

*МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ*

федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Курганский государственный университет»

Кафедра программного обеспечения автоматизированных систем

**УПРАВЛЕНИЕ КАЧЕСТВОМ И ТЕСТИРОВАНИЕ  
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

Методические указания  
к выполнению лабораторных работ  
для студентов направления подготовки 09.03.04  
«Программная инженерия»

Курган 2017

Кафедра: «Программное обеспечение автоматизированных систем».

Направление подготовки: 09.03.04 «Программная инженерия».

Дисциплина: «Тестирование и управление качеством программного обеспечения».

Составил: канд. техн. наук, доц. А.М. Семахин.

Утверждены  
на заседании кафедры « 29 » июня 2017 г.

Рекомендованы методическим советом университета

« 12 » декабря 2016 г.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 Основные понятия и определения тестирования	4
1.1 Этапы тестирования программы	4
1.2 Лабораторная работа № 1	
Тестирование компиляторов языка программирования Си на соответствие стандарту ANSI ISO/IEC 9899:1999	5
2 Структурное тестирование программ	7
2.1 Критерии структурного тестирования	7
2.2 Лабораторная работа № 2	
Тестирование программ методами «белого ящика»	7
2.3 Лабораторная работа № 3	
Разработка управляющего потокового графа	14
3 Функциональное тестирование программ	16
3.1 Функциональные критерии тестирования	17
3.2 Лабораторная работа № 4	
Тестирование программ методами «черного ящика». Дерево решений	17
3.3 Лабораторная работа № 5	
Тестирование программ методами «черного ящика». Диаграмма причинно-следственных связей	21
4 Интеграционное тестирование	24
4.1 Методы интеграционного тестирования	24
4.2 Лабораторная работа № 6	
Нисходящее тестирование	25
4.3 Лабораторная работа № 7	
Восходящее тестирование	30
5. Автоматизированное тестирование	35
5.1 Способы автоматизации тестирования	35
5.2 Лабораторная работа № 8	
«Автоматизированное тестирование программ»	35
ЗАКЛЮЧЕНИЕ	41
СПИСОК ЛИТЕРАТУРЫ	41

## **ВВЕДЕНИЕ**

Дисциплина «Управление качеством и тестирование программного обеспечения» имеет цель дать студентам теоретические знания и практические навыки в выявлении дефектов программных продуктов.

Предмет дисциплины – технология тестирования программных приложений.

Задачи дисциплины – дать представление о качестве и надёжности программного обеспечения, изучение методов тестирования программных продуктов, формирование навыков тестирования программных приложений.

Лабораторный практикум (34 часов).

Методические указания содержат теоретическое обоснование и варианты заданий для выполнения лабораторных работ по дисциплине «Управление качеством и тестирование программного обеспечения».

Методические указания разработаны в соответствии с требованиями государственного образовательного стандарта по подготовке бакалавров по направлению 09.03.04 «Программная инженерия».

### **1 Основные понятия и определения тестирования**

Тестирование – процесс обнаружения ошибок в программе.

Отладка – процесс обнаружения и исправления ошибок в программе.

Тестовый случай – набор входных данных, на которых программа исполняется в процессе тестирования.

Тестирование приемлемости – проверка соответствия программы требованиям пользователя.

Тестирование настройки – процесс выявления ошибок, возникших в процессе настройки системы.

#### **1.1 Этапы тестирования программы**

Тестирование включает три этапа:

- создание тестового набора (test case) для среды тестирования (testing environment);

- прогон программы на тестах с получением протокола результатов тестирования (test log);

- оценка результатов выполнения программы на наборе тестов с целью принятия решения о продолжении или остановке тестирования.

Основная проблема тестирования – определение достаточности множества тестов и определение данных тестов [1; 2; 3; 4].

## 1.2 Лабораторная работа № 1

### Тестирование компиляторов языка программирования Си на соответствие стандарту ANSI ISO/IEC 9899:1999

**Цель:** закрепить теоретические знания и получить практические навыки тестирования программных продуктов.

**Используемые приемы и технологии:** компилятор языка C/C++, набор тестовых вариантов.

**Ключевые термины:** тестирование, отладка, критерии тестирования, этапы тестирования.

#### Постановка задачи

Выполнить тестирование компиляторов языка программирования Си на соответствие стандарту ANSI ISO/IEC 9899:1999 в части декларативных операторов. На рисунке 1 показана постановка задачи. Для выполнения тестирования используется набор тестовых вариантов. Результаты тестирования приведены на рисунке 2 и сводятся в таблицу 1. Принимается решение о соответствии компилятора стандарту.

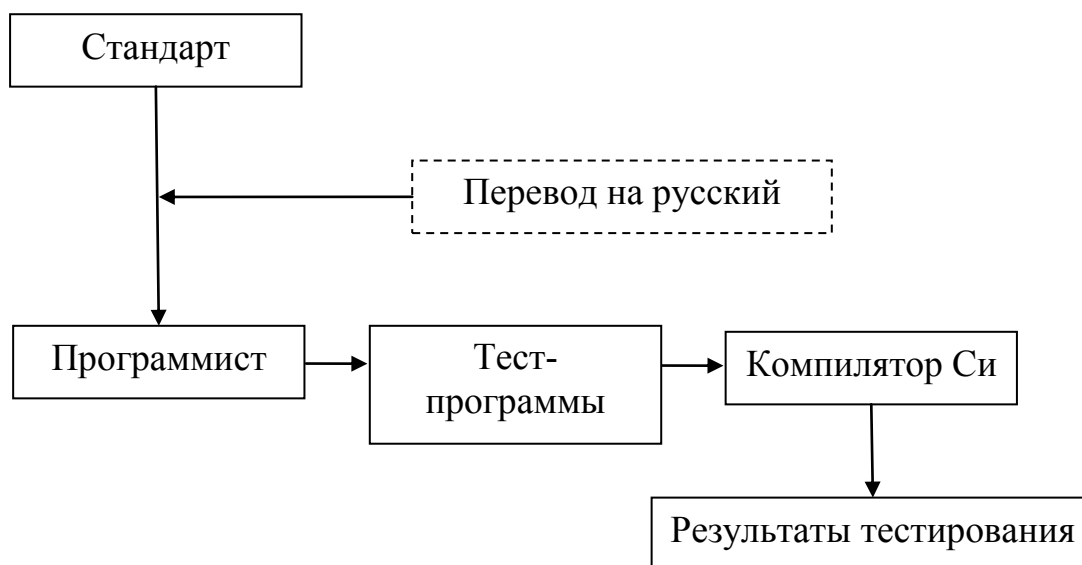


Рисунок 1 – Постановка задачи

Таблица 1 – Результаты тестирования

Тест	Результаты тестирования	
	Компилятор языка	Результат

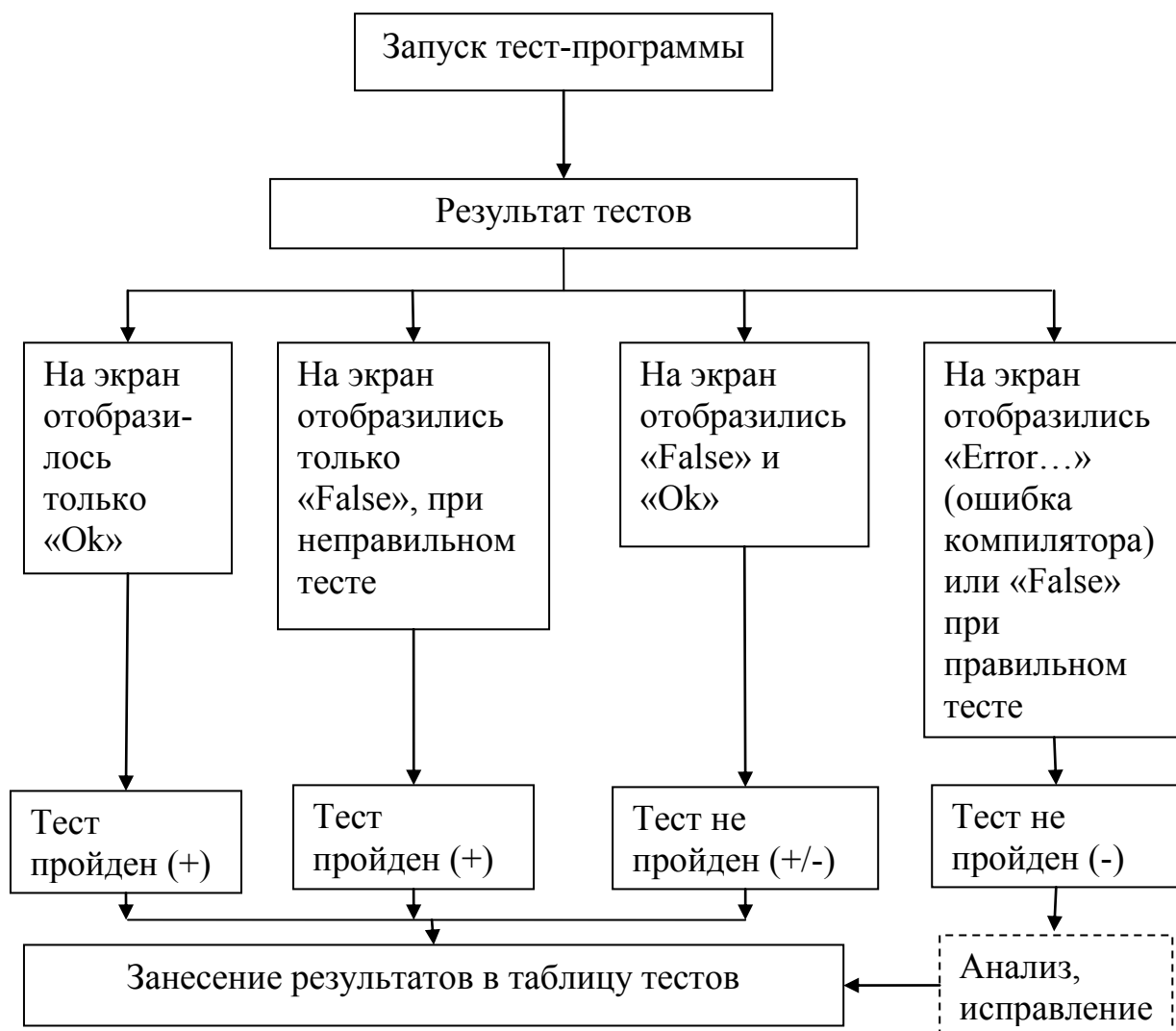


Рисунок 2 – Результаты тестирования

### Варианты заданий

Компилятор языка программирования C++ выбирается студентом или выдается преподавателем. Набор тестовых вариантов выдается преподавателем.

