

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Оренбургский государственный университет»

Кафедра программного обеспечения вычислительной техники и
автоматизированных систем

Л.Ф. Тагирова, Е.Н. Чернопрудова

РЕШЕНИЕ ПРАКТИЧЕСКИХ ЗАДАЧ СТРУКТУРНОГО ПРОГРАММИРОВАНИЯ. ЧАСТЬ 1

Методические указания

Рекомендовано к изданию редакционно-издательским советом федерального государственного бюджетного образовательного учреждения высшего образования «Оренбургский государственный университет» для обучающихся по образовательным программам высшего образования по направлениям подготовки 09.03.01 Информатика и вычислительная техника и 09.03.04 Программная инженерия

Оренбург
2019

УДК 004.422.63(076.5)

ББК 32.973-018я7

Т 13

Рецензент – доктор технических наук, профессор Т.М. Зубкова

Тагирова, Л. Ф.

Т 13

Решение практических задач структурного программирования. Часть 1: методические указания / Л.Ф. Тагирова, Е.Н. Чернопрудова; Оренбургский гос. унт. Оренбург: ОГУ, 2019. – 74 с.

Методические указания содержат рекомендации для проведения практических занятий по разработке программ с помощью языка программирования высокого уровня С++. Описывается создание приложений с использованием линейных, разветвляющихся и циклических алгоритмов. Помимо этого в методических указаниях представлены задания по работе с одномерными и двумерными массивами, а также указателями и строками в языке С++.

Методические указания предназначены для обучающихся по направлениям подготовки: 09.03.01 Информатика и вычислительная техника при изучении дисциплины «Программирование» и 09.03.04 Программная инженерия при изучении дисциплины «Программирование и алгоритмизация».

УДК 004.422.63(076.5)

ББК 32.973-018я7

© Тагирова Л.Ф., Чернопрудова Е.Н., 2019

© ОГУ, 2019

Содержание

Введение	4
1 Практическое занятие №1. Основы создания приложений в инструментальной среде MS Visual Studio	6
2 Практическое занятие №2. Программирование линейных алгоритмов на языке C++	18
3 Практическое занятие №3. Реализация разветвляющихся алгоритмов..	23
4 Практическое занятие №4. Разработка приложений для реализации циклических вычислительных процессов.....	28
5 Практическое занятие №5. Работа с одномерными массивами в языке C++	41
6 Практическое задание №6. Обработка двумерных массивов с помощью инструментов языка C++	51
7 Практическое занятие №7. Работа с динамической памятью. Указатели.....	59
8 Практическое занятие №8. Работа со строками символов в языке C++.....	66
Заключение.....	71
Список использованных источников	72
Приложение А.....	74

Введение

Язык C++ компилируемый строго типизированный язык программирования высокого уровня. Поддерживает разные парадигмы программирования: процедурную, обобщённую, функциональную; наибольшее внимание уделено поддержке объектно-ориентированного программирования.

Разработка языка началась в 1979 году. Целью создания C++ было дополнение C возможностями, удобными для масштабной разработки ПО, с сохранением гибкости, скорости и портативности C. Изначально новый язык назывался “C с классами”, но затем имя было изменено на C++. Это должно было подчеркнуть как его происхождение от C, так и его превосходство над последним.

Первый выпуск C++ для коммерческого использования состоялся в 1985 году. В 1989 году вышла вторая версия языка. В 1990-х годах язык стал одним из наиболее широко используемых языков программирования общего назначения.

Первым официальным стандартом языка стал ISO/IEC 14882:1998, более известный как C++98. В 2003 году была принята его дополненная версия, C++03, а в 2005 году был опубликован “Library Technical Report 1” (сокращенно TR1) - документ, описывающий расширения стандартной библиотеки. Наконец, в 2011 году был принят текущий стандарт, C++11.

Целью данных методических указаний является обучение студентов основам написания программ на языке C++ с использованием базовых инструментов языка.

Представленные методические указания состоят из введения, восьми глав, включающих описание практических работ, заключения, списка использованных источников, приложения.

В первой главе представлено задание на практическую работу по созданию приложений в инструментальной среде MS Visual Studio.

Во второй главе описана практическая работа по реализации линейных алгоритмов.

Далее, в третьей главе, рассматривается практическая работа по реализации разветвляющихся алгоритмов.

В следующей главе, в практической работе номер четыре представлено задание по разработки приложений по реализации циклических вычислительных процессов.

В пятой главе рассматривается работа с одномерными массивами языка C++.

В шестой главе представлена практическая работа по реализации многомерных массивов.

В седьмой главе представлено задание на выполнение работ с динамической памятью и указателями. Далее, в восьмой главе, рассматривается работа со строками символов.

В приложении представлены структура оформления отчета к практическим заданиям.

После окончания изучения представленных методических указаний студенты будут уметь самостоятельно разрабатывать приложения на языке высоко уровня C++.

Авторы выражают глубокую благодарность и признательность рецензенту за внимательное прочтение рукописи и замечания, способствовавшие улучшению качества предлагаемых методических указаний.

1 Практическое занятие №1. Основы создания приложений в инструментальной среде MS Visual Studio

1.1 Основное меню программного средства

Microsoft Visual Studio - это набор инструментов разработки, основанных на использовании компонентов и других технологий для создания мощных, производительных приложений. На основе данного инструментального средства разрабатываются приложения на языке C++.

1.1 Основное меню программного средства

Главная форма MS Visual Studio 2017 представлена на рисунке 1. В верхней части окна находится меню и панель инструментов. Меню состоит из 12 вкладок: «Файл», «Правка», «Вид», «Проект», «Отладка», «Команда», «Средства», «Тестирование», «Инструменты», «Анализ», «Окно» и «Справка».

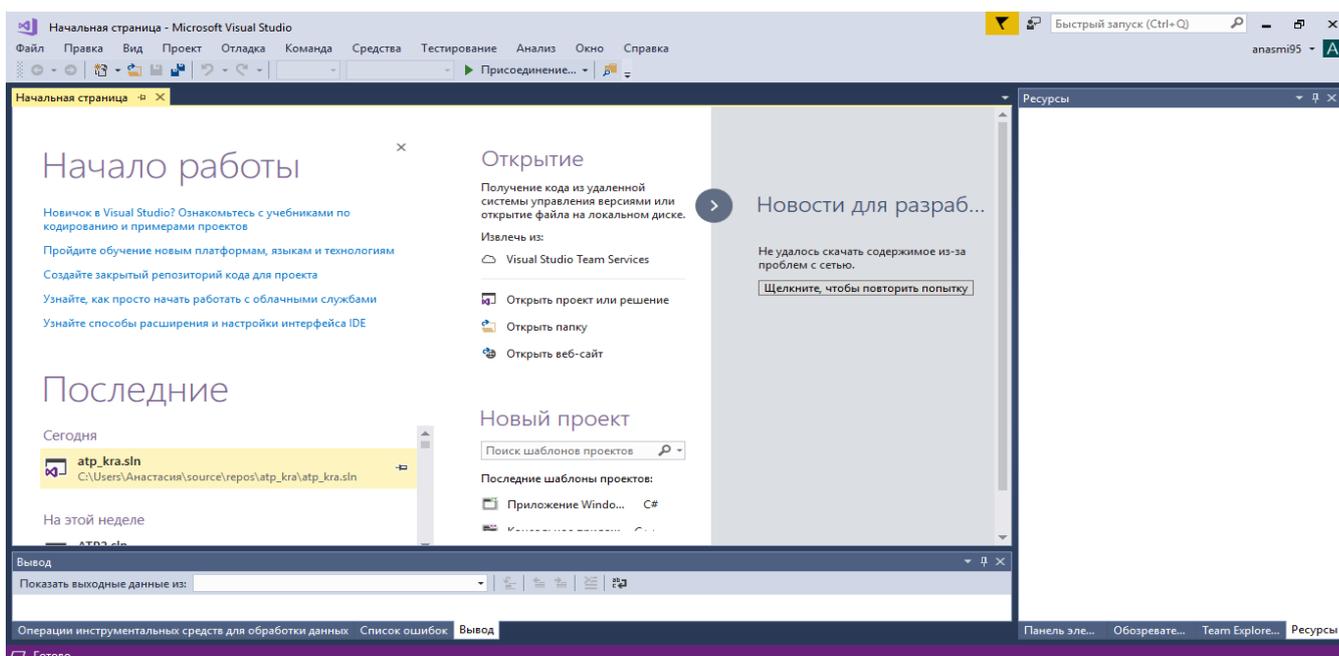


Рисунок 1 – Главная форма Visual Studio 2017

На рисунке 2 представлено основное меню программного средства MS Visual Studio 2017.

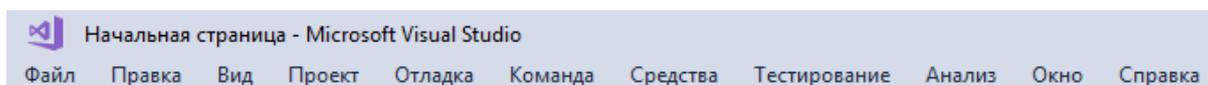


Рисунок 2 – Главное меню Visual Studio

Далее представлено описание основных компонентов главного меню программы. С помощью вкладки «Файл» пользователь может создавать новый проект, открывать ранее созданные проекты, сохранять и выводить на печать проект.

