 4, -4
5	0, 1, -1, 2, -2, 3, -3, 4, -4, 5, -5
...	...

**Задача 4.** Напишите программу, которая принимает с клавиатуры размер массива, заполняет его случайными числами из диапазона от 50 до 150 и выводит его на экран. Затем определите, сколько в этом массиве чётных чисел, и выведите их.

**Задача 5.** Напишите программу, которая принимает с клавиатуры строку и выводит на экран каждый второй символ этой строки. Например, если введена строка «1234567abcdef», то на экране должно появиться «1357bdf».

### **3.4. Список вопросов для подготовки к экзамену во 2 семестре**

1. Дайте определения процедуры (функции)?
2. Чем обусловлена необходимость использования процедур (функций)?
3. Из каких составных частей состоит определение функции?
4. Что такое аргументы функции?
5. Сколько аргументов может иметь функция?
6. Чем отличаются глобальные переменные от локальных?
7. Объясните работу оператора return.
8. Чем отличаются функции от процедур?
9. Что такое рекурсивная функция? Приведите примеры.
10. Объясните связь итеративных и рекурсивных функций.
11. Приведите примеры известных рекурсивных алгоритмов.
12. По каким причинам рекурсивная функция может зациклиться?
13. Что такое прототип (сигнатура, интерфейс) функции? Приведите примеры.
14. Что такое заголовочный файл и как с ним работать?
15. Чем следует руководствоваться при выборе имени для функции?
16. Приведите примеры «хороших» и «плохих» названий для функций.
17. Что такое формальные и фактические аргументы функции (процедуры)?
18. Какова связь между формальными и фактическими аргументами функции?
19. Что такое логические функции? Приведите примеры их использования.
20. Приведите примеры применения логических функций при обработке массивов.
21. Приведите примеры использования логических функций при обработке строк.
22. Объясните, что такое прикладной программный интерфейс?

23. Как объявить матрицу?
24. Как заполнить матрицу случайными числами?
25. Как вычислить сумму чисел в некотором столбце матрицы?
26. Как определить максимальный (минимальный) элемент в строке матрицы?
27. Как объявить массив строк?
28. Как отсортировать массив строк по алфавиту (по длине строки)?
29. Как записать данные в файл?
30. Как считать данные из файла?
31. Как прочитать содержимое файла целиком?
32. Как определить количество символов в файле?
33. Что такое структура?
34. Как обратиться к полю структуры?
35. Как объявить массив структур?
36. Приведите пример вложенных структур.
37. Приведите примеры моделирования объектов реального мира и геометрических объектов с помощью структур.
38. Приведите примеры передачи структур в качестве аргументов функций.
39. Дайте определение динамической структуры данных.
40. Назовите некоторые динамические структуры данных.
41. Как реализовать стек (очередь) на основе массива?

### **3.5. Пример экзаменационного билета по итогам 2 семестра**

**Задача 1.** Создайте структуру Point (Точка) с вещественными полями *x* и *y*, которые задают координаты этой точки в двумерном пространстве. Создайте структуру Triangle (Треугольник), состоящую из трёх полей типа Point, задающих координаты вершин этого треугольника. Создайте функцию, которая принимает аргумент типа Triangle и возвращает длину самой длинной его стороны.

**Задача 2.** Создайте процедуру, которая принимает в качестве аргумента строку и выводит её на экран столько раз, какова длина этой строки. Например, строка «сок» должна быть выведена 3 раза, а строка «барсук» – 6 раз.

**Задача 3.** Напишите программу, которая принимает с клавиатуры число N и заполняет матрицу размера N следующим образом:

N=1	N=2	N=3	N=4	N=5	...
1	11	111	1111	11111	
	11	010	0100	01000	
		111	0010	00100	
			1111	00010	...
				11111	

**Задача 4.** Напишите программу, которая принимает с клавиатуры размер массива, заполняет его случайными числами из диапазона от 0 до 200 и выводит на экран. Создайте логическую функцию, которая принимает в качестве аргумента целое число и проверяет, заканчивается ли оно цифрой 1 (например, 1, 71 или 121). Используя созданную логическую функцию, определите сколько в этом массиве чисел, которые не оканчиваются цифрой 1, и выведите их.