### Методические указания

- 1 Выбрать компилятор языка C++.
- 2 Получить набор тестовых вариантов у преподавателя:
  - запустить тестовые варианты;
  - занести результаты выполнения тестовых вариантов в таблицу;
  - представить результаты тестирования в графическом виде (диаграммы);
  - провести анализ результатов тестирования;
- 3 Оформить отчет по лабораторной работе, включающий разделы:

- Постановка задачи.
- Таблица выполнения тестовых вариантов.
- Графическое представление результатов тестирования.
- Выводы.

## **Контрольные вопросы**

- 1 Что называется тестированием программы?
- 2 Что называется отладкой программы?
- 3 В чем отличие тестирования программы от отладки программы?
- 4 Какие этапы включает процесс тестирования программы?

## **2 Структурное тестирование программ**

Структурное тестирование – процесс обнаружения ошибок в логике программы. Методы структурного тестирования:

- покрытие операторов;
- покрытие решений;
- покрытие условий;
- покрытие решений / условий;
- комбинаторное покрытие условий.

Покрываемость кода тестами – мера, показывающая на сколько процентов исходный код программы был протестирован.

### **2.1 Критерии структурного тестирования**

Структурные критерии используют модель программы в виде «белого ящика». Структурные критерии базируются на основных элементах управляющего потокового графа, операторах, ветвях и путях:

- критерий тестирования команд (критерий C0) – набор тестов, обеспечивающий прохождение команды не менее одного раза;
- критерий тестирования ветвей (критерий C1) – набор тестов, обеспечивающий прохождение каждой ветви не менее одного раза.
- критерий тестирования путей (критерий C2) – набор тестов, обеспечивающий прохождение каждого пути не менее одного раза [4; 5; 6; 7].

### **2.2 Лабораторная работа № 2**

#### **Тестирование программ методами «белого ящика»**

**Цель:** закрепить теоретические знания и получить практические навыки в разработке программы тестирования методами по стратегии «белого ящика».

**Используемые приемы и технологии:** технология визуального проектирования и событийного программирования, среда программирования

Visual Studio 2010 Professional, язык программирования Visual C++ 2010 Professional.

**Ключевые термины:** логика программы, критерии структурного тестирования, покрытие, методы структурного тестирования.

### Варианты заданий

Протестировать методами «белого ящика» программу, вычисляющую функцию, где  $a, b, c$  – действительные числа.

#### Вариант 1

$$y = \begin{cases} a * x^2 + b, & \text{при } x + 2 < 0, b \neq 0 \\ \frac{x - a}{x - c}, & \text{при } x + 2 > 0 \text{ и } c \neq 0 \\ \frac{10 * x}{c - 4}, & \text{в остальных случаях} \end{cases}$$

#### Вариант 2

$$y = \begin{cases} \frac{1}{a * x} - b, & \text{при } x + 5 < 0, c = 0 \\ \frac{x - a}{x}, & \text{при } x + 5 > 0 \text{ и } c \neq 0 \\ \frac{10 * x}{c - 4}, & \text{в остальных случаях} \end{cases}$$

#### Вариант 3

$$y = \begin{cases} a * x^2 + b * x + c, & \text{при } a < 0, c \neq 0 \\ \frac{-a}{x - c}, & \text{при } a > 0 \text{ и } c = 0 \\ a * (x + c), & \text{в остальных случаях} \end{cases}$$

#### Вариант 4

$$y = \begin{cases} -a * x - c, & \text{при } c < 0, x \neq 0 \\ \frac{x - a}{-c}, & \text{при } c > 0 \text{ и } x = 0 \\ \frac{b * x}{c - a}, & \text{в остальных случаях} \end{cases}$$



### Вариант 5

$$y = \begin{cases} a - \frac{x}{10 + b}, \text{ при } x < 0, b \neq 0 \\ \frac{x - a}{x - c}, \text{ при } x > 0 \text{ и } b = 0 \\ 3 * x + \frac{2}{c}, \text{ в остальных случаях} \end{cases}$$

### Вариант 6

$$y = \begin{cases} a * x^2 + b^2 * x, \text{ при } c < 0, b \neq 0 \\ \frac{x + a}{x + c}, \text{ при } c > 0 \text{ и } b = 0 \\ \frac{x}{c}, \text{ в остальных случаях} \end{cases}$$

### Вариант 7

$$y = \begin{cases} -a * x^2 - b, \text{ при } x < 5, c \neq 0 \\ \frac{x - a}{x}, \text{ при } x > 5 \text{ и } c = 0 \\ \frac{-x}{c}, \text{ в остальных случаях} \end{cases}$$

### Вариант 8

$$y = \begin{cases} -a * x^2, \text{ при } c < 0, a \neq 0 \\ \frac{a - x}{c * x}, \text{ при } c > 0 \text{ и } a = 0 \\ \frac{x}{c}, \text{ в остальных случаях} \end{cases}$$

### Вариант 9

$$y = \begin{cases} a * x^2 + b^2 * x, \text{ при } a < 0, x \neq 0 \\ x - \frac{a}{x - c}, \text{ при } a > 0 \text{ и } x = 0 \\ 1 + \frac{x}{c}, \text{ в остальных случаях} \end{cases}$$

### Вариант 10

$$y = \begin{cases} a * x^2 - b * x + c, \text{ при } x < 3, b \neq 0 \\ \frac{x - a}{x - c}, \text{ при } x > 3 \text{ и } b = 0 \\ \frac{x}{c}, \text{ в остальных случаях} \end{cases}$$

### Вариант 11

$$y = \begin{cases} a * x^2 + \frac{b}{c}, \text{ при } x < 15, c \neq 0 \\ \frac{x - a}{(x - c)^2}, \text{ при } x > 15 \text{ и } c = 0 \\ \frac{x^2}{c^2}, \text{ в остальных случаях} \end{cases}$$

### Вариант 12

$$y = \begin{cases} a * x^3 + b^2 + c, \text{ при } x < 0.6, b + c \neq 0 \\ \frac{x - a}{x - c}, \text{ при } x > 0.6 \text{ и } b + c = 0 \\ \frac{x}{c} + \frac{x}{a}, \text{ в остальных случаях} \end{cases}$$

### Вариант 13

$$y = \begin{cases} a * x^2 + b, \text{ при } x - 1 < 0, b - x \neq 0 \\ \frac{x - a}{x}, \text{ при } x - 1 > 0 \text{ и } b + x = 0 \\ \frac{x}{c}, \text{ в остальных случаях} \end{cases}$$

### Вариант 14

$$y = \begin{cases} -a * x^3 - b, \text{ при } x + c < 0, a \neq 0 \\ \frac{x - a}{x - c}, \text{ при } x + c > 0 \text{ и } a = 0 \\ \frac{x}{c} + \frac{c}{x}, \text{ в остальных случаях} \end{cases}$$

### Вариант 15

$$y = \begin{cases} -a * x^2 + b, \text{ при } x < 0, b \neq 0 \\ \frac{x}{x - c} + 5.5, \text{ при } x > 0 \text{ и } b = 0 \\ \frac{x}{-c}, \text{ в остальных случаях} \end{cases}$$

### Вариант 16

$$y = \begin{cases} a * (x + c)^2 - b, \text{ при } x = 0, b \neq 0 \\ \frac{x - a}{-c}, \text{ при } x = 0 \text{ и } b = 0 \\ a + \frac{x}{c}, \text{ в остальных случаях} \end{cases}$$

### Вариант 17

$$y = \begin{cases} a * x^2 - cx + b, \text{ при } x + 10 < 0, b \neq 0 \\ \frac{x - a}{x - c}, \text{ при } x + 10 > 0 \text{ и } b = 0 \\ \frac{-x}{a - c}, \text{ в остальных случаях} \end{cases}$$

### Вариант 18

$$y = \begin{cases} a * x^3 + b * x^2, \text{ при } x < 0, b \neq 0 \\ \frac{x - a}{x - c}, \text{ при } x > 0 \text{ и } b = 0 \\ \frac{x + 5}{c(x - 10)}, \text{ в остальных случаях} \end{cases}$$

**Вариант 19**

$$y = \begin{cases} a * (x + 7)^2 - b, \text{ при } x < 5, b \neq 0 \\ \frac{x - c * d}{a * x}, \text{ при } x > 5 \text{ и } b = 0 \\ \frac{x}{c}, \text{ в остальных случаях} \end{cases}$$

**Вариант 20**

$$y = \begin{cases} -\frac{2 * x - c}{c * x - a}, \text{ при } x < 0, b \neq 0 \\ \frac{x - a}{x - c}, \text{ при } x > 0 \text{ и } b = 0 \\ -\frac{x}{c} + \frac{-b}{2 * c}, \text{ в остальных случаях} \end{cases}$$

**Вариант 21**

$$y = \begin{cases} a * x^3 + b^2 + c, \text{ при } x < 0, b \neq 0 \\ \frac{x - a}{c}, \text{ при } x > 0 \text{ и } c \neq 0 \\ \frac{15 * x}{c}, \text{ в остальных случаях} \end{cases}$$

**Вариант 22**

$$y = \begin{cases} a * x^2 + 2 * b * x + 5, \text{ при } x < 5, b \neq 0 \\ \frac{3 * x - a}{c}, \text{ при } x \geq 5 \text{ и } c \neq 0 \\ \frac{10 * x}{c + 6}, \text{ в остальных случаях} \end{cases}$$

**Вариант 23**

$$y = \begin{cases} (a + 2) * x^2 + 5 * b, \text{ при } x < 10, b \neq 0 \\ \frac{x + a}{7 * c}, \text{ при } x \geq 10 \text{ и } c \neq 0 \\ \frac{x}{c + 14}, \text{ в остальных случаях} \end{cases}$$

### Вариант 24

$$y = \begin{cases} a * x^4 + b * x^2, & \text{при } x < 0, b \neq 0 \\ \frac{x + a}{c}, & \text{при } x \geq 0 \text{ и } c \neq 0 \\ \frac{15 * x}{c + 9}, & \text{в остальных случаях} \end{cases}$$

### Вариант 25

$$y = \begin{cases} a * x^2 + 25 * b * x, & \text{при } x < 0, b \neq 0 \\ \frac{x - a}{c}, & \text{при } x \geq 0 \text{ и } c \neq 0 \\ \frac{16 * x}{c + 8}, & \text{в остальных случаях} \end{cases}$$

### Методические указания

- 1 Разработать алгоритм решения задачи.
- 2 Разработать визуальное приложение, осуществляющее тестирование программы по вариантам задания методами структурного тестирования:
  - покрытие операторов;
  - покрытие решений (покрытие переходов);
  - покрытие условий;
  - покрытие условий / решений;
  - комбинаторное покрытие условий.
- 3 Результаты тестирования представить в виде таблиц.
- 4 Оформить отчет по лабораторной работе, включающий разделы:
  - Постановка задачи.
  - Блок-схема программы.
  - Тестовые варианты и результаты тестирования.
  - Скриншоты программы.
  - Код программы (функции обработчиков событий).
  - Выводы.