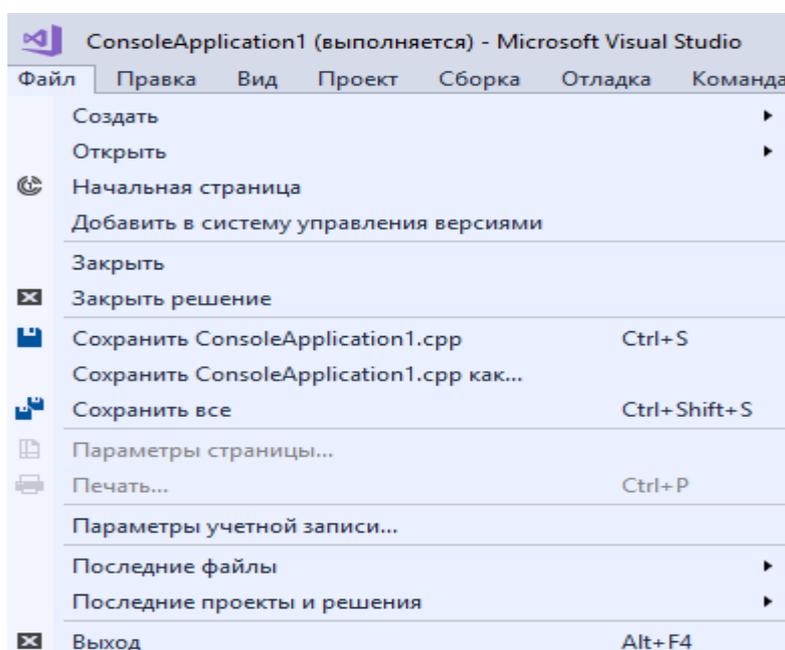


Рисунок 3 – Содержание вкладки «Файл»

Вкладка «Правка» содержит операции, которые позволят пользователю работать наиболее комфортно. К ним относятся такие операции, как: отменить последнее действие, вернуться вперед на одно действие, вырезать, копировать, вставить. Также, с помощью вкладки «Правка» можно вызвать строку поиска по документу.

Вкладка «Вид» позволяет работать с окнами и отвечает за такие функции как просмотр ошибок, уведомлений, отображение начального окна приложения. Также с помощью вкладки «Вид» можно настраивать рабочее окно пользователя.

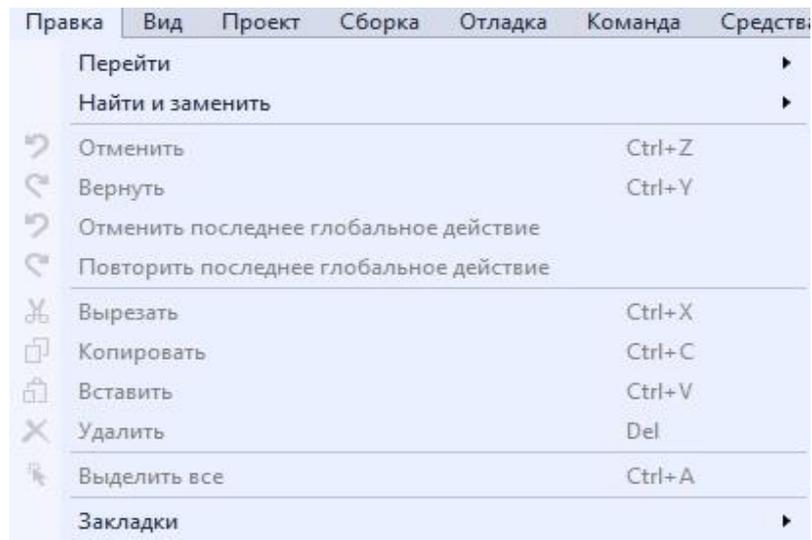


Рисунок 4 – Содержание вкладки «Правка»

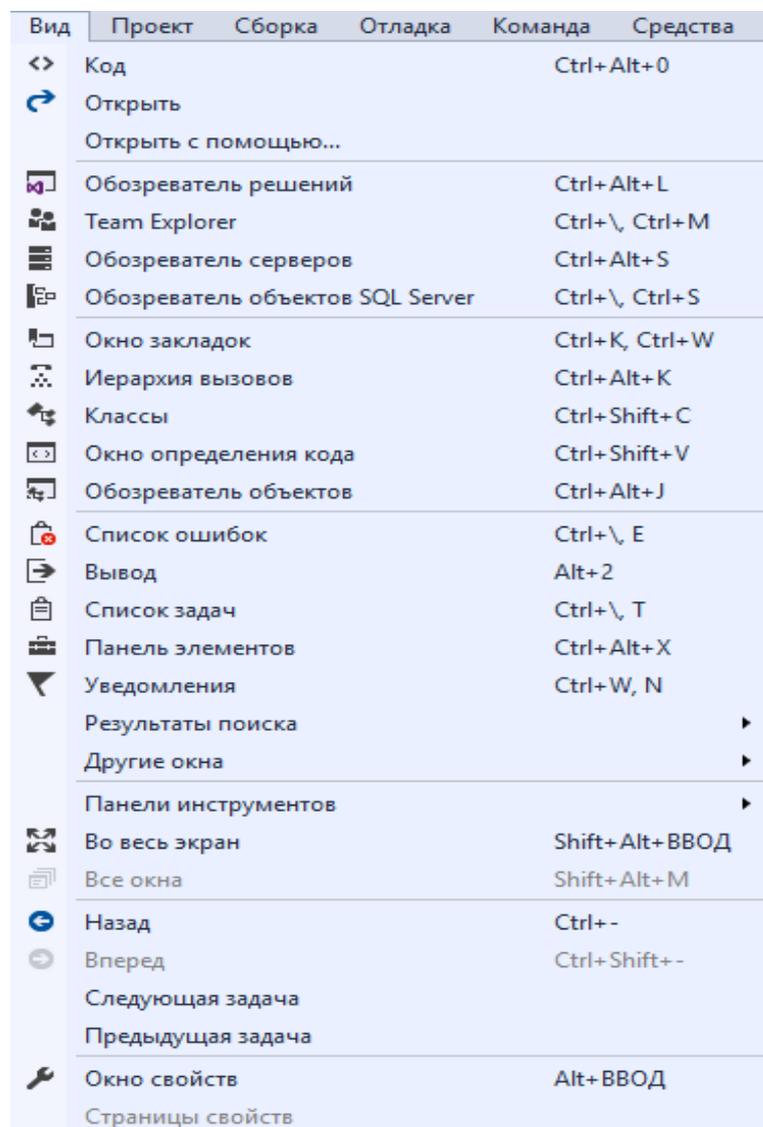


Рисунок 5 – Содержание вкладки «Вид»

С помощью вкладки «Проект» происходит добавление различных элементов/структурных единиц проекта.

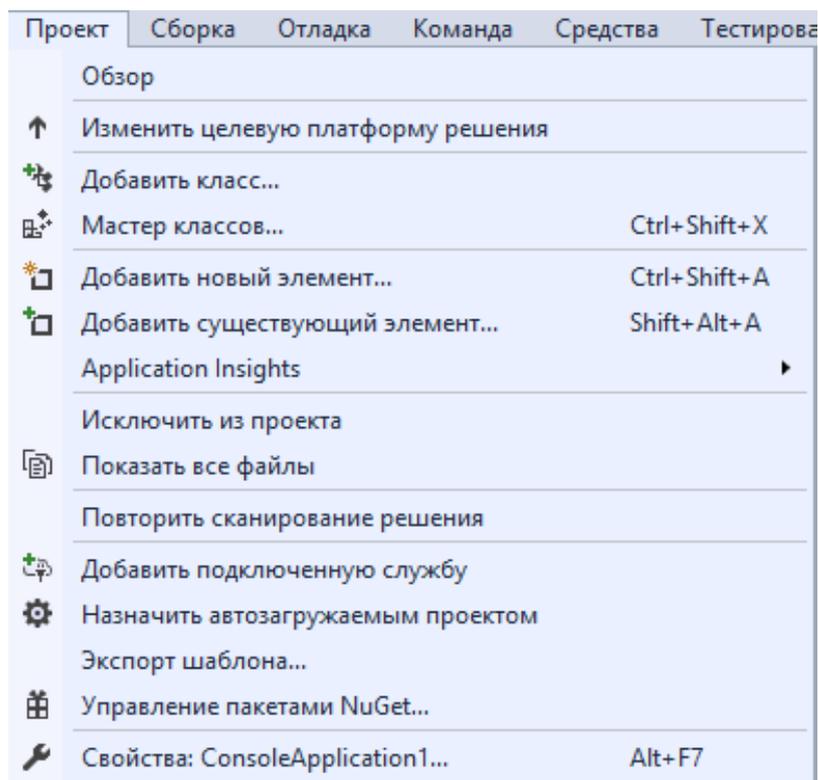


Рисунок 6 – Содержание вкладки «Проект»

Вкладка «Сборка» позволяет выполнить сборку/пересборку проекта для дальнейшего выполнения его работы при компиляции или экспорте.

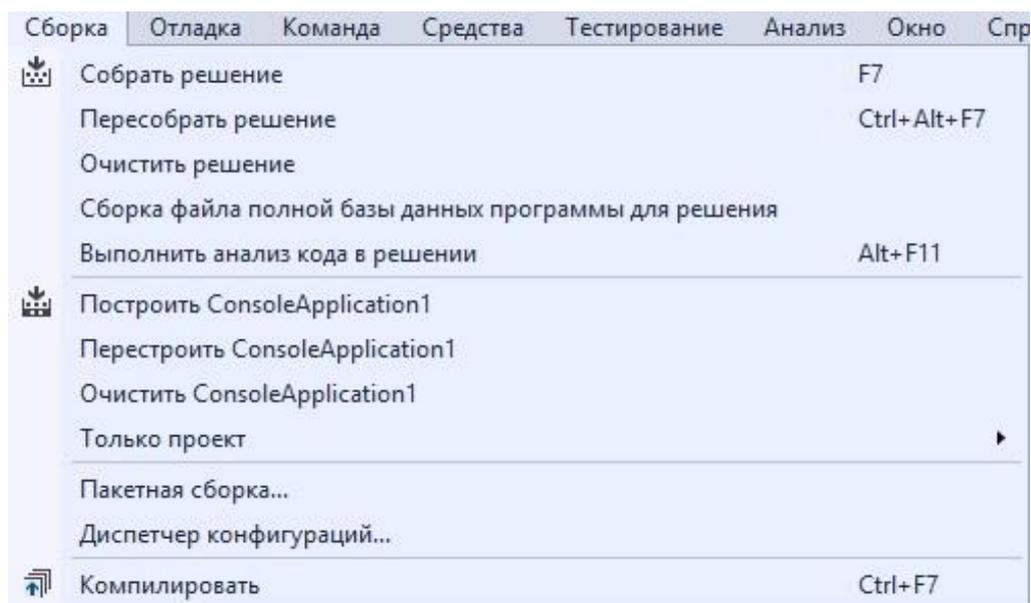


Рисунок 7 – Содержание вкладки «Сборка»

Вкладка «Отладка» позволяет реализовать выбор дополнительных компонентов для разработки программы.