**Задача 5.** Напишите программу, которая принимает с клавиатуры название файла и выводит на экран «Есть», если в файле присутствуют два одинаковых идущих подряд символа, и «Нет» – в противном случае. Например, если в файле находятся символы «123345», то программа должна вывести «Есть», поскольку две тройки расположены рядом.

### 3.6. Общие положения проведения экзамена и критерии оценки

Экзамены проходят с использованием компьютера. Билет содержит 5 задач, каждая из которых подразумевает написание программы. Оценка определяется количеством решенных задач. Схема оценивания следующая:

- 5 решенных задач – отлично;
- 4 решенные задачи – хорошо;
- 3 решенные задачи – удовлетворительно;
- менее 3-х решенных задач – неудовлетворительно.

## РАЗДЕЛ 4. ЗАДАНИЯ ПО ВЫПОЛНЕНИЮ РГР

### 4.1. Выбор варианта РГР

Варианты выбираются по первой букве фамилии.

<b>А</b>	Вариант 1	<b>М</b>	Вариант 11
<b>Б</b>	Вариант 2	<b>Н</b>	Вариант 12
<b>В</b>	Вариант 3	<b>О</b>	Вариант 13
<b>Г</b>	Вариант 4	<b>П</b>	Вариант 14
<b>Д</b>	Вариант 5	<b>Р</b>	Вариант 15
<b>Е, Ё</b>	Вариант 6	<b>С</b>	Вариант 16
<b>Ж, З</b>	Вариант 7	<b>Т</b>	Вариант 17
<b>И, Й</b>	Вариант 8	<b>У, Ф, Х, Ц</b>	Вариант 18
<b>К</b>	Вариант 9	<b>Ч, Ш, Щ</b>	Вариант 19
<b>Л</b>	Вариант 10	<b>Э, Ю, Я</b>	Вариант 20

### 4.2. Темы РГР согласно вариантам

1. Реализуйте пузырьковую сортировку.
2. Реализуйте сортировку методом прямого выбора.
3. Реализуйте сортировку методом вставок.
4. Реализуйте сортировку методом Шелла.
5. Реализуйте быструю сортировку методом Хоара.
6. Реализуйте сортировку слиянием.
7. Реализуйте пирамидальную сортировку.
8. Реализуйте шейкерную сортировку.
9. Реализуйте двоичный поиск в упорядоченном массиве.
10. Реализуйте решение задачи о Ханойских башнях.
11. Реализуйте алгоритм Рабина-Карпа для поиска подстроки в строке.
12. Реализуйте стек на основе массива.
13. Реализуйте очередь на основе массива.
14. Реализуйте метод правых прямоугольников для численного интегрирования.
15. Реализуйте метод левых прямоугольников для численного интегрирования.
16. Реализуйте метод трапеций для численного интегрирования.
17. Реализуйте метод Симпсона для численного интегрирования.
18. Реализуйте шифр Цезаря.
19. Реализуйте расширенный алгоритм Евклида.
20. Реализуйте алгоритм быстрого возведения в степень по модулю.

## РАЗДЕЛ 5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ КУРСОВОГО ПРОЕКТА

### 5.1. Цель и задачи курсового проекта

Целью курсового проекта является применение, обобщение и закрепление всех навыков и знаний, полученных в ходе изучения дисциплины, в рамках одной программной разработки.

Задачи курсового проекта состоят в закреплении следующих навыков:

- создание и вызов процедур и функций;
- работа с массивом структур;
- реализация динамической структуры данных на основе массиве;
- реализация основных алгоритмов работы со строками и массивами;
- запись/чтение данных в/из файл/файла.