### Контрольные вопросы

- 1 Какие этапы включает реализация и тестирование программного продукта?
- 2 Какие виды тестирования?
- 3 Какие критерии выбора тестов?
- 4 Какие свойства тестов?

5 Какие критерии надежности программ?

6 Какова оценка надежности программ?

### 2.3 Лабораторная работа № 3 Разработка управляющего потокового графа

**Цель:** закрепить теоретические знания и получить практические навыки в разработке программы тестирования методом базового пути.

**Используемые приемы и технологии:** технология визуального проектирования и событийного программирования, среда программирования Visual Studio 2010 Professional, язык программирования Visual C++ 2010 Professional.

**Ключевые термины:** управляющий потоковый граф программы, цикломатическая сложность, путь, ветвь, узел, дуга, операторный узел, предикатный узел, независимый путь, регион.

#### Варианты заданий

Протестировать методом базового пути программу, формализующую алгоритм варианта задачи.

**Вариант 1.** В одномерном динамическом массиве, состоящем из  $n$  элементов, вычислить сумму элементов массива, расположенных после минимального элемента.

**Вариант 2.** В одномерном динамическом массиве, состоящем из  $n$  элементов, вычислить сумму элементов массива, расположенных между минимальным и максимальным элементами.

**Вариант 3.** В одномерном динамическом массиве, состоящем из  $n$  элементов, вычислить среднеарифметическое значение элементов массива.

**Вариант 4.** В одномерном динамическом массиве, состоящем из  $n$  элементов, вычислить среднеквадратическое значение элементов массива.

**Вариант 5.** В одномерном динамическом массиве, состоящем из  $n$  элементов, вычислить среднегеометрическое значение ненулевых элементов массива.

**Вариант 6.** В одномерном динамическом массиве, состоящем из  $n$  элементов, вычислить среднегармоническое значение положительных элементов массива.

**Вариант 7.** В одномерном динамическом массиве, состоящем из  $n$  элементов, поменять местами максимальный и минимальный элементы.

**Вариант 8.** В одномерном динамическом массиве, состоящем из  $n$  элементов, вычислить сумму элементов массива, расположенных между первым и последним отрицательными элементами.

**Вариант 9.** В одномерном динамическом массиве, состоящем из  $n$  элементов, вычислить сумму элементов массива, расположенных до последнего положительного элемента.

**Вариант 10.** В одномерном динамическом массиве, состоящем из  $n$  элементов, вычислить сумму элементов массива, расположенных между первым и последним нулевыми элементами.

**Вариант 11.** В одномерном динамическом массиве, состоящем из  $n$  элементов, вычислить сумму модулей элементов массива, расположенных после первого элемента, равного нулю.

**Вариант 12.** В одномерном динамическом массиве, состоящем из  $n$  элементов, вычислить сумму положительных элементов массива, расположенных до максимального элемента.

**Вариант 13.** В одномерном динамическом массиве, состоящем из  $n$  элементов, вычислить сумму элементов массива, расположенных до минимального элемента.

**Вариант 14.** В одномерном динамическом массиве, состоящем из  $n$  элементов, вычислить сумму элементов массива, расположенных между первым и последним положительными элементами.

**Вариант 15.** В одномерном динамическом массиве, состоящем из  $n$  элементов, вычислить среднее значение элементов, расположенных в массиве между первым последним нулевыми элементами.

**Вариант 16.** В одномерном динамическом массиве, состоящем из  $n$  элементов, вычислить произведение элементов массива, расположенных между первым и вторым нулевыми элементами.

**Вариант 17.** В одномерном динамическом массиве, состоящем из  $n$  элементов, определить номер минимального и максимального элементов массива.

**Вариант 18.** В одномерном динамическом массиве, состоящем из  $n$  элементов, определить сумму модулей элементов массива, расположенных после первого отрицательного элемента.

**Вариант 19.** В одномерном динамическом массиве, состоящем из  $n$  элементов, определить количество элементов массива, больших  $C$ .

**Вариант 20.** В одномерном динамическом массиве, состоящем из  $n$  элементов, определить количество элементов массива, меньших  $C$ .

**Вариант 21.** В одномерном динамическом массиве, состоящем из  $n$  элементов, вычислить сумму отрицательных элементов массива.

**Вариант 22.** В одномерном динамическом массиве, состоящем из  $n$  элементов, вычислить сумму положительных элементов массива.

**Вариант 23.** В одномерном динамическом массиве, состоящем из  $n$  элементов, вычислить произведение элементов массива с четными номерами.

**Вариант 24.** В одномерном динамическом массиве, состоящем из  $n$  элементов, вычислить сумму элементов массива, расположенных после минимального элемента.

**Вариант 25.** В одномерном динамическом массиве, состоящем из  $n$  элементов, вычислить сумму элементов массива, расположенных до максимального элемента.

## Методические указания

1 Разработать алгоритм программы.

2 Разработать визуальное приложение, осуществляющее тестирование программы методом базового пути:

- построение потокового графа программы;
- определение цикломатической сложности потокового графа;
- построение базового множества независимых линейных путей;
- составление тестовых вариантов;
- представление таблиц результатов тестирования эталонной и проверяемой программ.

3 Оформить отчет по лабораторной работе, включающий разделы:

- Постановка задачи.
- Блок-схема программы.
- Программный код с нумерованными узлами потокового графа.
- Управляющий потоковый граф программы.
- Цикломатическая сложность потокового графа.
- Базовое множество независимых линейных путей.
- Тестовые варианты, инициирующие выполнение каждого пути.
- Скриншоты программы.
- Код программы (функции обработчиков событий).
- Выводы.

## Контрольные вопросы

1 Что называется управляющим потоковым графом программы?

2 Какие модели управляющего потокового графа?

3 Что такое путь, ветвь, узел, дуга управляющего потокового графа?

4 Что называется операторным узлом управляющего потокового графа?

5 Что называется предикатным узлом управляющего потокового графа?

6 Что называется цикломатической сложностью управляющего потокового графа?

7 Как рассчитывается цикломатическая сложность управляющего потокового графа?

## 3 Функциональное тестирование программ

Функциональное тестирование – процесс обнаружения ошибок в функциях программы на всей области определения. Методы функционального тестирования:

- эквивалентное разбиение (классы эквивалентности);
- анализ граничных значений;
- метод функциональных диаграмм;



- предположение об ошибке.

### 3.1 Функциональные критерии

Функциональный критерий обеспечивает контроль степени выполнения требований заказчика в программном продукте. Функциональные критерии подразделяются на виды:

- тестирование проектов спецификации – набор тестов, обеспечивающий проверку каждого тестируемого пункта не менее одного раза;
- тестирование классов входных данных – набор тестов, обеспечивающий проверку представителя каждого класса входных данных не менее одного раза;
- тестирование правил — набор тестов, обеспечивающий проверку каждого правила, если входные и выходные значения описываются набором правил грамматики;
- тестирование классов выходных данных – набор тестов, обеспечивающий проверку представителя каждого выходного класса при условии, что выходные результаты заранее расклассифицированы, отдельные классы результатов учитывают ограничения на ресурсы или на время (time out);
- тестирование функций – набор тестов, обеспечивающий проверку каждого действия, реализуемого тестируемым модулем, не менее одного раза;
- комбинированные критерии для программ и спецификаций – набор тестов, обеспечивающий проверку всех комбинаций непротиворечивых условий программ и спецификаций не менее одного раза [4; 8; 9; 10].

### 3.2 Лабораторная работа № 4

#### Тестирование программ методами «чёрного ящика».

#### Дерево решений

**Цель:** закрепить теоретические знания и получить практические навыки в разработке программы тестирования методом эквивалентного разбиения классов.

**Используемые приемы и технологии:** технология визуального проектирования и событийного программирования, среда программирования Visual Studio 2010 Professional, язык программирования Visual C++ 2010 Professional.

**Ключевые термины:** класс эквивалентности, правильный класс, неправильный класс, предусловия, постусловия, дерево разбиения области данных, анализ граничных значений.

## Варианты заданий

Протестировать методом классов эквивалентности с построением дерева разбиения области данных программу, формализующую алгоритм варианта задачи.

**Вариант 1.** Решить алгебраическое уравнение 2-й степени (квадратное уравнение)  $a * x^2 + b * x + c = 0$ .

**Вариант 2.** Решить алгебраическое уравнение 3-й степени (кубическое уравнение)  $a * x^3 + b * x^2 + c * x + d = 0$ . Корни приведенного уравнения рассчитать по формулам Кардано.

**Вариант 3.** Решить алгебраическое уравнение 3-й степени (кубическое уравнение)  $a * x^3 + b * x^2 + c * x + d = 0$ . Корни рассчитать по тригонометрической формуле Виета.