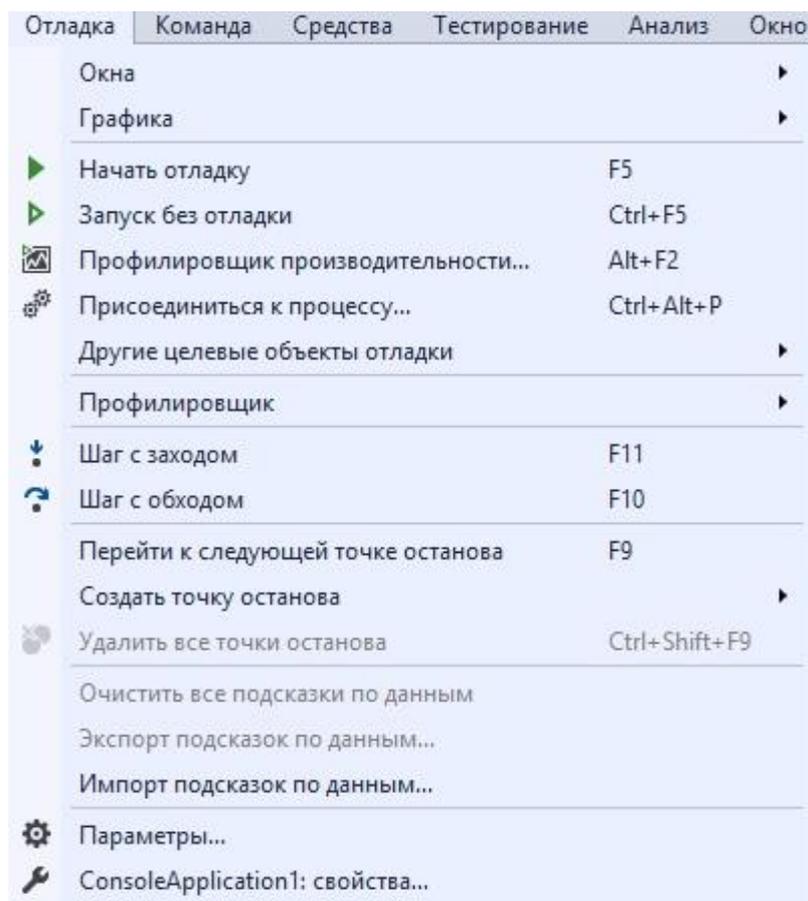


Рисунок 8 – Содержание вкладки «Отладка»

С помощью вкладки «Команда» происходит управление подключением к поставщикам размещенных служб (т.е. к корпорации Майкрософт) и к локальной репозитории.

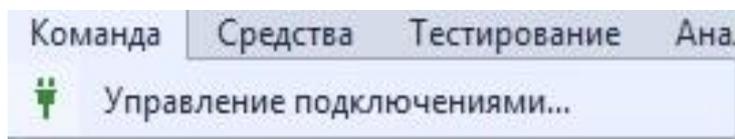


Рисунок 9 – Содержание вкладки «Команда»

С помощью вкладки «Средства» происходит процесс подключения к базе данным и к серверам (рисунок 9).

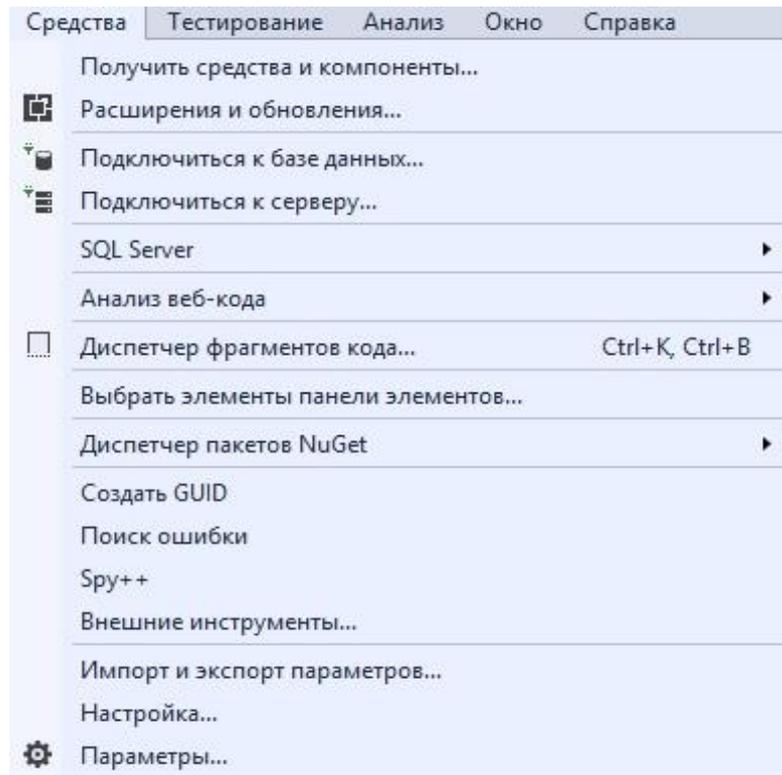


Рисунок 10 – Содержание вкладки «Средства»

Во вкладке «Тестирование» происходит тестирование написанной программы.

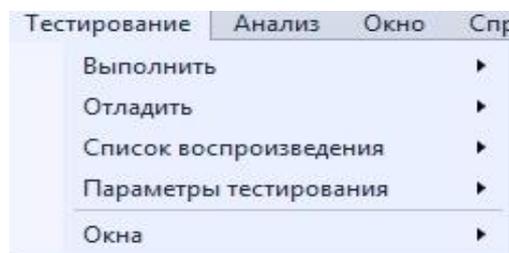


Рисунок 11 – Содержание вкладки «Тестирование»

Вкладка «Анализ» содержит профилировщик производительности, с помощью которого происходит расчет метрики кода как для выбранного проекта, так и для решения. Также в данной вкладке разрабатывается отчет работы.

С помощью вкладки «Окно» выполняются такие действия с рабочим окном, как закрепить как вкладку, скрыть, создание макета окна и т.д.

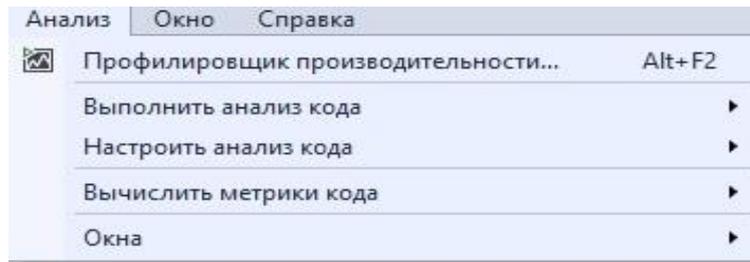


Рисунок 12 – Содержание вкладки «Анализ»

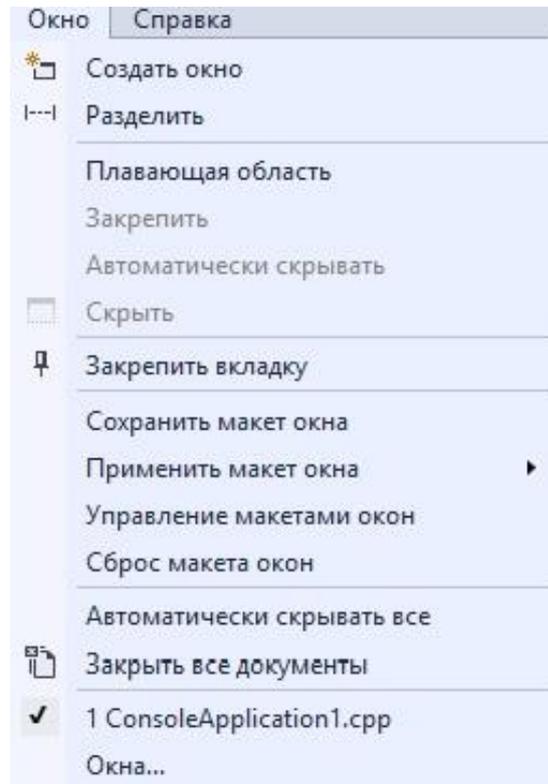


Рисунок 13 – Содержание вкладки «Окно»

Вкладке «Справка» содержит документацию по текущей версии Microsoft Visual Studio.

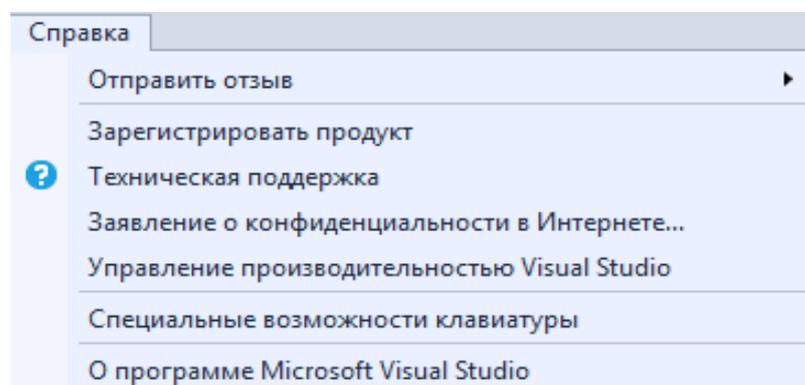


Рисунок 14 – Содержание вкладки «Справка»

Таким образом, с помощью перечисленных пунктов меню можно создавать приложения в инструментальной среде Microsoft Visual Studio 2017.

1.2 Пример создания консольного приложения

Для ознакомления работы с инструментальной среды MS Visual Studio представлен пример создания простого приложения, которое при запуске выводит на экран надпись: «Hello World!». Для создания простого приложения в среде MS Visual Studio 2017 необходимо выполнить следующие шаги:

Шаг 1: Создать новый проект в Visual Studio.