### 5.2. Этапы выполнения курсового проекта

- выбор темы курсового проекта согласно варианту;
- изучение программного кода эталонной программы;
- разработка программы согласно выбранному варианту;
- создание скриншотов выполнения программы;
- оформление текста курсовой работы (см. разделы 3 и 4);
- сдача преподавателю курсового проекта в печатном виде.

### 5.3. Выбор темы курсового проекта

Номер темы	Последние две цифры номера зачетной книжки студента									
	Тема 1	01	11	21	31	41	51	61	71	81
Тема 2	02	12	22	32	42	52	62	72	82	92
Тема 2	02	12	23	33	43	53	63	73	83	93
Тема 4	04	14	24	34	44	54	64	74	84	94
Тема 5	05	15	25	35	45	55	65	75	85	95
Тема 6	06	16	26	36	46	56	66	76	86	96
Тема 7	07	17	27	37	47	57	67	77	87	97
Тема 8	08	18	28	38	48	58	68	78	88	98
Тема 9	09	19	29	39	49	59	69	79	89	99
Тема 10	10	20	30	40	50	60	70	80	90	00



Курсовые проекты, выполненные не по своему варианту, не принимаются.

#### **5.4. Варианты тем курсового проекта**

##### **Тема 1. «Магазин бытовой техники».**

Разработайте консольное приложение, которое предназначено для работы со списком товаров магазина бытовой техники. Для этого создайте структуру из четырёх полей: название товара, производитель, срок гарантии (месяцев), цена. Создайте массив, который может содержать до 20 таких структур. При запуске программы в массиве уже должно содержаться 5 структур, добавленных по умолчанию.

Реализуйте меню из следующих пунктов:

1. вывод всех товаров в наличии;
2. добавление новоготовара;
3. удаление товара с указанным названием;
4. распродажа (снижение цены всех товаров на заданное число процентов);
5. сортировка товаров по сроку гарантии;
6. запись списка всех товаров в файл;
7. считывание списка товаров из файла;
8. выход.

##### **Тема 2. «Спортивная команда».**

Разработайте консольное приложение, которое предназначено для работы со списком членов спортивной команды. Для этого создайте структуру из четырёх полей: фамилияспортсмена, родной город, рост, вес. Создайте массив, который может содержать до 20 таких структур. При запуске программы в массиве уже должно содержаться 5 структур, добавленных по умолчанию.

Реализуйте меню из следующих пунктов:

1. вывод всех спортсменов;
2. добавление новогоспортсмена;
3. удаление спортсмена с указанной фамилией;
4. вывод самого высокого спортсмена;
5. сортировка спортсменов по весу;

6. запись списка всех спортсменов в файл;
7. считывание списка спортсменов из файла;
8. выход.

### **Тема 3. «Список сотрудников».**

Разработайте консольное приложение, которое предназначено для работы со списком сотрудников фирмы. Для этого создайте структуру из четырёх полей: фамилия сотрудника, стаж работы, возраст, зарплата. Создайте массив, который может содержать до 20 таких структур. При запуске программы в массиве уже должно содержаться 5 структур, добавленных по умолчанию.

Реализуйте меню из следующих пунктов:

1. вывод всех сотрудников;
2. добавление новогосотрудника;
3. удаление сотрудника с указанной фамилией;
4. вывод сотрудника с самой большой зарплатой;
5. сортировка сотрудников в алфавитном порядке;
6. запись списка всех сотрудников в файл;
7. считывание списка сотрудников из файла;
8. выход.

### **Тема 4. «Города России».**

Разработайте консольное приложение, которое предназначено для работы со списком городов России. Для этого создайте структуру из четырёх полей: название города, население, географическая широта, регион (край, область и т.д.). Создайте массив, который может содержать до 20 таких структур. При запуске программы в массиве уже должно содержаться 5 структур, добавленных по умолчанию.

Реализуйте меню из следующих пунктов:

1. вывод всех городов;
2. добавление новогорода в базу;
3. удаление города с указанным названием из базы;
4. вывод города с самым большим населением;

5. сортировка городов по алфавиту;
6. запись списка всех городов в файл;
7. считывание списка городов из файла;
8. ВЫХОД.