**Вариант 4.** Решить биквадратное уравнение  $a * x^4 + b * x^2 + c = 0$ .

**Вариант 5.** В одномерном динамическом массиве, состоящем из  $n$  элементов, выполнить поиск элемента по заданному ключу последовательным методом поиска.

**Вариант 6.** В одномерном динамическом массиве, состоящем из  $n$  элементов, выполнить поиск элемента по заданному ключу бинарным методом поиска.

**Вариант 7.** В одномерном динамическом массиве, состоящем из  $n$  элементов, выполнить поиск элемента по заданному ключу интерполяционным методом поиска.

**Вариант 8.** В одномерном динамическом массиве, состоящем из  $n$  элементов, определить количество и индексы элементов массива, больших  $C$ .

**Вариант 9.** В одномерном динамическом массиве, состоящем из  $n$  элементов, определить количество и индексы элементов массива, меньших  $C$ .

**Вариант 10.** В одномерном динамическом массиве, состоящем из  $n$  элементов, определить количество и индексы элементов массива, больших  $A$  и меньших  $B$  ( $A < B$ ).

**Вариант 11.** В одномерном динамическом массиве, состоящем из  $n$  элементов, определить количество и индексы элементов массива, меньших  $A$  и больших  $B$  ( $A < B$ ).

**Вариант 12.** Дано действительное положительное число  $\varepsilon$ . Методом деления отрезка пополам найти приближённое значение корня уравнения  $x + \ln(x + 0,5) - 0,5 = 0$ ,  $[0; 2]$ . Абсолютная погрешность найденного значения не должна превосходить  $\varepsilon$ . В квадратных скобках указан отрезок, содержащий корень.

**Вариант 13.** Дано действительное положительное число  $\varepsilon$ . Методом касательных вычислить с точностью  $\varepsilon$  корень уравнения  $x^3 - 2 * x^2 + x - 3 = 0$ ,  $(2,2)$ . В круглых скобках указано начальное приближение к корню.

**Вариант 14.** Дано действительное положительное число  $\varepsilon$ . Методом хорд вычислить с точностью  $\varepsilon$  корень уравнения  $x * 2^x - 1 = 0$ ,  $[0; 1]$ . В квадратных скобках указан отрезок, содержащий корень.

**Вариант 15.** Дано действительное положительное число  $\varepsilon$ . Методом итераций вычислить с точностью  $\varepsilon$  корень уравнения  $x - \frac{\sin x}{2} - 1 = 0$ ,  $(0)$ . В круглых скобках указано начальное приближение к корню.

**Вариант 16.** Дано действительное положительное число  $\varepsilon$ . Методом деления отрезка пополам найти приближённое значение корня уравнения  $x^5 - x - 0,2 = 0$ ,  $[1; 1,1]$ . Абсолютная погрешность найденного значения не должна превосходить  $\varepsilon$ . В квадратных скобках указан отрезок, содержащий корень.

**Вариант 17.** Дано действительное положительное число  $\varepsilon$ . Методом касательных вычислить с точностью  $\varepsilon$  корень уравнения  $\operatorname{tg} x - x = 0$ ,  $(4,67)$ . В круглых скобках указано начальное приближение к корню.

**Вариант 18.** Дано действительное положительное число  $\varepsilon$ . Методом хорд вычислить с точностью  $\varepsilon$  корень уравнения  $x^2 - \sin 5 * x = 0$ ,  $[0,5; 0,6]$ . В квадратных скобках указан отрезок, содержащий корень.

**Вариант 19.** Дано действительное положительное число  $\varepsilon$ . Методом итераций вычислить с точностью  $\varepsilon$  корень уравнения  $2 * x^3 + 4 * x - 1 = 0$ ,  $(0,11)$ . В круглых скобках указано начальное приближение к корню.

**Вариант 20.** Дано действительное положительное число  $\varepsilon$ . Методом деления отрезка пополам найти приближённое значение корня уравнения  $x^4 + 2 * x^3 - x - 1 = 0$ ,  $[0; 1]$ . Абсолютная погрешность найденного значения не должна превосходить  $\varepsilon$ . В квадратных скобках указан отрезок, содержащий корень.

**Вариант 21.** Дано действительное положительное число  $\varepsilon$ . Методом касательных вычислить с точностью  $\varepsilon$  корень уравнения  $1,8 * x^4 - \sin 10 * x = 0$ ,  $(0,22)$ . В круглых скобках указано начальное приближение к корню.

**Вариант 22.** Дано действительное положительное число  $\varepsilon$ . Методом хорд вычислить с точностью  $\varepsilon$  корень уравнения  $\frac{2 * \sin^2 2 * x}{3} - \frac{3 * \cos^2 2 * x}{4} = 0$ ,  $[0; \pi/4]$ . В квадратных скобках указан отрезок, содержащий корень.

**Вариант 23.** Дано действительное положительное число  $\varepsilon$ . Методом итераций вычислить с точностью  $\varepsilon$  корень уравнения  $x^3 + 12 * x - 2 = 0$ ,  $(0,95)$ . В круглых скобках указано начальное приближение к корню.

**Вариант 24.** Дано действительное положительное число  $\varepsilon$ . Методом деления отрезка пополам найти приближённое значение корня уравнения

$x^3 - 0,2 * x^2 - 0,2 * x - 1,2 = 0$ ,  $[1,0; 1,5]$ . Абсолютная погрешность найденного значения не должна превосходить  $\varepsilon$ . В квадратных скобках указан отрезок, содержащий корень.

**Вариант 25.** Дано действительное положительное число  $\varepsilon$ . Методом касательных вычислить с точностью  $\varepsilon$  корень уравнения  $x^4 - 3 * x^2 + 75 * x - 10000 = 0$ ,  $(-11)$ . В круглых скобках указано начальное приближение к корню.

### Методические указания

1 Разработать алгоритм программы.

2 Разработать визуальное приложение, осуществляющее тестирование программы методом классов эквивалентности:

- построение дерева области разбиения данных;
- определение количества тестовых вариантов (количество листьев дерева);
- генерация тестовых вариантов;
- представление таблиц результатов тестирования эталонной и проверяемой программ.

3 Оформить отчет по лабораторной работе, включающий разделы:

- Постановка задачи.
- Блок-схема программы.
- Дерево разбиения области данных с пронумерованными листьями.
- Таблицы результатов тестирования.
- Скриншоты программы.
- Код программы (функции обработчиков событий).
- Выводы.

### Контрольные вопросы

- 1 Что называется функциональным тестированием?
- 2 Что называется классом эквивалентности?
- 3 Какие правила формирования классов эквивалентности?
- 4 Какие этапы включает алгоритм метода эквивалентного разбиения классов?
- 5 В чём сущность метода анализ граничных значений?
- 6 Какие правила метода анализ граничных значений?
- 7 Что называется деревом разбиения области данных?
- 8 Что такое предусловия и постусловия?

### 3.3 Лабораторная работа № 5

#### Тестирование программ методами «чёрного ящика».

#### Диаграмма причинно-следственных связей»

**Цель:** закрепить теоретические знания и получить практические навыки в разработке программы тестирования методом функциональных диаграмм.

**Используемые приемы и технологии:** технология визуального проектирования и событийного программирования, среда программирования Visual Studio 2010 Professional, язык программирования Visual C++ 2010 Professional.

**Ключевые термины:** диаграмма причинно-следственных связей, функции, ограничения, таблица решений с ограниченным входом, причина, следствие,

#### Варианты заданий

**Вариант 1.** Решить биквадратное уравнение  $a * x^4 + b * x^2 + c = 0$ .

**Вариант 2.** Решить алгебраическое уравнение 3-й степени (кубическое уравнение)  $a * x^3 + b * x^2 + c * x + d = 0$ . Корни рассчитать по тригонометрической формуле Виета.

**Вариант 3.** Решить алгебраическое уравнение 3-й степени (кубическое уравнение)  $a * x^3 + b * x^2 + c * x + d = 0$ . Корни приведенного уравнения рассчитать по формулам Кардано.

**Вариант 4.** Решить алгебраическое уравнение 2-й степени (квадратное уравнение)  $a * x^2 + b * x + c = 0$ .

**Вариант 5.** Дано действительное положительное число  $\varepsilon$ . Методом касательных вычислить с точностью  $\varepsilon$  корень уравнения  $x^4 - 3 * x^2 + 75 * x - 10000 = 0$ , (-11). В круглых скобках указано начальное приближение к корню.

**Вариант 6.** Дано действительное положительное число  $\varepsilon$ . Методом деления отрезка пополам найти приближённое значение корня уравнения  $x^3 - 0,2 * x^2 - 0,2 * x - 1,2 = 0$ , [1,0; 1,5]. Абсолютная погрешность найденного значения не должна превосходить  $\varepsilon$ . В квадратных скобках указан отрезок, содержащий корень.

**Вариант 7.** Дано действительное положительное число  $\varepsilon$ . Методом итераций вычислить с точностью  $\varepsilon$  корень уравнения  $x^3 + 12 * x - 2 = 0$ , (0,95). В круглых скобках указано начальное приближение к корню.

**Вариант 8.** Дано действительное положительное число  $\varepsilon$ . Методом хорд вычислить с точностью  $\varepsilon$  корень уравнения  $\frac{2 * \sin^2 2 * x}{3} - \frac{3 * \cos^2 2 * x}{4} = 0$ , [0;  $\pi/4$ ]. В квадратных скобках указан отрезок, содержащий корень.

**Вариант 9.** Дано действительное положительное число  $\varepsilon$ . Методом касательных вычислить с точностью  $\varepsilon$  корень уравнения  $1,8 * x^4 - \sin 10 * x = 0$ , (0,22). В круглых скобках указано начальное приближение к корню.

**Вариант 10.** Дано действительное положительное число  $\varepsilon$ . Методом деления отрезка пополам найти приближённое значение корня уравнения  $x^4 + 2 * x^3 - x - 1 = 0$ , [0; 1]. Абсолютная погрешность найденного значения не должна превосходить  $\varepsilon$ . В квадратных скобках указан отрезок, содержащий корень.

**Вариант 11.** Дано действительное положительное число  $\varepsilon$ . Методом итераций вычислить с точностью  $\varepsilon$  корень уравнения  $2 * x^3 + 4 * x - 1 = 0$ , (0,11). В круглых скобках указано начальное приближение к корню.