Для создания приложения Visual Studio необходимо запустить приложение. После запуска в верхней части окна, в списке меню, необходимо открыть вкладку «Файл». В данной вкладке выбрать «Создать» и из предложенного списка выбрать «Проект» (рисунок 15).

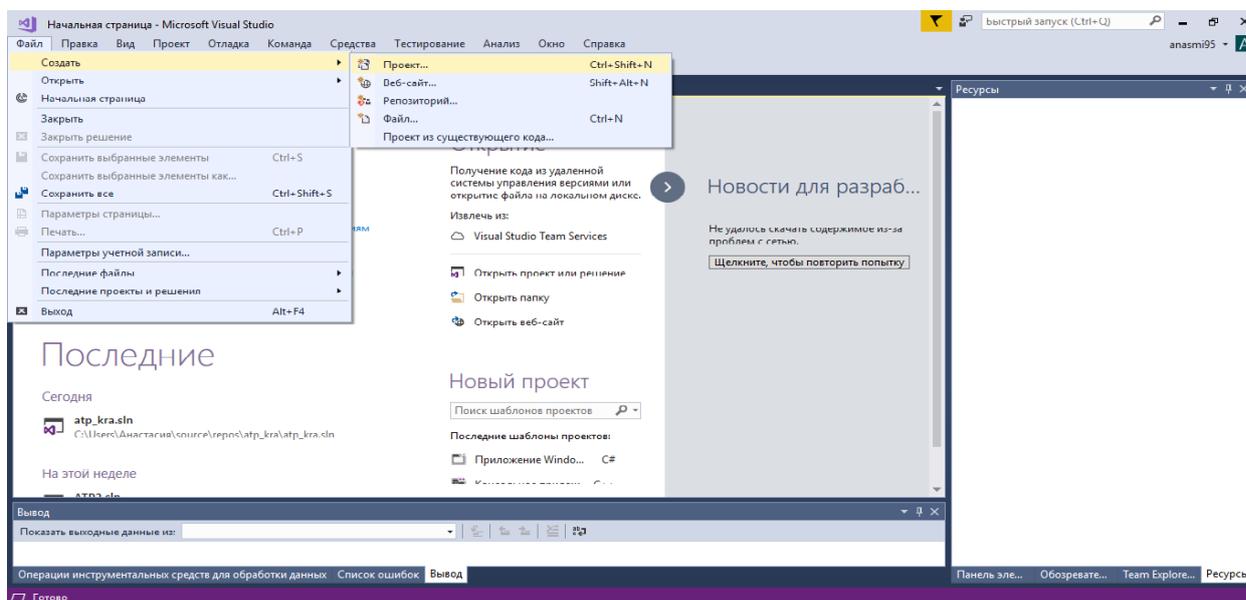


Рисунок 15 – Создание проекта

Шаг 2: Выбрать типа проекта.

При правильном выполнении шага 1, на экране отобразится окно «Создание проекта» (рисунок 15). Для выполнения шага 2, в данном окне следует выбрать «Консольное приложение Windows» и нажать кнопку «Ок».

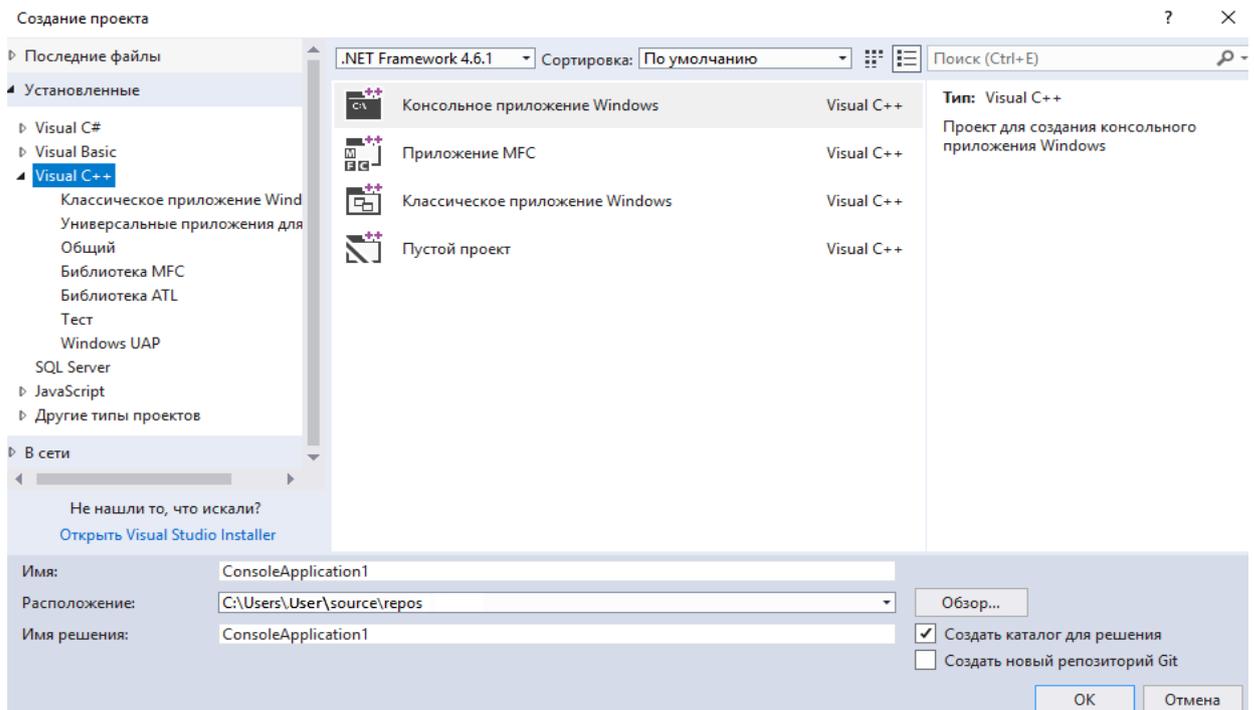


Рисунок 16 – Создание проекта

Шаг 3: Добавить код.

После выполнения шага 3 будет открыто основное рабочее окно проекта (рисунок 17).

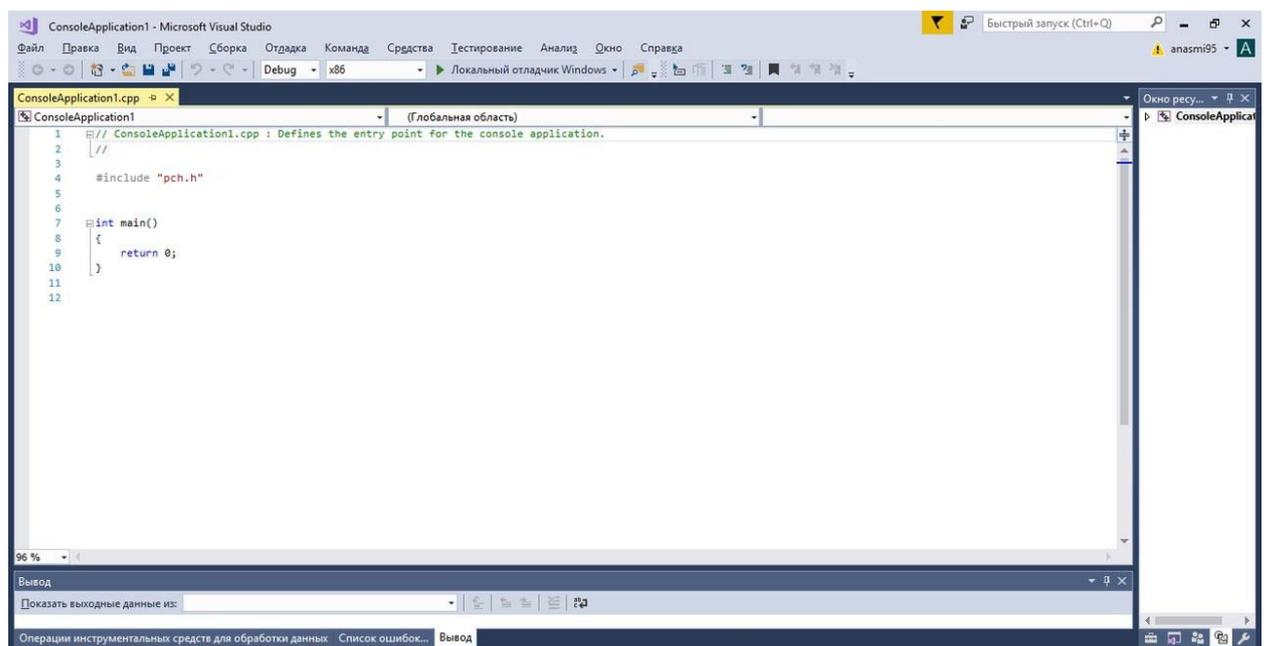


Рисунок 17 - Оконная форма кода проекта

Как видно на рисунке 17, в проекте уже подключена стандартная библиотека `#include "pch.h"`, а также функция `main()`, которая задает основное тело программы, внутри которой следует писать код.