### **Тема 5. «Страны мира».**

Разработайте консольное приложение, которое предназначено для работы со списком стран мира. Для этого создайте структуру из четырёх полей: название страны, население, площадь, часть света (Европа, Азия и т.д.). Создайте массив, который может содержать до 20 таких структур. При запуске программы в массиве уже должно содержаться 5 структур, добавленных по умолчанию.

Реализуйте меню из следующих пунктов:

1. вывод всех стран;
2. добавление новой страны в базу;
3. удаление страны с указанным названием из базы;
4. вывод страны с самой большой площадью;
5. сортировка стран по населению;
6. запись списка всех стран в файл;
7. считывание списка стран из файла;
8. ВЫХОД.

### **Тема 6. «Музыкальные группы».**

Разработайте консольное приложение, которое предназначено для работы со списком музыкальных групп. Для этого создайте структуру из четырёх полей: название группы, фамилия лидера группы, число участников, родная страна. Создайте массив, который может содержать до 20 таких структур. При запуске программы в массиве должно содержаться 5 структур, добавленных по умолчанию.

Реализуйте меню из следующих пунктов:

1. вывод всех групп;
2. добавление новой группы;

3. удаление группы с указанным названием;
4. вывод отечественных групп;
5. сортировка групп по фамилии лидера;
6. запись списка всех групп в файл;
7. считывание списка групп из файла;
8. выход.

### **Тема 7. «Футбольные клубы».**

Разработайте консольное приложение, которое предназначено для работы со списком футбольных клубов. Для этого создайте структуру из четырёх полей: название клуба, тренер, количество побед в чемпионате, бюджет. Создайте массив, который может содержать до 20 таких структур. При запуске программы в массиве уже должно содержаться 5 структур, добавленных по умолчанию.

Реализуйте меню из следующих пунктов:

1. вывод всех клубов;
2. добавление нового клуба;
3. удаление клуба с указанным названием;
4. вывод клубов, побеждавших в чемпионате хотя бы раз;
5. сортировка клубов по фамилии тренера;
6. запись списка всех клубов в файл;
7. считывание списка клубов из файла;
8. выход.

### **Тема 8. «База данных отелей».**

Разработайте консольное приложение, которое предназначено для работы со списком отелей. Для этого создайте структуру из четырёх полей: название отеля, город, количество звёзд, цена за сутки. Создайте массив, который может содержать до 20 таких структур. При запуске программы в массиве уже должно содержаться 5 структур, добавленных по умолчанию.

Реализуйте меню из следующих пунктов:

1. вывод всех отелей;
2. добавление нового отеля;

3. удаление отеля с указанным названием;
4. вывод пятизвёздочных отелей;
5. сортировка отелей по цене размещения;
6. запись списка всех отелей в файл;
7. считывание списка отелей из файла;
8. выход.

### **Тема 8. «База данных отелей».**

Разработайте консольное приложение, которое предназначено для работы со списком отелей. Для этого создайте структуру из четырёх полей: название отеля, город, количество звёзд, цена за сутки. Создайте массив, который может содержать до 20 таких структур. При запуске программы в массиве уже должно содержаться 5 структур, добавленных по умолчанию.

Реализуйте меню из следующих пунктов:

1. вывод всех отелей;
2. добавление нового отеля;
3. удаление отеля с указанным названием;
4. вывод пятизвёздочных отелей;
5. сортировка отелей по цене размещения;
6. запись списка всех отелей в файл;
7. считывание списка отелей из файла;
8. выход.

### **Тема 9. «Автосалон».**

Разработайте консольное приложение, которое предназначено для работы со списком автомобилей для продажи. Для этого создайте структуру из четырёх полей: марка-модель, страна-производитель, объём двигателя, цена. Создайте массив, который может содержать до 20 таких структур. При запуске программы в массиве уже должно содержаться 5 структур, добавленных по умолчанию.