**Вариант 12.** Дано действительное положительное число  $\varepsilon$ . Методом хорд вычислить с точностью  $\varepsilon$  корень уравнения  $x^2 - \sin 5 * x = 0$ , [0,5; 0,6]. В квадратных скобках указан отрезок, содержащий корень.

**Вариант 13.** Дано действительное положительное число  $\varepsilon$ . Методом касательных вычислить с точностью  $\varepsilon$  корень уравнения  $\operatorname{tg} x - x = 0$ , (4,67). В круглых скобках указано начальное приближение к корню.

**Вариант 14.** Дано действительное положительное число  $\varepsilon$ . Методом деления отрезка пополам найти приближённое значение корня уравнения  $x^5 - x - 0,2 = 0$ , [1; 1,1]. Абсолютная погрешность найденного значения не должна превосходить  $\varepsilon$ . В квадратных скобках указан отрезок, содержащий корень.

**Вариант 15.** Дано действительное положительное число  $\varepsilon$ . Методом деления отрезка пополам найти приближённое значение корня уравнения  $x + \ln(x + 0,5) - 0,5 = 0$ , [0; 2]. Абсолютная погрешность найденного значения не должна превосходить  $\varepsilon$ . В квадратных скобках указан отрезок, содержащий корень.

**Вариант 16.** Дано действительное положительное число  $\varepsilon$ . Методом касательных вычислить с точностью  $\varepsilon$  корень уравнения  $x^3 - 2 * x^2 + x - 3 = 0$ , (2,2). В круглых скобках указано начальное приближение к корню.

**Вариант 17.** Дано действительное положительное число  $\varepsilon$ . Методом хорд вычислить с точностью  $\varepsilon$  корень уравнения  $x * 2^x - 1 = 0$ , [0; 1]. В квадратных скобках указан отрезок, содержащий корень.

**Вариант 18.** Дано действительное положительное число  $\varepsilon$ . Методом итераций вычислить с точностью  $\varepsilon$  корень уравнения  $x - \frac{\sin x}{2} - 1 = 0$ , (0). В круглых скобках указано начальное приближение к корню.

**Вариант 19.** В одномерном динамическом массиве, состоящем из  $n$  элементов, выполнить поиск элемента по заданному ключу последовательным методом поиска.

**Вариант 20.** В одномерном динамическом массиве, состоящем из  $n$  элементов, выполнить поиск элемента по заданному ключу бинарным методом поиска.

**Вариант 21.** В одномерном динамическом массиве, состоящем из  $n$  элементов, выполнить поиск элемента по заданному ключу интерполяционным методом поиска.

**Вариант 22.** В одномерном динамическом массиве, состоящем из  $n$  элементов, определить количество и индексы элементов массива, больших  $C$ .

**Вариант 23.** В одномерном динамическом массиве, состоящем из  $n$  элементов, определить количество и индексы элементов массива, меньших  $C$ .

**Вариант 24.** В одномерном динамическом массиве, состоящем из  $n$  элементов, определить количество и индексы элементов массива, больших  $A$  и меньших  $B$  ( $A < B$ ).

**Вариант 25.** В одномерном динамическом массиве, состоящем из  $n$  элементов, определить количество и индексы элементов массива, меньших  $A$  и больших  $B$  ( $A < B$ ).

### Методические указания

1 Разработать алгоритм программы.

2 Разработать визуальное приложение, осуществляющее тестирование программы методом диаграмм причинно-следственных связей:

- построение функциональной диаграммы;
- построение таблицы решений;
- определение количества тестовых вариантов;
- генерация тестовых вариантов;
- представление таблиц результатов тестирования.

3 Оформить отчет по лабораторной работе, включающий разделы:

- Постановка задачи.
- Блок-схема программы.
- Разбиение внешней спецификации на отдельные функции, комбинаторные свойства которых должны тестироваться.
  - Анализ спецификации в поисках условий на входе и действий на выходе.
  - Построение функциональной диаграммы.
  - Преобразование функциональной диаграммы в таблицу решений с ограниченным входом.
  - Таблицы результатов тестирования.
  - Скриншоты программы.
  - Код программы (функции обработчиков событий).
  - Выводы.

## Контрольные вопросы

- 1 Что называется функциональной диаграммой?
- 2 Для чего предназначены функции диаграммы причинно-следственных связей? Сколько функций? Какие функции?
- 3 Для чего предназначены ограничения диаграммы причинно-следственных связей? Сколько ограничений? Какие ограничения?
- 4 Для чего создается таблица решений с ограниченным входом?
- 5 Какие этапы метода функциональных диаграмм?
- 6 Какие преимущества и недостатки метода причинно-следственных диаграмм?

## 4 Интеграционное тестирование

Интеграционное тестирование – тестирование программного обеспечения, на этапе сборки модулей в единый комплекс. Цель интеграционного тестирования — нахождение проблем взаимодействия модулей (компонент, подсистем). Тестирование выполняется через интерфейс модулей с использованием метода «чёрного ящика» [6].

### 4.1 Методы интеграционного тестирования

Существуют два метода сборки модулей:

- монолитный, характеризующийся одновременным объединением модулей в тестируемый комплекс;
- инкрементальный, характеризующийся пошаговым (помодульным) наращиванием комплекса программ с пошаговым тестированием собираемого комплекса [4].

В инкрементальном методе выделяют две стратегии добавления модулей:

- восходящее тестирование;
- нисходящее тестирование.

При восходящем тестировании сначала тестируются терминальные модули, независимые от других модулей, затем тестируются модули, которые зависят от проверенных модулей. Для модуля разрабатывают драйвер.

Драйвер – программный модуль, эмулирующий окружение модуля при тестировании. Драйвер вызывает тестируемый модуль, передает тестовые данные, проверяет результаты работы модуля и корректность реализации его интерфейса. При восходящем тестировании упрощается локализация ошибок.

При нисходящем тестировании сначала тестируют верхний управляющий модуль программной системы без модулей низкого уровня, вместо которых используют заглушки, затем тестируют модули более низкого уровня.

Заглушка – программный модуль, обладающий тем же интерфейсом, что и замещаемый модуль нижележащего уровня. Заглушка не реализует



функциональность замещаемого модуля, а возвращает значения, позволяющие проверить функционирование тестируемого модуля [6].

## 4.2 Лабораторная работа № 6 Нисходящее тестирование

**Цель:** закрепить теоретические знания и получить практические навыки в разработке программы тестирования методом нисходящего тестирования.

**Используемые приемы и технологии:** технология визуального проектирования и событийного программирования, среда программирования Visual Studio 2010 Professional, язык программирования Visual C++ 2010 Professional.

**Ключевые термины:** интеграционное тестирование, модуль-заглушка, категории заглушек, монолитная сборка модулей, пошаговая сборка модулей.

### Варианты заданий

**Вариант 1.** В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) сумму отрицательных элементов массива;
- 2) произведение элементов массива, расположенных между максимальным и минимальным элементами.

Упорядочить элементы массива по возрастанию.

**Вариант 2.** В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) сумму положительных элементов массива;
- 2) произведение элементов массива, расположенных между максимальным по модулю и минимальным по модулю элементами.

Упорядочить элементы массива по убыванию.

**Вариант 3.** В одномерном массиве, состоящем из  $n$  целых элементов, вычислить:

- 1) произведение элементов массива с четными номерами;
- 2) сумму элементов массива, расположенных между первым и последним нулевыми элементами.

Преобразовать массив таким образом, чтобы сначала располагались все положительные элементы, а потом – все отрицательные (элементы, равные нулю, считать положительными).

**Вариант 4.** В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) сумму элементов массива с нечетными номерами;
- 2) сумму элементов массива, расположенных между первым и последним отрицательными элементами.

Сжать массив, удалив из него все элементы, модуль которых не превышает 1. Освободившиеся в конце массива элементы заполнить нулями.

**Вариант 5.** В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) максимальный элемент массива;
- 2) сумму элементов массива, расположенных до последнего положительного элемента.

Сжать массив, удалив из него все элементы, модуль которых находится в интервале  $[a, b]$ . Освободившиеся в конце массива элементы заполнить нулями.

**Вариант 6.** В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) минимальный элемент массива;
- 2) сумму элементов массива, расположенных между первым и последним положительными элементами.

Преобразовать массив таким образом, чтобы сначала располагались все элементы, равные нулю, а потом – все остальные.

**Вариант 7.** В одномерном массиве, состоящем из  $n$  целых элементов, вычислить:

- 1) номер максимального элемента массива;
- 2) произведение элементов массива, расположенных между первым и вторым нулевыми элементами.

Преобразовать массив таким образом, чтобы в первой его половине располагались элементы, стоявшие в нечетных позициях, а во второй – элементы, стоявшие в четных позициях.

**Вариант 8.** В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) номер минимального элемента массива;
- 2) сумму элементов массива, расположенных между первым и вторым отрицательными элементами.

Преобразовать массив таким образом, чтобы сначала располагались все элементы, модуль которых не превышает 1, а потом – все остальные.

**Вариант 9.** В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) максимальный по модулю элемент массива;
- 2) сумму элементов массива, расположенных между первым и вторым положительными элементами.

Преобразовать массив таким образом, чтобы элементы, равные нулю, располагались после всех остальных.

**Вариант 10.** В одномерном массиве, состоящем из  $n$  целых элементов, вычислить:

- 1) минимальный по модулю элемент массива;
- 2) сумму модулей элементов массива, расположенных после первого элемента, равного нулю.

Преобразовать массив таким образом, чтобы в первой его половине располагались элементы, стоявшие в четных позициях, а во второй половине – элементы, стоявшие в нечетных позициях.

**Вариант 11.** В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) номер минимального по модулю элемента массива;
- 2) сумму модулей элементов массива, расположенных после первого отрицательного элемента.

Сжать массив, удалив из него все элементы, величина которых находится в интервале  $[a,b]$ . Освободившиеся в конце массива элементы заполнить нулями.

**Вариант 12.** В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) номер максимального по модулю элемента массива;
- 2) сумму элементов массива, расположенных после первого положительного элемента.