В данное окно следует ввести ниже указанный код.

```
#include "pch.h"  
#include <iostream>  
using namespace std;  
int main()  
{  
cout << ("Hello world!");  
return(0);  
}
```

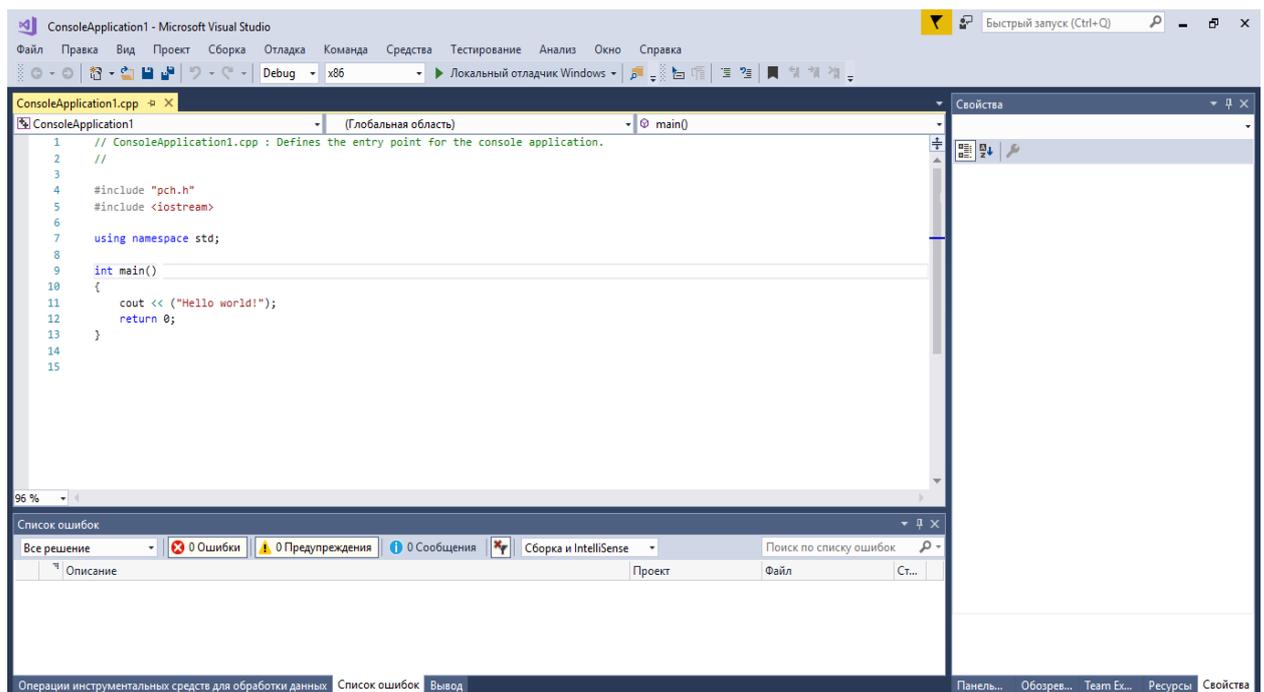


Рисунок 18 – Код программы

Для запуска написанной программы необходимо нажать на кнопку «Локальный отладчик Windows», либо же клавишу F5.

В результате выполнения программного кода будет выведено сообщение, представленное на рисунке 19.

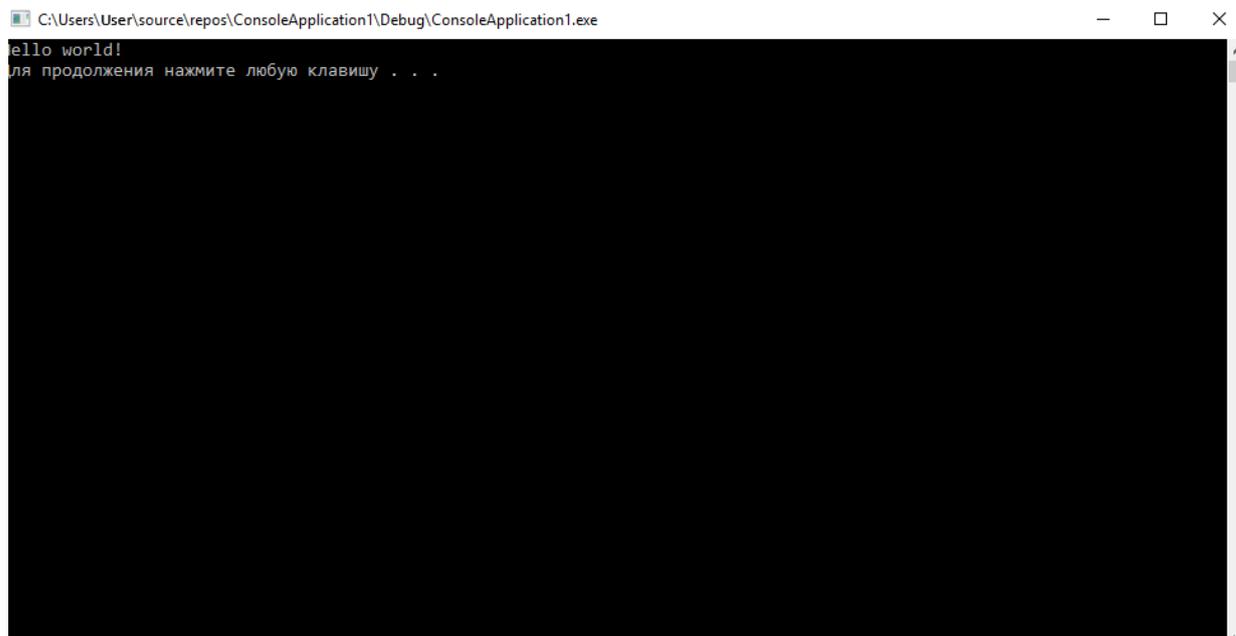


Рисунок 19 – Результат работы программы кода

Функцией отвечающей за вывод заданного текста является функция `cout()`. Для использования данной функции, необходимо подключить библиотеку(файл) `iostream`, а так же указать пространство имен `std`. Для лучшего понимания понятия подключения нужно ознакомиться с понятием препроцессора.

Препроцессор - это программа, которая выполняет обработку файла исходного кода перед началом собственно компиляции. Препроцессор запускается автоматически во время компиляции программы. В листинге использовалась директива `#include`, которая заставляет препроцессор добавить содержимое файла `iostream` в программу. Добавление или замена текста в исходном коде перед его компиляцией является одним из обычных действий препроцессора. Причина состоит в необходимости коммуникаций между программой и внешним миром. В соответствии с директивой `#include`, содержимое файла `windows` будет отправлено компилятору вместе с содержимым вашего файла. В действительности содержимое файла `iostream` заменяет строку `#include` в программе. Ваш исходный файл не

изменяется, а результирующий файл, сформированный из вашего файла и windows, передается на следующий этап компиляции [4].

Файлы, подобные windows, называются включаемыми файлами (поскольку они включаются в другие файлы) или заголовочными файлами (т.к. они включаются в самом начале файла). Компиляторы C++ поставляются вместе с множеством заголовочных файлов, каждый из которых поддерживает отдельное семейство возможностей.

В зависимости от подключаемого файла, зависит и стиль его написания. Так при написании windows в конце стоит .h, а в iostream окончание.h отсутствует.

Помимо описанных выше в C++ также используются другие заголовочные файлы (таблица 1).

Таблица 1 - Именованию заголовочных файлов

Тип заголовка	Соглашение	Пример	Комментарии
Старый стиль C	Заканчивается на .h	math.h	Используется в программах на C и C++
Старый стиль C++	Заканчивается на .h	iostream.h	Используется в программах на C++
Преобразованный C	Префикс с без расширения	cmath	Используется в программах на C++, можно использовать средства не характерные для C, например пространство std
Новый стиль C++	Без расширения	iostream	Используется в программах на C++, использует пространство std

2 Практическое занятие №2. Программирование линейных алгоритмов на языке C++

Цель: Закрепление теоретических знаний и получение практических навыков программирования линейных алгоритмов на языке C++.

Задание. Разработать программу на языке C++ с использованием линейных алгоритмов.

Вариант 1. Необходимо составить программу для вычисления значения функции $y=3x+6$ при любом значении x . Результат работы программы вывести на экран.

Вариант 2. Необходимо составить программу для вычисления периметра квадрата, если известна его сторона. Результат работы программы вывести на экран.

Вариант 3. Необходимо составить программу для вычисления длины окружности, если известен ее радиус. Результат работы программы вывести на экран.

Вариант 4. Необходимо составить программу для вычисления площади окружности, если известен ее диаметр. Результат работы программы вывести на экран.

Вариант 5. Необходимо составить программу для вычисления гипотенузы прямоугольного треугольника, если известны его катеты. Результат работы программы вывести на экран.

Вариант 6. Необходимо составить программу для вычисления периметра прямоугольного треугольника, если известны его катеты. Результат работы программы вывести на экран.

Вариант 7. Необходимо составить программу для вычисления периметра прямоугольника и его диагонали, если известны его стороны. Результат работы программы вывести на экран.

Вариант 8. Необходимо составить программу для вычисления площади кольца, если известны радиуса внешней и внутренней окружности. Результат работы программы вывести на экран.

Вариант 9. Необходимо составить программу для вычисления площади поверхности и объема куба, если известно его ребро. Результат работы программы вывести на экран.

Вариант 10. Необходимо составить программу для вычисления периметра равнобедренной трапеции, если известны ее основания и высота. Результат работы программы вывести на экран.

Вариант 11. Необходимо составить программу для определения плотности тела, если известны его объем и масса. Результат работы программы вывести на экран.

Вариант 12. Необходимо составить программу для вычисления суммы, разности, произведения и частного двух чисел. Результат работы программы вывести на экран.

Вариант 13. Необходимо составить программу для вычисления среднего арифметического и среднего геометрического двух положительных чисел. Результат работы программы вывести на экран.

Вариант 14. Необходимо составить программу вычисления плотности населения в государстве, если известны его площадь и количество жителей. Результат работы программы вывести на экран.

Вариант 15. Необходимо составить программу, в которую ввести числа $a = 3$, $b = 8$. Требуется вычислить сумму S и разность R чисел a и b . Результат работы программы вывести на экран.

Вариант 16. Необходимо составить программу для вычисления значений функций $y = \sin x$ и $z = \ln x$ при x , который считывается с экрана (клавиатуры). Результат работы программы вывести на экран.

Вариант 17. Необходимо составить программу для вычисления значений функций $U = x^2 + \frac{y+3}{x-1}$ и $V = \frac{1}{y^e} \sin^2(\sqrt{x} + 1.5)$ при различных значениях аргументов x, y . Переменные x, y считать с клавиатуры. Результат работы программы вывести на экран.

Задание 18. Необходимо составить программу, в которой найти длину окружности и площадь круга, если известен радиус. Результат работы программы вывести на экран.

Задание 19. Необходимо составить программу для вычисления значений функций $z1 = \arccos^3(2, 5x - 0, 6y)$, $z2 = 0, 45|x^3 - y^2| + 2 \lg^{2,5} y$ при различных значениях аргументов x и y . Результат работы программы вывести на экран.