Реализуйте меню из следующих пунктов:

1. вывод всех автомобилей;
2. добавление нового автомобиля;

3. удаление автомобиля с указанной маркой-моделью;
4. вывод отечественных автомобилей;
5. сортировка автомобилей по объёму двигателя;
6. запись списка всех автомобилей в файл;
7. считывание списка автомобилей из файла;
8. выход.

### **Тема 10. «Киноактеры».**

Разработайте консольное приложение, которое предназначено для работы со списком известных киноактеров. Для этого создайте структуру из четырёх полей: фамилия, принёсший популярность фильм, возраст, родная страна. Создайте массив, который может содержать до 20 таких структур. При запуске программы в массиве уже должно содержаться 5 структур, добавленных по умолчанию.

Реализуйте меню из следующих пунктов:

1. вывод всех киноактеров;
2. добавление нового киноактера;
3. удаление киноактера с указанной фамилией;
4. вывод отечественных киноактеров;
5. сортировка киноактеров по возрасту;
6. запись списка всех киноактеров в файл;
7. считывание списка киноактеров из файла;
8. выход.

### **5.5. Правила оформления текста курсового проекта**

Оформление текста курсового проекта осуществляется в редакторе Microsoft Word или ему подобном. Примеры оформления отдельных структурных элементов приведены далее в этом разделе. Титульный лист оформляют в соответствии с образцом (см. п. 2.6).

Работа выполняется на белой бумаге на одной стороне листа А4 (210×297 мм) через 1 или 1,5 интервала, шрифтом Times New Roman, 12 пт., форматирование текста по ширине, заголовков — по центру; страница должна иметь поля: левое – 2,5 см, правое – 1,5 см, верхнее – 2 см, нижнее – 2 см. Аб-

защный отступ –1,25 см. Для оформления программного кода следует использовать моноширинный шрифт (например, CourierNew), выровненный по левому краю.

Страницы текста курсового проекта нумеруются арабскими цифрами в правом нижнем углу. На титульном листе и оглавлении цифры не проставляются, но включаются в общую нумерацию страниц.

## **5.6. Пример программного кода, полученного в результате разработки приложения согласно теме курсового проекта**

**Задание.**Разработайте консольное приложение «Книжный каталог». Для этого создайте структуру из четырех полей: название книги, автор, тираж, цена. Создайте массив, который может содержать до 20 книг. При запуске программы в массиве уже должно быть добавлено 5 структур по умолчанию.

Реализуйте меню из следующих пунктов:

- вывод всех книг;
- добавление новой книги (если каталог уже содержит 20 книг, то сообщить о невозможности добавления);
- удаление выбранной книги;
- распродажа (снижение цены всех книг на заданное число процентов);
- сортировка книг по цене;
- запись каталога в файл;
- считывание каталога из файла;
- ВЫХОД.

## **5.7. Пример программного кода, полученного в результате разработки приложения согласно теме курсового проекта**

```
#include <iostream>
#include <fstream>
#include <windows.h>
#include <string>
using namespace std;

const int MAX = 20; int n;

//Объявлениеструктуры
struct Book {
    string name;
```

```

        string author;
        int circulation;
        double price;
};

//Создание массива структур
Book books[MAX];
ofstream fout;
ifstream fin;

//Запись в каталог книг по умолчанию
void initBooks() {
    books[0].name = "Преступление и наказание";
    books[0].author = "Достоевский";
    books[0].circulation = 100000;
    books[0].price = 12.55;
    books[1].name = "Война и мир";
    books[1].author = "Толстой";
    books[1].circulation = 200000;
    books[1].price = 15.61;
    books[2].name = "Совершенный код";
    books[2].author = "Макконнелл";
    books[2].circulation = 5000;
    books[2].price = 20.55;
    books[3].name = "Теория вероятностей";
    books[3].author = "Боровков";
    books[3].circulation = 2000;
    books[3].price = 62.45;
    books[4].name = "Диалоги";
    books[4].author = "Платон";
    books[4].circulation = 70000;
    books[4].price = 34.53;
    n = 5;
}

//Вывод книг на экран
void printBooks() {
    for (int i = 0; i < n; i++) {
        cout.width(25);
        cout << books[i].name;
        cout.width(15);
        cout << books[i].author;
        cout.width(10);
        cout << books[i].circulation;
        cout.width(10);
        cout << books[i].price << endl;
    }
}

//Вычисление скидки
void discount() {