Преобразовать массив таким образом, чтобы сначала располагались все элементы, целая часть которых лежит в интервале  $[a,b]$ , а потом – все остальные.

**Вариант 13.** В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) количество элементов массива, лежащих в диапазоне от  $A$  до  $B$ ;
- 2) сумму элементов массива, расположенных после максимального элемента.

Упорядочить элементы массива по убыванию модулей элементов.

**Вариант 14.** В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) количество элементов массива, равных нулю;
- 2) сумму элементов массива, расположенных после минимального элемента.

Упорядочить элементы массива по возрастанию модулей элементов.

**Вариант 15.** В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) количество элементов массива, больших  $C$ ;
- 2) произведение элементов массива, расположенных после максимального по модулю элемента.

Преобразовать массив таким образом, чтобы сначала располагались все отрицательные элементы, а потом — все положительные (элементы, равные нулю, считать положительными).

**Вариант 16.** В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) количество отрицательных элементов массива;
- 2) сумму модулей элементов массива, расположенных после минимального по модулю элемента.

Заменить все отрицательные элементы массива их квадратами и упорядочить элементы массива по возрастанию.

**Вариант 17.** В одномерном массиве, состоящем из  $n$  целых элементов, вычислить:

- 1) количество положительных элементов массива;
- 2) сумму элементов массива, расположенных после последнего элемента, равного нулю.

Преобразовать массив таким образом, чтобы сначала располагались все элементы, целая часть которых не превышает 1, а потом – все остальные.

**Вариант 18.** В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) количество элементов массива, меньших  $C$ ;
- 2) сумму целых частей элементов массива, расположенных после последнего отрицательного элемента.

Преобразовать массив таким образом, чтобы сначала располагались все элементы, отличающиеся от максимального не более чем на 20%, а потом – все остальные.

**Вариант 19.** В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) произведение отрицательных элементов массива;
- 2) сумму положительных элементов массива, расположенных до максимального элемента.

Изменить порядок следования элементов в массиве на обратный.

**Вариант 20.** В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) произведение положительных элементов массива;
- 2) сумму элементов массива, расположенных до минимального элемента.

Упорядочить по возрастанию отдельно элементы, стоящие на четных местах, и элементы, стоящие на нечетных местах.

**Вариант 21.** В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) среднее арифметическое элементов, расположенных до первого и после последнего нулевых значений;
- 2) поменять местами первый и максимальный элементы, последний и минимальный элементы.

Упорядочить по убыванию отдельно элементы, стоящие на четных местах, и элементы, стоящие на нечетных местах.

**Вариант 22.** В одномерном массиве, состоящем из  $n$  целых элементов, вычислить:

- 1) произведение элементов, находящихся между минимальным и максимальным элементами массива;
- 2) удалить из массива простое число.

Упорядочить по возрастанию элементы массива.

**Вариант 23.** В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) произведение отрицательных элементов массива;

2) сумму элементов массива, расположенных до максимального элемента.

Упорядочить по возрастанию отдельно элементы, стоящие на четных местах, и элементы, стоящие на нечетных местах.

**Вариант 24.** В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

1) среднее значение элементов, расположенных в массиве между первым последним нулевыми элементами;

2) поменять местами максимальный и минимальный элементы.

Упорядочить по возрастанию отдельно элементы, стоящие на четных местах, и элементы, стоящие на нечетных местах.

**Вариант 25.** В одномерном массиве, состоящем из  $n$  целых элементов, определить:

1) количество совершенных чисел;

2) удалить из массива все нулевые элементы.

Упорядочить по убыванию отдельно элементы, стоящие на четных местах, и элементы, стоящие на нечетных местах.

### Методические указания

1 Разработать алгоритм программы.

2 Разработать приложение, осуществляющее тестирование программы методом нисходящего тестирования:

- построение модулей-заглушек;
- тестирование верхнего головного модуля программы;
- тестирование модулей программы;
- представление результатов тестирования.

3 Оформить отчет по лабораторной работе, включающий разделы:

- Постановка задачи.
- Блок-схема программы.
- Структурная схема комплекса программы
- Таблицы результатов тестирования.
- Скриншоты программы.
- Код программы.
- Выводы.

### Контрольные вопросы

1 Что называется интеграционным тестированием?

2 Что такое модуль-заглушка?

3 Какие виды модулей-заглушек?

4 Какие этапы алгоритма метода нисходящего тестирования?

5 Какие преимущества и недостатки метода нисходящего тестирования?

## 4.2 Лабораторная работа № 7

### Восходящее тестирование

**Цель:** закрепить теоретические знания и получить практические навыки в разработке программы тестирования методом восходящего тестирования.

**Используемые приемы и технологии:** технология визуального проектирования и событийного программирования, среда программирования Visual Studio 2010 Professional, язык программирования Visual C++ 2010 Professional.

**Ключевые термины:** интеграционное тестирование, модуль-драйвер, категории модулей-драйверов, монолитная сборка модулей, пошаговая сборка модулей.

### Варианты заданий

**Вариант 1.** В одномерном динамическом массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) количество элементов массива, меньших  $C$ ;
- 2) сумму целых частей элементов массива, расположенных после последнего отрицательного элемента.

Преобразовать массив таким образом, чтобы сначала располагались все элементы, отличающиеся от максимального не более чем на 20%, а потом – все остальные.

**Вариант 2.** В одномерном динамическом массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) произведение отрицательных элементов массива;
- 2) сумму положительных элементов массива, расположенных до максимального элемента.

Изменить порядок следования элементов в массиве на обратный.

**Вариант 3.** В одномерном динамическом массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) произведение положительных элементов массива;
- 2) сумму элементов массива, расположенных до минимального элемента.

Упорядочить по возрастанию отдельно элементы, стоящие на четных местах, и элементы, стоящие на нечетных местах.

**Вариант 4.** В одномерном динамическом массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) среднее арифметическое элементов, расположенных до первого и после последнего нулевых значений;
- 2) поменять местами первый и максимальный элементы, последний и минимальный элементы.

Упорядочить по убыванию отдельно элементы, стоящие на четных местах, и элементы, стоящие на нечетных местах.

**Вариант 5.** В одномерном динамическом массиве, состоящем из  $n$  целых элементов, вычислить:

1) произведение элементов, находящихся между минимальным и максимальным элементами массива;

2) удалить из массива простое число.

Упорядочить по возрастанию элементы массива.

**Вариант 6.** В одномерном динамическом массиве, состоящем из  $n$  вещественных элементов, вычислить:

1) произведение отрицательных элементов массива;

2) сумму элементов массива, расположенных до максимального элемента.

Упорядочить по возрастанию отдельно элементы, стоящие на четных местах, и элементы, стоящие на нечетных местах.

**Вариант 7.** В одномерном динамическом массиве, состоящем из  $n$  вещественных элементов, вычислить:

1) среднее значение элементов, расположенных в массиве между первым последним нулевыми элементами;

2) поменять местами максимальный и минимальный элементы.

Упорядочить по возрастанию отдельно элементы, стоящие на четных местах, и элементы, стоящие на нечетных местах.

**Вариант 8.** В одномерном динамическом массиве, состоящем из  $n$  целых элементов, определить:

1) количество совершенных чисел;

2) удалить из массива все нулевые элементы.

Упорядочить по убыванию отдельно элементы, стоящие на четных местах, и элементы, стоящие на нечетных местах.

**Вариант 9.** В одномерном динамическом массиве, состоящем из  $n$  вещественных элементов, вычислить:

1) сумму отрицательных элементов массива;

2) произведение элементов массива, расположенных между максимальным и минимальным элементами.

Упорядочить элементы массива по возрастанию.

**Вариант 10.** В одномерном динамическом массиве, состоящем из  $n$  вещественных элементов, вычислить:

1) сумму положительных элементов массива;

2) произведение элементов массива, расположенных между максимальным по модулю и минимальным по модулю элементами.

Упорядочить элементы массива по убыванию.

**Вариант 11.** В одномерном динамическом массиве, состоящем из  $n$  целых элементов, вычислить:

1) произведение элементов массива с четными номерами;

2) сумму элементов массива, расположенных между первым и последним нулевыми элементами.

Преобразовать массив таким образом, чтобы сначала располагались все положительные элементы, а потом – все отрицательные (элементы, равные нулю, считать положительными).

**Вариант 12.** В одномерном динамическом массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) сумму элементов массива с нечетными номерами;
- 2) сумму элементов массива, расположенных между первым и последним отрицательными элементами.

Сжать массив, удалив из него все элементы, модуль которых не превышает 1. Освободившиеся в конце массива элементы заполнить нулями.

**Вариант 13.** В одномерном динамическом массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) максимальный элемент массива;
- 2) сумму элементов массива, расположенных до последнего положительного элемента.

Сжать массив, удалив из него все элементы, модуль которых находится в интервале  $[a, b]$ . Освободившиеся в конце массива элементы заполнить нулями.

**Вариант 14.** В одномерном динамическом массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) минимальный элемент массива;
- 2) сумму элементов массива, расположенных между первым и последним положительными элементами.

Преобразовать массив таким образом, чтобы сначала располагались все элементы, равные нулю, а потом – все остальные.

**Вариант 15.** В одномерном динамическом массиве, состоящем из  $n$  целых элементов, вычислить:

- 1) номер максимального элемента массива;
- 2) произведение элементов массива, расположенных между первым и вторым нулевыми элементами.

Преобразовать массив таким образом, чтобы в первой его половине располагались элементы, стоявшие в нечетных позициях, а во второй – элементы, стоявшие в четных позициях.

**Вариант 16.** В одномерном динамическом массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) номер минимального элемента массива;
- 2) сумму элементов массива, расположенных между первым и вторым отрицательными элементами.

Преобразовать массив таким образом, чтобы сначала располагались все элементы, модуль которых не превышает 1, а потом – все остальные.

**Вариант 17.** В одномерном динамическом массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) максимальный по модулю элемент массива;
- 2) сумму элементов массива, расположенных между первым и вторым положительными элементами.



Преобразовать массив таким образом, чтобы элементы, равные нулю, располагались после всех остальных.