Задание 20. Необходимо составить программу, в которой вычислить значение многопараметрической функции:

$$Y = \sin(x^3 + 2a) + \frac{\cos x + e^x}{\sqrt{|1 - \operatorname{arctg}(a - 0,5)|}}$$
. Результат работы программы вывести на экран.

Пример выполнения практического задания №2.

Задание.

Составить программу вычисления периметра квадрата, если известна сторона.

Решение.

1) Для работы с потоками вывода и ввода была использована библиотека `iostream`. Вывод на экран осуществлялся с помощью оператора `cout`. Ввод – `cin`.

В случаях, когда необходимо было вывести текст, надпись заключалась в кавычки: `cout << "Выводимый текст"`.

С помощью функции `setlocale` настраивались локальные параметры в программе. Функция `setlocale(LC_ALL, "Rus")` использовалась для установки русского языка в консоли.

Так как для решения задачи требовалось найти периметр квадрата, был использован линейным алгоритм. Для нахождения периметра использовалась формула: $P = 4 * a$.

2) Схема алгоритма. Схема алгоритма программы представлена на рисунке 20.

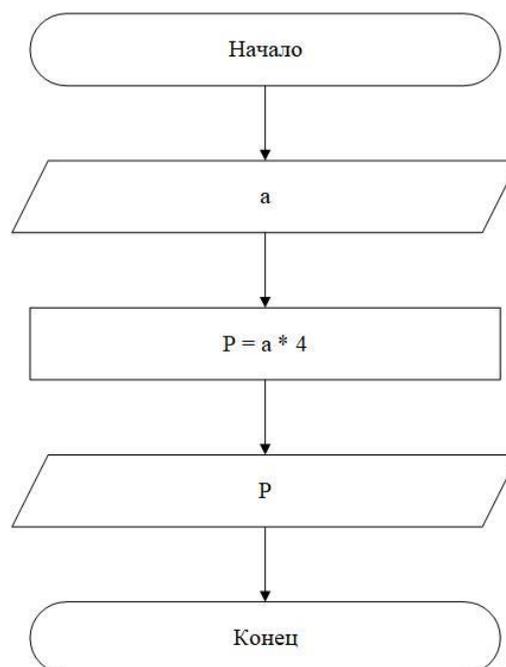


Рисунок 20 - Схема алгоритма реализации линейного алгоритма

3) Код программы.

```
#include "pch.h"
```

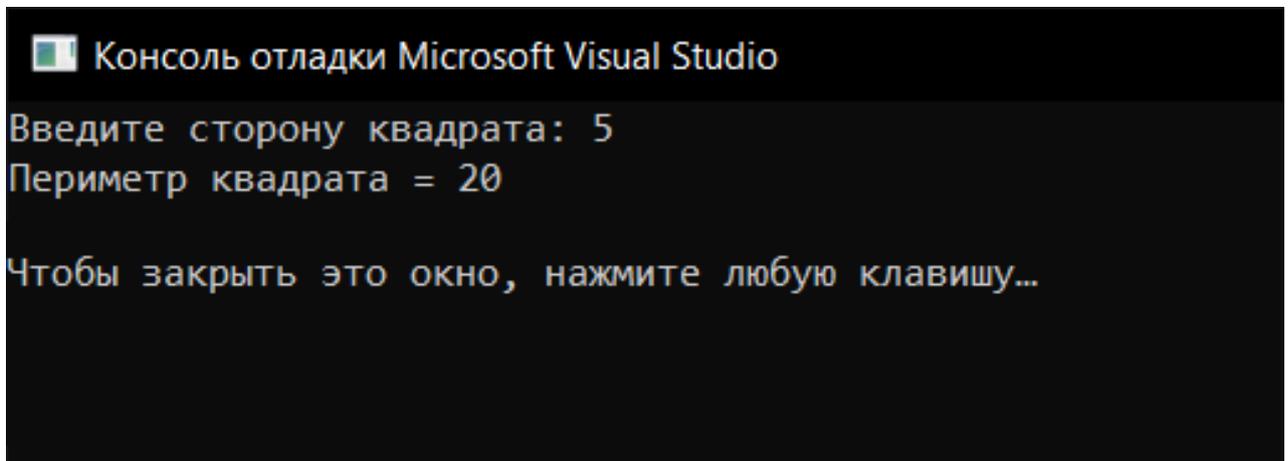
```
#include <iostream> // Подключение библиотеки для работы с потоками  
ввода-вывода
```

```

using namespace std;
int main() {
    setlocale(LC_ALL, "Rus");           // Установка русского языка в
консоли
    int a, P;                           // Инициализация переменных
    cout << "Введите сторону квадрата: "; // Вывод текста в
консоль
    cin >> a;                             // Ввод переменной с клавиатуры
    P = a * 4;
    cout << "Периметр квадрата = " << P << endl; // Вывод результата на
экран
}

```

4) Реализация программного кода. Реализация программного кода представлена на рисунке 21.



```

Консоль отладки Microsoft Visual Studio
Введите сторону квадрата: 5
Периметр квадрата = 20
Чтобы закрыть это окно, нажмите любую клавишу...

```

Рисунок 21 – Результат работы обработки линейных алгоритмов

3 Практическое занятие №3. Реализация разветвляющихся алгоритмов

Цель: Закрепление теоретических знаний и получение практических навыков программирования разветвляющихся алгоритмов на языке C++

Задание. Разработать программу на языке C++ с использованием разветвляющихся алгоритмов.

Вариант 1. Необходимо составить программу, в которую ввести числа $a = 3$, $b = 8$. Требуется вычислить сумму S и разность R чисел a и b . Сравнить полученные значения S и R и указать большее из них.

Вариант 2. Необходимо составить программу для вычисления значений функций $y = \sin x$ и $z = \ln x$ при различных значениях аргумента. Необходимо учесть область определения функций.

Вариант 3. Необходимо составить программу для вычисления значения функции:

$$Z = \begin{cases} x + 3, & x < 1 \\ 4x, & x \geq 1 \end{cases}$$

Вариант 4. Необходимо составить программу для вычисления значения функции:

$$Z = \begin{cases} x^2, & \text{если } x < 0 \\ |x|, & \text{если } 0 \leq x \leq 5 \\ 5, & \text{если } x > 5 \end{cases}$$

При различных значениях аргумента x (аргумент x вводить с клавиатуры).

Вариант 5. Необходимо составить программу для вычисления значения функции:

$$Z = \begin{cases} x + y, & xy < 1 \\ x - y, & xy \geq 1 \end{cases}$$

Вариант 6. Необходимо составить программу для вычисления значения функции:

$$Z(x, y) = \begin{cases} x + y, & \text{если } xy < 0 \\ x - y, & \text{если } 0 \leq xy \leq 5 \\ y, & \text{если } xy > 5 \end{cases}$$

При различных значениях аргументов x и y (аргументы x и y вводить с клавиатуры).

Вариант 7. Необходимо составить программу для нахождения наибольшего (максимального) из трех чисел.

Вариант 8. Необходимо составить программу для вычисления значения функции: $Z(x) = \max\{3x; 10 - x; |x|\}$. При различных значениях аргумента x (аргумент x вводить с клавиатуры).

Вариант 9. Необходимо составить программу для вычисления значения функции: $Z(x, y) = \max\{x + y, x - y, xy\}$. При различных значениях аргументов x и y (аргументы x и y вводить с клавиатуры).

Вариант 10. Необходимо составить программу по нахождению корней квадратного уравнения через дискриминант. Коэффициенты квадратного уравнения считывайте с клавиатуры.

Вариант 11. Необходимо составить программу, которая спрашивает у пользователя возраст и в соответствии с возрастной группой (“дошкольник”, “школьник”, “взрослый”, “пожилой”) выдает соответствующую фразу на экран.

Вариант 12. Необходимо составить программу для вычисления значения функции:

$$Z(x, y) = \begin{cases} \sin^{0.4}(4^{4y}) + \lg \sqrt{y - \frac{x}{y-0.8}}, & \text{если } x + y < -2 \\ 2.3, & \text{если } -2 \leq x + y \leq 0 \\ \frac{\sqrt[3]{x} - \sqrt[3]{y}}{x} - y, & \text{если } x + y > 0 \end{cases}$$

При разных значениях аргументов x и y .

Вариант 13. Необходимо составить программу для вычисления значения функции: $Z(x, y) = \min\{\max\{\sin x, \sin y\}, x, y\}$. При разных значениях аргументов x и y .

Вариант 14. Необходимо составить программу для вычисления суммы всех целых чисел от 1 до 100 включительно.

Вариант 15. Необходимо составить программу и построить таблицу значений функции $y(x) = 5x^2 - 3x + 4$ при $-20 \leq x \leq 20$ с шагом 0,5. Требуется определить наименьшее значение функции и значение x , при котором оно достигается.

Вариант 16. Необходимо составить программу для вычисления таблицы значений функций

$z_1 = \arccos^3(0,05x)$, $z_2 = 0,45|x^3 - x^2| + 2 \lg^{2,5} x$ при $0,8 \leq x \leq 19$ с шагом 1. Найдите сумму всех значений z_1 , количество значений z_2 , меньших 10, наименьшее значение z_1 , наибольшее значение z_2 .

Вариант 17. Необходимо составить программу для вычисления таблицы значений функции

$$Z(x, y) = \begin{cases} \sin^{0.4}(x^{4y}) + \lg \sqrt{y - \frac{x}{y-0.8}}, & \text{если } x + y < -2 \\ 2.3, & \text{если } -2 \leq x + y \leq 0 \\ \frac{\sqrt[3]{x} - \sqrt[3]{y}}{x} - y, & \text{если } x + y > 0 \end{cases}$$

При $0,5 \leq x \leq 1$ с шагом 0,05; y вводить с клавиатуры.

Вариант 18. Необходимо составить программу для вычисления таблицы значений функции

$$Z(x, y) = \min\{\max\{\sin x, \sin y\}, x, y\}$$

при $-1,7 \leq x \leq 9,4$ с шагом 9,3; при $0 \leq y \leq 3$ с шагом 0,75. Определите наибольшее отрицательное значение функции, наименьшее положительное значение функции, произведение ненулевых значений функции, количество значений функции, больших 1 или меньших -1.