```



```

    cout << "На сколько процентов?\n";
    double percent;
    cin >> percent;
    for (int i = 0; i<n; i++) {
        books[i].price *= (100 - percent) / 100;
    }
}

//Добавление новой книги
void addBook() {
    if (n == MAX) {
        cout<< "Каталогполон\n";
        return;
    }
    cout << "Введите название:\n";
    getline(cin, books[n].name);
    cout << "Введите автора:\n";
    getline(cin, books[n].author);
    cout << "Введите тираж:\n";
    cin >> books[n].circulation;
    cout << "Введите цену:\n";
    cin >> books[n].price;
    n++;
}

//Удаление книги
void removeBook() {
    if (n == 0) {
        cout<< "Каталогпуст\n";
        return;
    }

    cout << "Введите название книги:\n";
    string bname;
    getline(cin, bname);

    int i_book = -1;
    for (int i = 0; i < n; i++) {
        if (bname == books[i].name) {
            i_book = i;
            break;
        }
    }

    if (i_book == -1) {
        cout << "Такой книги в каталоге нет\n";
        return;
    }

    books[i_book] = books[n-1];
    n--;
}

```

```

        cout << "Книга удалена\n";
    }

//Сортировка книг по цене
void sortBooks() {
    for (int i = 0; i < n - 1; i++) {
        int j_max = i;
        for (int j = i + 1; j < n; j++) {
            if (books[j_max].price < books[j].price) {
                j_max = j;
            }
        }
        Book temp = books[i];
        books[i] = books[j_max];
        books[j_max] = temp;
    }
}

//Вывод меню на экран
void printMenu() {
    cout << "1 - Вывод всех книг\n";
    cout << "2 - Добавить книгу\n";
    cout << "3 - Удалить книгу\n";
    cout << "4 - Сортировка книг по цене\n";
    cout << "5 - Снизить цену всех книг\n";
    cout << "6 - Записать данные в файл\n";
    cout << "7 - Считать данные из файла\n";
    cout << "0 - Выход\n";
    cout << "Ваш выбор: \n";
}

//Запись каталога в файл
void writeData() {
    fout.open("books.txt");
    fout << n << endl;
    for (int i = 0; i < n; i++) {
        fout << books[i].name << endl;
        fout << books[i].author << endl;
        fout << books[i].circulation << endl;
        fout << books[i].price << endl;
    }
    fout.close();
}

//Чтение каталога из файла
void readData() {
    fin.open("books.txt");
    fin >> n;
    fin.ignore();
    for (int i = 0; i < n; i++) {
        getline(fin, books[i].name);
    }
}

```

```

        getline(fin, books[i].author);
        fin >> books[i].circulation;
        fin.ignore();
        fin >> books[i].price;
        fin.ignore();
    }
    fin.close();
}

```

Реализация работы меню

```

intmain() {
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    initBooks();
    while (true) {
        system("cls");
        printMenu();
        int choice;
        cin >> choice;
        cin.ignore();
        switch (choice) {
            case 1: printBooks(); break;
            case 2: addBook(); break;
            case 3: removeBook(); break;
            case 4: sortBooks(); break;
            case 5: discount(); break;
            case 6: writeData(); break;
            case 7: readData(); break;
            case 0: exit(0);
            default: cout<< "Неверныйвыбор\n";
        }
        system("pause");
    }
    return 0;
}