**Вариант 18.** В одномерном динамическом массиве, состоящем из  $n$  целых элементов, вычислить:

- 1) минимальный по модулю элемент массива;
- 2) сумму модулей элементов массива, расположенных после первого элемента, равного нулю.

Преобразовать массив таким образом, чтобы в первой его половине располагались элементы, стоявшие в четных позициях, а во второй половине – элементы, стоявшие в нечетных позициях.

**Вариант 19.** В одномерном динамическом массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) номер минимального по модулю элемента массива;
- 2) сумму модулей элементов массива, расположенных после первого отрицательного элемента.

Сжать массив, удалив из него все элементы, величина которых находится в интервале  $[a, b]$ . Освободившиеся в конце массива элементы заполнить нулями.

**Вариант 20.** В одномерном динамическом массиве, состоящем из  $n$  целых элементов, вычислить:

- 1) количество положительных элементов массива;
- 2) сумму элементов массива, расположенных после последнего элемента, равного нулю.

Преобразовать массив таким образом, чтобы сначала располагались все элементы, целая часть которых не превышает 1, а потом – все остальные.

**Вариант 21.** В одномерном динамическом массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) количество отрицательных элементов массива;
- 2) сумму модулей элементов массива, расположенных после минимального по модулю элемента.

Заменить все отрицательные элементы массива их квадратами и упорядочить элементы массива по возрастанию.

**Вариант 22.** В одномерном динамическом массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) количество элементов массива, больших  $C$ ;
- 2) произведение элементов массива, расположенных после максимального по модулю элемента.

Преобразовать массив таким образом, чтобы сначала располагались все отрицательные элементы, а потом – все положительные (элементы, равные нулю, считать положительными).

**Вариант 23.** В одномерном динамическом массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) количество элементов массива, равных нулю;

2) сумму элементов массива, расположенных после минимального элемента.

Упорядочить элементы массива по возрастанию модулей элементов.

**Вариант 24.** В одномерном динамическом массиве, состоящем из  $n$  вещественных элементов, вычислить:

1) количество элементов массива, лежащих в диапазоне от  $A$  до  $B$ ;

2) сумму элементов массива, расположенных после максимального элемента.

Упорядочить элементы массива по убыванию модулей элементов.

**Вариант 25.** В одномерном динамическом массиве, состоящем из  $n$  вещественных элементов, вычислить:

1) номер максимального по модулю элемента массива;

2) сумму элементов массива, расположенных после первого положительного элемента.

Преобразовать массив таким образом, чтобы сначала располагались все элементы, целая часть которых лежит в интервале  $[a,b]$ , а потом – все остальные.

### Методические указания

1 Разработать алгоритм программы.

2 Разработать приложение, осуществляющее тестирование программы методом восходящего тестирования:

- построение модулей-драйверов;
- тестирование терминальных модулей;
- тестирование кластеров программы;
- представление результатов тестирования.

3 Оформить отчет по лабораторной работе, включающий разделы:

- Постановка задачи.
- Блок-схема программы.
- Структурная схема комплекса программы
- Таблицы результатов тестирования.
- Скриншоты программы.
- Код программы.
- Выводы.

### Контрольные вопросы

1 Что называется интеграционным тестированием?

2 Что такое модуль-драйвер?

3 Какие виды модулей-драйверов?

4 Какие этапы алгоритма метода восходящего тестирования?

5 Какие преимущества и недостатки метода восходящего тестирования?

## 5 Автоматизированное тестирование

Автоматизированное тестирование – тестирование, использующее программные средства для выполнения тестов и проверки корректности результатов выполнения.

Автоматизированное тестирование позволяет проводить повторный прогон тестов без участия человека [6].

### 5.1 Способы автоматизации тестирования

Автоматизированное тестирование включает два способа:

- тестирование на уровне кода;
- использование инструментальных средств (GUI (Graphical User Interface) -тестирование).

GUI-тестирование относится к тестированию методом «чёрного ящика».

Существуют четыре поколения инструментов и техник, предназначенных для организации GUI-тестирования:

1 Утилиты записи и воспроизведения (capture/playback tools). Они записывают действия во время ручного тестирования. В дальнейшем позволяют воспроизвести записанные действия без участия человека.

2 Сценарии (scripting) – форма автоматизации тестирования с использованием специализированных скриптовых языков. Язык должен поддерживать эмуляцию действий пользователя и получение результатов.

3 Data-driven testing – методология автоматизации тестирования, основанная на использовании в скриптах параметров выполнения тестов. Параметры, задающие логику работы тестов, находятся во внешнем хранилище. Эта методология автоматизированного тестирования позволяет организовать выполнение сценариев с различными наборами входных параметров и повысить гибкость тестирования.

4 Специализированный словарь ключевых слов, описывающих системные события (keyword-based testing). С каждым ключевым словом связаны необходимые параметры и ожидаемые результаты. Данный подход позволяет писать функциональные тесты на отечественном языке, не требуя от тестировщиков навыков программирования [6; 11; 12; 13].

### 5.2 Лабораторная работа № 8 «Автоматизированное тестирование»

**Цель:** закрепить теоретические знания и получить практические навыки в применении автоматизированного тестирования.

**Используемые приемы и технологии:** фреймворк CppUnit, среда программирования Visual Studio 2010 Ultimate и Microsoft Test Professional 2010.

**Ключевые термины:** автоматизированное тестирование, фреймворк, Microsoft Test Manager, Team Foundation Server, Lab Management.

### Варианты заданий

**Вариант 1.** Дана целочисленная прямоугольная матрица  $\|a_{n*m}\|$ .

Написать программу, определяющую величины:

- 1) количество строк, не содержащих ни одного нулевого элемента;
- 2) максимальное значение из чисел, встречающихся в заданной матрице более одного раза;
- 3) суммы элементов строк и столбцов матрицы.

Суммы элементов строк и столбцов матрицы упорядочить по возрастанию методом обмена.

**Вариант 2.** Дана целочисленная прямоугольная матрица  $\|a_{n*m}\|$ .

Написать программу, определяющую величины:

- 1) количество столбцов, не содержащих ни одного нулевого элемента;
- 2) максимальное и минимальное значение элементов матрицы;
- 3) среднее арифметическое значение элементов строк и столбцов матрицы.

Средние арифметические значения элементов строк и столбцов матрицы упорядочить по возрастанию методом выбора.

**Вариант 3.** Дана целочисленная прямоугольная матрица  $\|a_{n*m}\|$ .

Написать программу, определяющую величины:

- 1) количество столбцов, содержащих хотя бы один нулевой элемент;
- 2) номер строки, в которой находится самая длинная серия одинаковых элементов;
- 3) среднее геометрическое значение элементов строк и столбцов матрицы.

Средние геометрические значения элементов строк и столбцов матрицы упорядочить по возрастанию методом вставки

**Вариант 4.** Дана целочисленная квадратная матрица  $\|a_{n*n}\|$ . Написать

программу, определяющую величины:

- 1) сумму элементов в строках, которые не содержат отрицательных элементов;
- 2) максимум среди элементов выше главной диагонали матрицы;
- 3) суммы четных элементов строк и столбцов матрицы.

Суммы четных элементов строк и столбцов матрицы упорядочить по убыванию методом обмена.

**Вариант 5.** Дана целочисленная квадратная матрица  $\|a_{n*n}\|$ . Написать

программу, определяющую величины:

- 1) суммы элементов в столбцах, которые не содержат отрицательных элементов;

2) минимум среди модулей элементов ниже побочной диагонали матрицы;

3) суммы нечетных элементов строк и столбцов матрицы.

Суммы нечетных элементов строк и столбцов матрицы упорядочить по убыванию методом выбора.

**Вариант 6.** Дана целочисленная прямоугольная матрица  $\|a_{n*m}\|$ .

Написать программу, определяющую величины:

1) суммы элементов в строках, которые содержат хотя бы один отрицательный элемент;

2) минимальный элемент матрицы;

3) среднее арифметическое значение четных элементов строк и столбцов матрицы.

Средние арифметические значения четных элементов строк и столбцов матрицы упорядочить по убыванию методом вставки.

**Вариант 7.** Дана целочисленная прямоугольная матрица  $\|a_{n*m}\|$ .

Написать программу, определяющую величины:

1) количество отрицательных элементов матрицы;

2) максимальный элемент матрицы;

3) суммы элементов матрицы, находящихся ниже главной диагонали.

Элементы матрицы, находящиеся выше главной диагонали, упорядочить по возрастанию методом обмена.

**Вариант 8.** Дана целочисленная прямоугольная матрица  $\|a_{n*m}\|$ .

Написать программу, определяющую величины:

1) количество положительных элементов матрицы;

2) максимальный и минимальный нечетные элементы строк матрицы;

3) суммы элементов матрицы, находящихся ниже побочной диагонали.

Элементы матрицы, находящиеся ниже побочной диагонали, упорядочить по возрастанию методом выбора.

**Вариант 9.** Дана целочисленная прямоугольная матрица  $\|a_{n*m}\|$ .

Написать программу, определяющую величины:

1) количество нулевых элементов матрицы;

2) максимальный и минимальный нечетные элементы столбцов матрицы;

3) суммы элементов матрицы, находящихся выше побочной диагонали.

Элементы матрицы, находящиеся выше побочной диагонали, упорядочить по возрастанию методом вставки.

**Вариант 10.** Дана целочисленная прямоугольная матрица  $\|a_{n*m}\|$ .

Написать программу, определяющую величины:

1) количество ненулевых элементов матрицы;

2) максимальный и минимальный четные элементы столбцов матрицы;

3) суммы элементов матрицы, находящихся ниже побочной диагонали.

Элементы матрицы, находящиеся ниже побочной диагонали, упорядочить по убыванию методом обмена.

**Вариант 11.** Дана целочисленная прямоугольная матрица  $\|a_n * m\|$ .

Написать программу, определяющую величины:

- 1) количество элементов матрицы, меньших величины В;
- 2) максимальный и минимальный нечетные элементы столбцов матрицы;
- 3) суммы элементов матрицы, находящихся выше побочной диагонали.

Элементы матрицы, находящиеся выше побочной диагонали, упорядочить по возрастанию методом вставки.