Вариант 19. Необходимо составить программу, вычисляющую корни квадратного уравнения $ax^2 + bx + c = 0$. Если дискриминант меньше нуля, то вывести сообщение, что действительных корней нет.

Вариант 20. Необходимо составить программу для вычисления значения выражения: $\max(a, bc, \min(d, b))$.

Пример выполнения практического задания №3.

Задание.

Написать программу для поиска наименьшего из трех чисел.

Решение.

1) Словесный алгоритм.

Для решения данной задачи был использован разветвляющийся алгоритм.

Для нахождения минимального из трёх чисел сравнивались операторы if

2) Схема алгоритма программы представлена на рисунке 22.

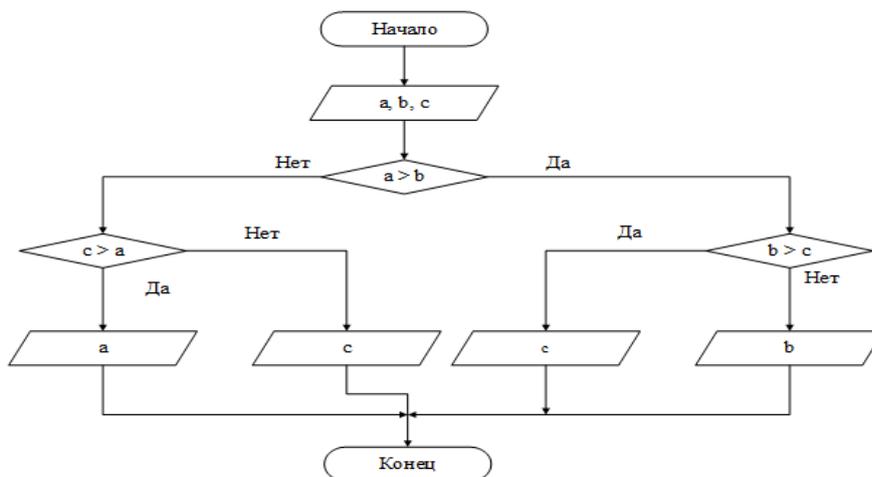


Рисунок 22- Схема алгоритма программы разветвляющегося алгоритма

3) Код программы.

```
#include "pch.h"
#include <iostream>
using namespace std;
int main()
{
    setlocale(LC_ALL, "Rus");
    int a, b, c, min;
    cout << "Введите числа: ";
    cin >> a >> b >> c;
    cout << "Минимальное = ";
    if (a > b) // Сравниваются переменные a и b
    {
        if (b > c) cout << c; // Если b > c выводится переменная c
        else cout << b; } // Иначе выводится b
    else
        if (a > c) cout << c; // Если a > c выводится c
        else cout << a; } // Иначе выводится a
```

4) Реализации программного кода. Реализации программного кода представлена на рисунке 23.

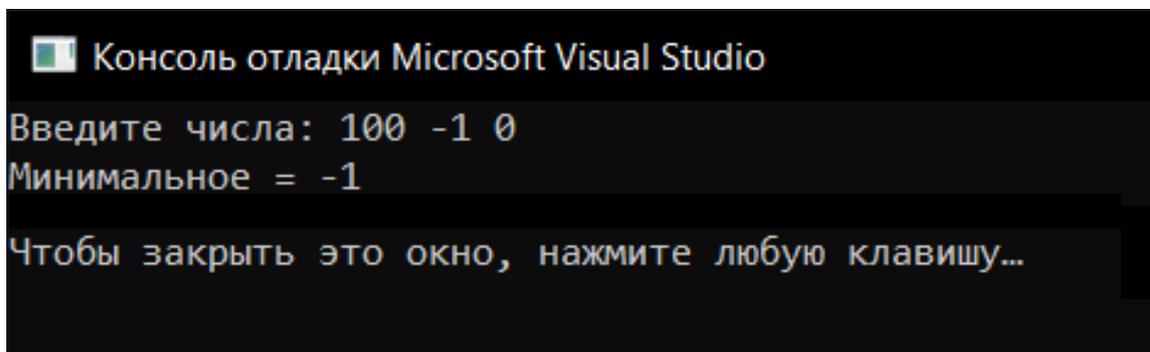


Рисунок 23 – Результат работы программы по обработке разветвляющихся алгоритмов

4 Практическое занятие №4. Разработка приложений для реализации циклических вычислительных процессов

Цель: Закрепление теоретических знаний и получение практических навыков программирования циклических алгоритмов на языке C++

Задание. Разработать программу на языке C++ с использованием циклических алгоритмов.

Вариант 1. Необходимо составить программу, в которой вычислить значения функции F на интервале от $X_{\text{нач}}$ до $X_{\text{кон}}$ с шагом dX . Затем значения функции необходимо вывести в виде таблицы на экран.

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

Значения a, b, c - действительные числа.

Если выражение $(Aц \text{ ИЛИ } Bц)$ И $(Aц \text{ ИЛИ } Cц)$ не равно нулю, то функция должна принимать действительное значение. В противном случае функция должна принимать целое значение. Через $Aц, Bц$ и $Cц$ - целые части значений a, b, c , операции И и ИЛИ - поразрядные. Необходимо вводить с клавиатуры значения $a, b, c, X_{\text{нач}}, X_{\text{кон}}, dX$. Результат работы программы вывести на экран.

Вариант 2. Необходимо составить программу, в которой вычислить значения функции F на интервале от $X_{\text{нач}}$ до $X_{\text{кон}}$ с шагом dX . Затем значения функции необходимо вывести в виде таблицы на экран.

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

Значения a, b, c - действительные числа.

Если выражение $(Aц ИЛИ Bц)$ И $(Aц ИЛИ Cц)$ не равно нулю, то функция должна принимать действительное значение. В противном случае функция должна принимать целое значение. Через $Aц, Bц$ и $Cц$ - целые части значений a, b, c , операции И и ИЛИ - поразрядные. Необходимо вводить с клавиатуры значения $a, b, c, X_{нач}, X_{кон}, dX$. Результат работы программы вывести на экран.

Вариант 3. Необходимо составить программу, в которой вычислить значения функции F на интервале от $X_{нач}$ до $X_{кон}$ с шагом dX . Затем значения функции необходимо вывести в виде таблицы на экран.

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

Значения a, b, c - действительные числа.

Если выражение $(Aц ИЛИ Bц)$ И $(Aц ИЛИ Cц)$ не равно нулю, то функция должна принимать действительное значение. В противном случае функция должна принимать целое значение. Через $Aц, Bц$ и $Cц$ - целые части значений a, b, c , операции И и ИЛИ - поразрядные. Необходимо вводить с клавиатуры значения $a, b, c, X_{нач}, X_{кон}, dX$. Результат работы программы вывести на экран.

Вариант 4. Необходимо составить программу, в которой вычислить значения функции F на интервале от $X_{нач}$ до $X_{кон}$ с шагом dX . Затем значения функции необходимо вывести в виде таблицы на экран.

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

Значения a, b, c - действительные числа.

Если выражение $(Aц ИЛИ Bц)$ И $(Aц ИЛИ Cц)$ не равно нулю, то функция должна принимать действительное значение. В противном случае функция должна принимать целое значение. Через $Aц, Bц$ и $Cц$ - целые части значений a, b, c ,

операции И и ИЛИ - поразрядные. Необходимо вводить с клавиатуры значения a, b, c, X_{нач}, X_{кон}, dX. Результат работы программы вывести на экран.

Вариант 5. Необходимо составить программу, в которой вычислить значения функции F на интервале от X_{нач} до X_{кон} с шагом dX. Затем значения функции необходимо вывести в виде таблицы на экран.

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

Значения a, b, c - действительные числа.

Если выражение (Ац ИЛИ Вц) И (Ац ИЛИ Сц) не равно нулю, то функция должна принимать действительное значение. В противном случае функция должна принимать целое значение. Через Ац, Вц и Сц - целые части значений a, b, c, операции И и ИЛИ - поразрядные. Необходимо вводить с клавиатуры значения a, b, c, X_{нач}, X_{кон}, dX. Результат работы программы вывести на экран.

Вариант 6. Необходимо составить программу, в которой вычислить значения функции F на интервале от X_{нач} до X_{кон} с шагом dX. Затем значения функции необходимо вывести в виде таблицы на экран.

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

Значения a, b, c - действительные числа.

Если выражение (Ац ИЛИ Вц) И (Ац ИЛИ Сц) не равно нулю, то функция должна принимать действительное значение. В противном случае функция должна принимать целое значение. Через Ац, Вц и Сц - целые части значений a, b, c, операции И и ИЛИ - поразрядные. Необходимо вводить с клавиатуры значения a, b, c, X_{нач}, X_{кон}, dX. Результат работы программы вывести на экран.

Вариант 7. Необходимо составить программу, в которой вычислить значения функции F на интервале от $X_{\text{нач}}$ до $X_{\text{кон}}$ с шагом dX . Затем значения функции необходимо вывести в виде таблицы на экран.

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

Значения a, b, c - действительные числа.

Если выражение $(Aц \text{ ИЛИ } Bц)$ И $(Aц \text{ ИЛИ } Cц)$ не равно нулю, то функция должна принимать действительное значение. В противном случае функция должна принимать целое значение. Через $Aц, Bц$ и $Cц$ - целые части значений a, b, c , операции И и ИЛИ - поразрядные. Необходимо вводить с клавиатуры значения $a, b, c, X_{\text{нач}}, X_{\text{кон}}, dX$. Результат работы программы вывести на экран.

Вариант 8. Необходимо составить программу, в которой вычислить значения функции F на интервале от $X_{\text{нач}}$ до $X_{\text{кон}}$ с шагом dX . Затем значения функции необходимо вывести в виде таблицы на экран.

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

Значения a, b, c - действительные числа.

Если выражение $(Aц \text{ ИЛИ } Bц)$ И $(Aц \text{ ИЛИ } Cц)$ не равно нулю, то функция должна принимать действительное значение. В противном случае функция должна принимать целое значение. Через $Aц, Bц$ и $Cц$ - целые части значений a, b, c , операции И и ИЛИ - поразрядные. Необходимо вводить с клавиатуры значения $a, b, c, X_{\text{нач}}, X_{\text{кон}}, dX$. Результат работы программы вывести на экран.