```

## 5.8. Примеры оформления скриншотов программы

Вывод всех книг на экран:

```
1 - Вывод всех книг
2 - Добавить книгу
3 - Удалить книгу
4 - Сортировка книг по цене
5 - Снизить цену всех книг
6 - Записать данные в файл
7 - Считать данные из файла
0 - Выход
Ваш выбор:
1
Преступление и наказание      Достоевский      100000      12.55
    Война и мир                Толстой          200000      15.61
    Совершенный код            Макконнелл       5000        20.55
    Теория вероятностей        Боровков         2000        62.45
    Диалоги                    Платон           70000       34.53
Для продолжения нажмите любую клавишу . . . _
```

Добавление новой книги:

```
1 - Вывод всех книг
2 - Добавить книгу
3 - Удалить книгу
4 - Сортировка книг по цене
5 - Снизить цену всех книг
6 - Записать данные в файл
7 - Считать данные из файла
0 - Выход
Ваш выбор:
2
Введите название:
Великая шахматная доска
Введите автора:
Бжезинский
Введите тираж:
1000
Введите цену:
700
Для продолжения нажмите любую клавишу . . .
```

Вывод списка после добавления:

```
1 - Вывод всех книг
2 - Добавить книгу
3 - Удалить книгу
4 - Сортировка книг по цене
5 - Снизить цену всех книг
6 - Записать данные в файл
7 - Считать данные из файла
0 - Выход
Ваш выбор:
1
Преступление и наказание      Достоевский      100000      12.55
    Война и мир                Толстой          200000      15.61
    Совершенный код            Макконнелл       5000        20.55
    Теория вероятностей        Боровков         2000        62.45
    Диалоги                    Платон           70000       34.53
    Великая шахматная доска    Бжезинский       1000         700
Для продолжения нажмите любую клавишу . . . _
```

Удаление книги (книга найдена):

```
1 - Вывод всех книг
2 - Добавить книгу
3 - Удалить книгу
4 - Сортировка книг по цене
5 - Снизить цену всех книг
6 - Записать данные в файл
7 - Считать данные из файла
0 - Выход
Ваш выбор:
3
Введите название книги:
Война и мир
Книга удалена
Для продолжения нажмите любую клавишу . . .
```

Вывод списка после удаления:

```
1 - Вывод всех книг
2 - Добавить книгу
3 - Удалить книгу
4 - Сортировка книг по цене
5 - Снизить цену всех книг
6 - Записать данные в файл
7 - Считать данные из файла
0 - Выход
Ваш выбор:
1
Преступление и наказание    Достоевский    100000    12.55
Великая шахматная доска    Бжезинский     1000      700
Совершенный код             Макконнелл     5000      20.55
Теория вероятностей         Боровков       2000      62.45
Диалоги                     Платон         70000     34.53
Для продолжения нажмите любую клавишу . . .
```

Снижение цены:

```
1 - Вывод всех книг
2 - Добавить книгу
3 - Удалить книгу
4 - Сортировка книг по цене
5 - Снизить цену всех книг
6 - Записать данные в файл
7 - Считать данные из файла
0 - Выход
Ваш выбор:
5
На сколько процентов?
19
Для продолжения нажмите любую клавишу . . .
```

Вывод каталога после снижения цены:

```
1 - Вывод всех книг
2 - Добавить книгу
3 - Удалить книгу
4 - Сортировка книг по цене
5 - Снизить цену всех книг
6 - Записать данные в файл
7 - Считать данные из файла
0 - Выход
Ваш выбор:
1
Преступление и наказание    Достоевский    100000    10.1655
Война и мир                 Толстой        200000    12.6441
Совершенный код             Макконнелл     5000      16.6455
Теория вероятностей         Боровков       2000      50.5845
Диалоги                     Платон         70000     27.9693
Для продолжения нажмите любую клавишу . . .
```

Вывод каталога после сортировки по цене:

```
1 - Вывод всех книг
2 - Добавить книгу
3 - Удалить книгу
4 - Сортировка книг по цене
5 - Снизить цену всех книг
6 - Записать данные в файл
7 - Считать данные из файла
0 - Выход
Ваш выбор:
1
    Теория вероятностей      Боровков      2000      50.5845
      Диалоги                Платон        70000     27.9693
    Совершенный код          Макконнелл    5000      16.6455
      Война и мир            Толстой       200000    12.6441
    Преступление и наказание Достоевский   100000    10.1655
Для продолжения нажмите любую клавишу . . .
```