**Вариант 12.** Дана целочисленная прямоугольная матрица  $\|a_n * m\|$ .

Написать программу, определяющую величины:

- 1) количество элементов матрицы, больших величины В;
- 2) минимальный по модулю элемент матрицы;
- 3) суммы элементов, расположенных по периметру матрицы.

Элементы, расположенные по периметру матрицы, упорядочить по убыванию методом обмена.

**Вариант 13.** Дана целочисленная квадратная матрица  $\|a_n * n\|$ . Написать программу, определяющую величины:

- 1) количество элементов матрицы, находящихся в диапазоне от А до В;
- 2) максимальный по модулю элемент матрицы;
- 3) суммы элементов, расположенных на диагоналях матрицы.

Элементы, расположенные на диагоналях матрицы, упорядочить по убыванию методом выбора.

**Вариант 14.** Дана целочисленная прямоугольная матрица  $\|a_n * m\|$ .

Написать программу, определяющую величины:

- 1) количество элементов матрицы, содержащих простые числа;
- 2) минимум среди модулей элементов, расположенных ниже побочной диагонали;
- 3) средние квадратичные значения столбцов и строк матрицы.

Средние квадратичные значения столбцов и строк матрицы упорядочить по убыванию методом вставки.

**Вариант 15.** Дана целочисленная квадратная матрица  $\|a_n * n\|$ . Написать программу, определяющую величины:

- 1) количество элементов матрицы, содержащих простые числа;
- 2) максимум среди модулей элементов, расположенных выше главной диагонали;
- 3) средние квадратичные значения четных столбцов и нечетных строк матрицы.

Средние квадратичные значения четных столбцов и нечетных строк матрицы упорядочить по возрастанию методом обмена.

**Вариант 16.** Дана целочисленная квадратная матрица  $\|a_n * n\|$ . Написать программу, определяющую величины:

- 1) количество элементов матрицы, содержащих простые числа;
- 2) минимум среди элементов, расположенных выше побочной диагонали;

3) средние квадратичные значения нечетных столбцов и четных строк матрицы.

Средние квадратичные значения нечетных столбцов и четных строк матрицы упорядочить по возрастанию методом выбора.

**Вариант 17.** Дана целочисленная квадратная матрица  $\|a_n * n\|$ . Написать программу, определяющую величины:

- 1) количество элементов матрицы, содержащих простые числа;
- 2) максимум среди элементов побочной диагонали;
- 3) средние арифметические значения столбцов и строк матрицы.

Средние арифметические значения столбцов и строк матрицы упорядочить по возрастанию методом вставки.

**Вариант 18.** Дана целочисленная квадратная матрица  $\|a_n * n\|$ . Написать программу, определяющую величины:

- 1) количество элементов матрицы, содержащих простые числа;
- 2) минимум среди элементов главной диагонали;
- 3) средние арифметические значения четных столбцов и нечетных строк матрицы.

Средние арифметические значения четных столбцов и нечетных строк матрицы упорядочить по убыванию методом обмена.

**Вариант 19.** Дана целочисленная квадратная матрица  $\|a_n * n\|$ . Написать программу, определяющую величины:

- 1) количество элементов матрицы, содержащих нулевые значения;
- 2) максимум элементов, расположенных ниже побочной диагонали;
- 3) суммы положительных элементов столбцов и строк матрицы.

Суммы положительных элементов столбцов и строк матрицы упорядочить по убыванию методом выбора.

**Вариант 20.** Дана целочисленная квадратная матрица  $\|a_n * n\|$ . Написать программу, определяющую величины:

- 1) количество элементов матрицы, содержащих простые числа;
- 2) минимум элементов, расположенных выше побочной диагонали;
- 3) суммы отрицательных элементов столбцов и строк матрицы.

Суммы отрицательных элементов столбцов и строк матрицы упорядочить по убыванию методом вставки.

**Вариант 21.** Дана целочисленная квадратная матрица  $\|a_n * n\|$ . Написать программу, определяющую величины:

- 1) количество элементов матрицы, содержащих значения больше С;
- 2) минимум элементов побочной диагонали;
- 3) суммы элементов сторон и диагоналей элементов матрицы;

Суммы элементов сторон и диагоналей матрицы упорядочить по возрастанию методом обмена.

**Вариант 22.** Дана целочисленная квадратная матрица  $\|a_n * n\|$ . Написать программу, определяющую величины:

- 1) количество элементов матрицы, содержащих значения меньше С;

- 2) максимум элементов, расположенных выше главной диагонали;
- 3) средние квадратичные значения столбцов и строк матрицы.

Средние квадратичные значения столбцов и строк матрицы упорядочить по убыванию методом вставки.

**Вариант 23.** Дана целочисленная квадратная матрица  $\|a_n * n\|$ . Написать программу, определяющую величины:

- 1) количество элементов матрицы, содержащих четные числа;
- 2) максимум элементов, расположенных ниже главной диагонали;
- 3) суммы элементов, находящихся между минимальным и максимальным элементами строк.

Суммы элементов, находящихся между минимальным и максимальным элементами строк матрицы, упорядочить по возрастанию методом вставки.

**Вариант 24.** Дана целочисленная прямоугольная матрица  $\|a_n * m\|$ . Написать программу, определяющую величины:

- 1) количество элементов матрицы, содержащих нечетные числа;
- 2) максимум элементов, расположенных по периметру прямоугольной матрицы;
- 3) суммы элементов, находящихся между минимальным и максимальным элементами столбцов.

Суммы элементов, находящихся между минимальным и максимальным элементами столбцов матрицы, упорядочить по убыванию методом обмена.

**Вариант 25.** Дана целочисленная прямоугольная матрица  $\|a_n * m\|$ . Написать программу, определяющую величины:

- 1) количество отрицательных элементов матрицы;
- 2) максимум элементов, расположенных по периметру прямоугольной матрицы;
- 3) средние геометрические значения элементов строк и столбцов матрицы.

Средние геометрические значения элементов строк и столбцов матрицы упорядочить по возрастанию методом вставки.

### Методические указания

- 1 Разработать алгоритм программы.
- 2 Провести тестирование программы с помощью Visual Studio 2010.
- 3 Оформить отчет по лабораторной работе, включающий разделы:
  - Постановка задачи.
  - Блок-схема программы.
  - Этапы автоматизированного тестирования.
  - Результаты тестирования.
  - Скриншоты.
  - Код программы.
  - Выводы.



## Контрольные вопросы

- 1 Что называется автоматизированным тестированием?
- 2 Какие особенности тестирования с помощью CppUnit?
- 3 Какие этапы тестирования с помощью Visual Studio 2010?
- 4 Какие преимущества и недостатки автоматизированного тестирования?

## ЗАКЛЮЧЕНИЕ

Тестирование программного обеспечения – процесс исследования с целью получения информации о качестве программного продукта. Цель тестирования – выявление дефектов в программном обеспечении.

Тестирование – важный этап жизненного цикла программного обеспечения. Затраты на создание программ увеличиваются при возрастании требований к их качеству. При создании программного проекта около 50% общего времени и более 50% общей стоимости расходуется на тестирование разрабатываемой программы, поэтому требуются методы и средства, позволяющие повысить качество программ при небольших затратах труда.

В методическом указании приводятся задания к выполнению лабораторных работ по дисциплине «Управление качеством и тестирование программного обеспечения». Задания сопровождаются кратким теоретическим обоснованием. Выполнение лабораторных работ позволит приобрести практические навыки тестирования программных средств и закрепить теоретические знания, полученные студентами на лекциях по дисциплине «Управление качеством и тестирование программного обеспечения».

## СПИСОК ЛИТЕРАТУРЫ

- 1 Джеф Решка, Джон Пол, Элфрид Дастин. Тестирование программного обеспечения. Внедрение, управление и автоматизация. – Москва. : Лори, 2012. – 600 с.
- 2 Кори Сандлер, Том Баджет, Гленфорд Майерс. Искусство тестирования программ. – Москва. : Вильямс, 2012. – 272 с.
- 3 Канер Сэм, Фолк Джек. Тестирование программного обеспечения. – Москва. : ДиаСофт, 2001. – 538 с.
- 4 Котляров В. П. Основы тестирования программного обеспечения. – Москва. : Интернет Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2006. — 285 с.
- 5 Майерс Г. Дж. Надежность программного обеспечения. – Москва.: Мир, 1980. – 360 с.
- 6 Анашкина Н. В. Технологии и методы программирования. – Москва. : Издательский центр «Академия», 2012. – 384 с.
- 7 Макконнелл С. Совершенный код. – Санкт-Петербург : «Питер», 2005. – 896 с.

8 Бейзер Б. Тестирование черного ящика. – Санкт-Петербург : «Питер», 2005. – 318 с.

9 Брауде Э. Технология разработки программного обеспечения. – Санкт-Петербург : «Питер», 2004. – 655 с.

10 Степанченко И. В. Методы тестирования программного обеспечения : учебное пособие. – Волгоград : ВолгГТУ, 2006. – 74 с.

URL: <http://window.edu.ru/resource/765/45765> (дата обращения: 06.03.2017).

11 Левинсон Дж. Тестирование ПО с помощью Visual Studio 2010. – Москва. : ЭКОМ Паблшерз, 2012. – 336 с.

12 Винниченко С. И. Автоматизация процессов тестирования. – Санкт-Петербург : «Питер», 2005. — 203 с.

13 Элфрид Дастин, Джеф Решка, Джон Пол. Автоматизированное тестирование программного обеспечения. – Москва. : Лори, 2003. – 592 с.

Семахин Андрей Михайлович

УПРАВЛЕНИЕ КАЧЕСТВОМ И ТЕСТИРОВАНИЕ ПРОГРАММНОГО  
ОБЕСПЕЧЕНИЯ

Методические указания  
к выполнению лабораторных работ  
для студентов направления подготовки 09.03.04  
«Программная инженерия»

Редактор Г. В. Меньщикова

---

Подписано в печать 16.10.17	Формат 60x84 1/16	Бумага 65 г/м <sup>2</sup>
Печать цифровая	Усл. печ. л. 2,75	Уч.-изд. л. 2,75
Заказ №170	Тираж 25	Не для продажи

---

РИЦ Курганского государственного университета.  
640020, г. Курган, ул. Советская, 63/4.  
Курганский государственный университет.