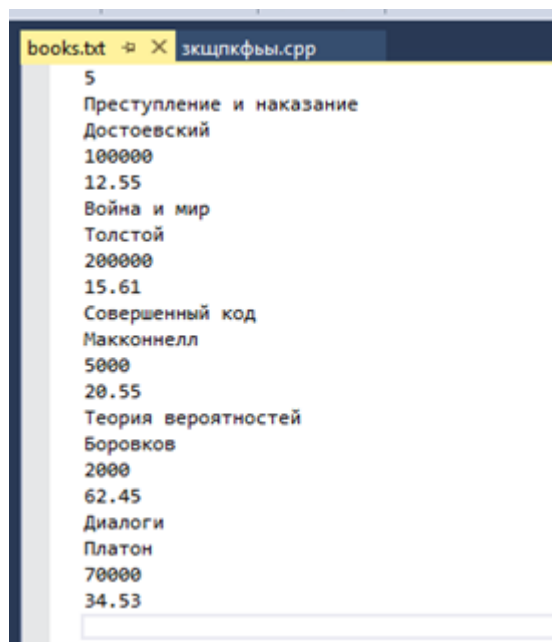
Файл, из которого планируется считать информацию

```
books.txt  X  зкщпкфьы.cpp
3
Бесы
Достоевский
10000
17.55
Структура научных революций
Кун
1000
23.55
Искусство программирования
Кнут
10000
29.55
```

Каталог после считывания информации из файла:

```
1 - Вывод всех книг
2 - Добавить книгу
3 - Удалить книгу
4 - Сортировка книг по цене
5 - Снизить цену всех книг
6 - Записать данные в файл
7 - Считать данные из файла
0 - Выход
Ваш выбор:
1
    Структура научных революций  Бесы      Достоевский   10000      17.55
      Искусство программирования  Кун        1000          23.55
      Искусство программирования  Кнут       10000         29.55
Для продолжения нажмите любую клавишу . . .
```

Файл после записи в него книг по умолчанию:



```
books.txt  X  зкщпкфвы.cpp
5
Преступление и наказание
Достоевский
100000
12.55
Война и мир
Толстой
200000
15.61
Совершенный код
Макконнелл
5000
20.55
Теория вероятностей
Боровков
2000
62.45
Диалоги
Платон
70000
34.53
```

## 5.9. Критерии оценивания

Оценка выставляется в зависимости от количества и качества реализованных пунктов меню согласно заданию: 8 число реализованных пунктов – «отлично»; 6-7 пунктов – «хорошо»; 4-5 пунктов – «удовлетворительно»; менее 4 пунктов – «неудовлетворительно».

«Чисто реализованный пункт» означает, что он правильно работает, имеет комментарии и скриншоты, показывающие его работу.

## 5.10. Оформление титульного листа

Министерство образования и науки Российской Федерации  
Новосибирский государственный университет экономики и управления «НИНХ»

Кафедра прикладных информационных технологий

Учебная дисциплина: Программирование

Тема курсового проекта: \_\_\_\_\_

Номер группы: \_\_\_\_\_

Наименование направления подготовки: «Бизнес-информатика»

Наименование профиля подготовки: «Электронный бизнес»

Номер зачетной книжки: \_\_\_\_\_

ФИО студента: \_\_\_\_\_

Подпись студента: \_\_\_\_\_

Дата регистрации кафедрой: «\_\_\_\_\_» \_\_\_\_\_ 2016 г.

Проверил: к.ф.-м.н., доцент кафедры ПИТ, Пестунов Андрей Игоревич

### Выполненные пункты:

Номер пункта	Количество баллов	Номер пункта	Количество баллов
1		5	
2		6	
3		7	
4		8	

Количество выполненных пунктов:

Оценка:

Новосибирск, 2016