Вариант 9. Необходимо составить программу, в которой вычислить значения функции F на интервале от $X_{\text{нач}}$ до $X_{\text{кон}}$ с шагом dX . Затем значения функции необходимо вывести в виде таблицы на экран.

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

Значения a, b, c - действительные числа.

Если выражение $(Aц \text{ ИЛИ } Bц)$ И $(Aц \text{ ИЛИ } Cц)$ не равно нулю, то функция должна принимать действительное значение. В противном случае функция должна принимать целое значение. Через $Aц, Bц$ и $Cц$ - целые части значений a, b, c , операции И и ИЛИ - поразрядные. Необходимо вводить с клавиатуры значения $a, b, c, X_{нач}, X_{кон}, dX$. Результат работы программы вывести на экран.

Вариант 10. Необходимо составить программу, в которой вычислить значения функции F на интервале от $X_{нач}$ до $X_{кон}$ с шагом dX . Затем значения функции необходимо вывести в виде таблицы на экран.

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

Значения a, b, c - действительные числа.

Если выражение $(Aц \text{ ИЛИ } Bц)$ И $(Aц \text{ ИЛИ } Cц)$ не равно нулю, то функция должна принимать действительное значение. В противном случае функция должна принимать целое значение. Через $Aц, Bц$ и $Cц$ - целые части значений a, b, c , операции И и ИЛИ - поразрядные. Необходимо вводить с клавиатуры значения $a, b, c, X_{нач}, X_{кон}, dX$. Результат работы программы вывести на экран.

Вариант 11. Необходимо составить программу, в которой вычислить значения функции F на интервале от $X_{нач}$ до $X_{кон}$ с шагом dX . Затем значения функции необходимо вывести в виде таблицы на экран.

$$F = \begin{cases} ax^2 + \frac{b}{c}, & \text{при } x < 1 \text{ и } c \neq 0 \\ \frac{x-a}{(x-c)^2}, & \text{при } x > 1.5 \text{ и } c = 0 \\ \frac{x^2}{c^2}, & \text{в остальных случаях} \end{cases}$$

Значения a, b, c - действительные числа.

Если выражение $(Aц \text{ ИЛИ } Bц)$ И $(Aц \text{ ИЛИ } Cц)$ не равно нулю, то функция должна принимать действительное значение. В противном случае функция должна принимать целое значение. Через $Aц, Bц$ и $Cц$ - целые части значений a, b, c , операции И и ИЛИ - поразрядные. Необходимо вводить с клавиатуры значения $a, b, c, X_{нач}, X_{кон}, dX$. Результат работы программы вывести на экран.

Вариант 12. Необходимо составить программу, в которой вычислить значения функции F на интервале от $X_{нач}$ до $X_{кон}$ с шагом dX . Затем значения функции необходимо вывести в виде таблицы на экран.

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

Значения a, b, c - действительные числа.

Если выражение $(Aц \text{ ИЛИ } Bц)$ И $(Aц \text{ ИЛИ } Cц)$ не равно нулю, то функция должна принимать действительное значение. В противном случае функция должна принимать целое значение. Через $Aц, Bц$ и $Cц$ - целые части значений a, b, c , операции И и ИЛИ - поразрядные. Необходимо вводить с клавиатуры значения $a, b, c, X_{нач}, X_{кон}, dX$. Результат работы программы вывести на экран.

Вариант 13. Необходимо составить программу, в которой вычислить значения функции F на интервале от $X_{нач}$ до $X_{кон}$ с шагом dX . Затем значения функции необходимо вывести в виде таблицы на экран.

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

Значения a, b, c - действительные числа.

Если выражение $(Aц \text{ ИЛИ } Bц)$ И $(Aц \text{ ИЛИ } Cц)$ не равно нулю, то функция должна принимать действительное значение. В противном случае функция должна принимать целое значение. Через $Aц, Bц$ и $Cц$ - целые части значений a, b, c , операции И и ИЛИ - поразрядные. Необходимо вводить с клавиатуры значения $a, b, c, X_{нач}, X_{кон}, dX$. Результат работы программы вывести на экран.

Вариант 14. Необходимо составить программу, в которой вычислить значения функции F на интервале от $X_{нач}$ до $X_{кон}$ с шагом dX . Затем значения функции необходимо вывести в виде таблицы на экран.

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

Значения a, b, c - действительные числа.

Если выражение $(Aц \text{ ИЛИ } Bц)$ И $(Aц \text{ ИЛИ } Cц)$ не равно нулю, то функция должна принимать действительное значение. В противном случае функция должна принимать целое значение. Через $Aц, Bц$ и $Cц$ - целые части значений a, b, c , операции И и ИЛИ - поразрядные. Необходимо вводить с клавиатуры значения $a, b, c, X_{нач}, X_{кон}, dX$. Результат работы программы вывести на экран.

Вариант 15. Необходимо составить программу, в которой вычислить значения функции F на интервале от $X_{нач}$ до $X_{кон}$ с шагом dX . Затем значения функции необходимо вывести в виде таблицы на экран.

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

Значения a, b, c - действительные числа.

Если выражение $(Aц ИЛИ Bц)$ И $(Aц ИЛИ Cц)$ не равно нулю, то функция должна принимать действительное значение. В противном случае функция должна принимать целое значение. Через $Aц, Bц$ и $Cц$ - целые части значений a, b, c , операции И и ИЛИ - поразрядные. Необходимо вводить с клавиатуры значения $a, b, c, X_{нач}, X_{кон}, dX$. Результат работы программы вывести на экран.

Вариант 16. Необходимо составить программу, в которой вычислить значения функции F на интервале от $X_{нач}$ до $X_{кон}$ с шагом dX . Затем значения функции необходимо вывести в виде таблицы на экран.

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

Значения a, b, c - действительные числа.

Если выражение $(Aц ИЛИ Bц)$ И $(Aц ИЛИ Cц)$ не равно нулю, то функция должна принимать действительное значение. В противном случае функция должна принимать целое значение. Через $Aц, Bц$ и $Cц$ - целые части значений a, b, c , операции И и ИЛИ - поразрядные. Необходимо вводить с клавиатуры значения $a, b, c, X_{нач}, X_{кон}, dX$. Результат работы программы вывести на экран.

Вариант 17. Необходимо составить программу, в которой вычислить значения функции F на интервале от $X_{нач}$ до $X_{кон}$ с шагом dX . Затем значения функции необходимо вывести в виде таблицы на экран.

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

Значения a, b, c - действительные числа.

Если выражение $(Aц ИЛИ Bц)$ И $(Aц ИЛИ Cц)$ не равно нулю, то функция должна принимать действительное значение. В противном случае функция должна

принимать целое значение. Через Ац, Вц и Сц - целые части значений a, b, c, операции И и ИЛИ - поразрядные. Необходимо вводить с клавиатуры значения a, b, c, X_{нач}, X_{кон}, dX. Результат работы программы вывести на экран.

Вариант 18. Необходимо составить программу, в которой вычислить значения функции F на интервале от X_{нач} до X_{кон} с шагом dX. Затем значения функции необходимо вывести в виде таблицы на экран.

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

Значения a, b, c - действительные числа.

Если выражение (Ац ИЛИ Вц) И (Ац ИЛИ Сц) не равно нулю, то функция должна принимать действительное значение. В противном случае функция должна принимать целое значение. Через Ац, Вц и Сц - целые части значений a, b, c, операции И и ИЛИ - поразрядные. Необходимо вводить с клавиатуры значения a, b, c, X_{нач}, X_{кон}, dX. Результат работы программы вывести на экран.

Вариант 19. Необходимо составить программу, в которой вычислить значения функции F на интервале от X_{нач} до X_{кон} с шагом dX. Затем значения функции необходимо вывести в виде таблицы на экран.

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

Значения a, b, c - действительные числа.

Если выражение (Ац ИЛИ Вц) И (Ац ИЛИ Сц) не равно нулю, то функция должна принимать действительное значение. В противном случае функция должна принимать целое значение. Через Ац, Вц и Сц - целые части значений a, b, c, операции И и ИЛИ - поразрядные. Необходимо вводить с клавиатуры значения a, b, c, X_{нач}, X_{кон}, dX. Результат работы программы вывести на экран.

Вариант 20. Необходимо составить программу, в которой вычислить значения функции F на интервале от $X_{\text{нач}}$ до $X_{\text{кон}}$ с шагом dX . Затем значения функции необходимо вывести в виде таблицы на экран.

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

Значения a, b, c - действительные числа.

Если выражение $(Aц \text{ ИЛИ } Bц) \text{ И } (Aц \text{ ИЛИ } Cц)$ не равно нулю, то функция должна принимать действительное значение. В противном случае функция должна принимать целое значение. Через $Aц, Bц$ и $Cц$ - целые части значений a, b, c , операции И и ИЛИ - поразрядные. Необходимо вводить с клавиатуры значения $a, b, c, X_{\text{нач}}, X_{\text{кон}}, dX$. Результат работы программы вывести на экран.

Пример выполнения практического задания №4.

Задание. Необходимо составить программу для вычисления таблицы значений функции $Z(x, y) = \min\{\max\{\sin x, \sin y\}, x, y\}$ при $3 \leq x \leq 9.1$ с шагом 4.3; при $3 \leq y \leq 6$ с шагом 0.85. Определите наибольшее отрицательное значение функции, наименьшее положительное значение функции, произведение ненулевых значений функции, количество значений функции, больших 1 или меньших -1.

Решение.

1) Словесный алгоритм.

Так как переменные x и y принимали несколько значений необходимо было использовать вложенный цикл. Во внешнем цикле счетчик x изменялся от 3 до 9.9 с шагом 4.3, а во внутреннем счетчик y – от 3 до 6 с шагом 0.85. Т. к. переменные

были вещественного типа использовался тип `double`. Для постановки условий необходимо было воспользоваться оператором `if` во внутреннем цикле. Для использования математических функций как синус, косинус и т.д. была подключена библиотека `cmath`.

2) Схема алгоритма. Схема алгоритма программы представлена на рисунке 24.

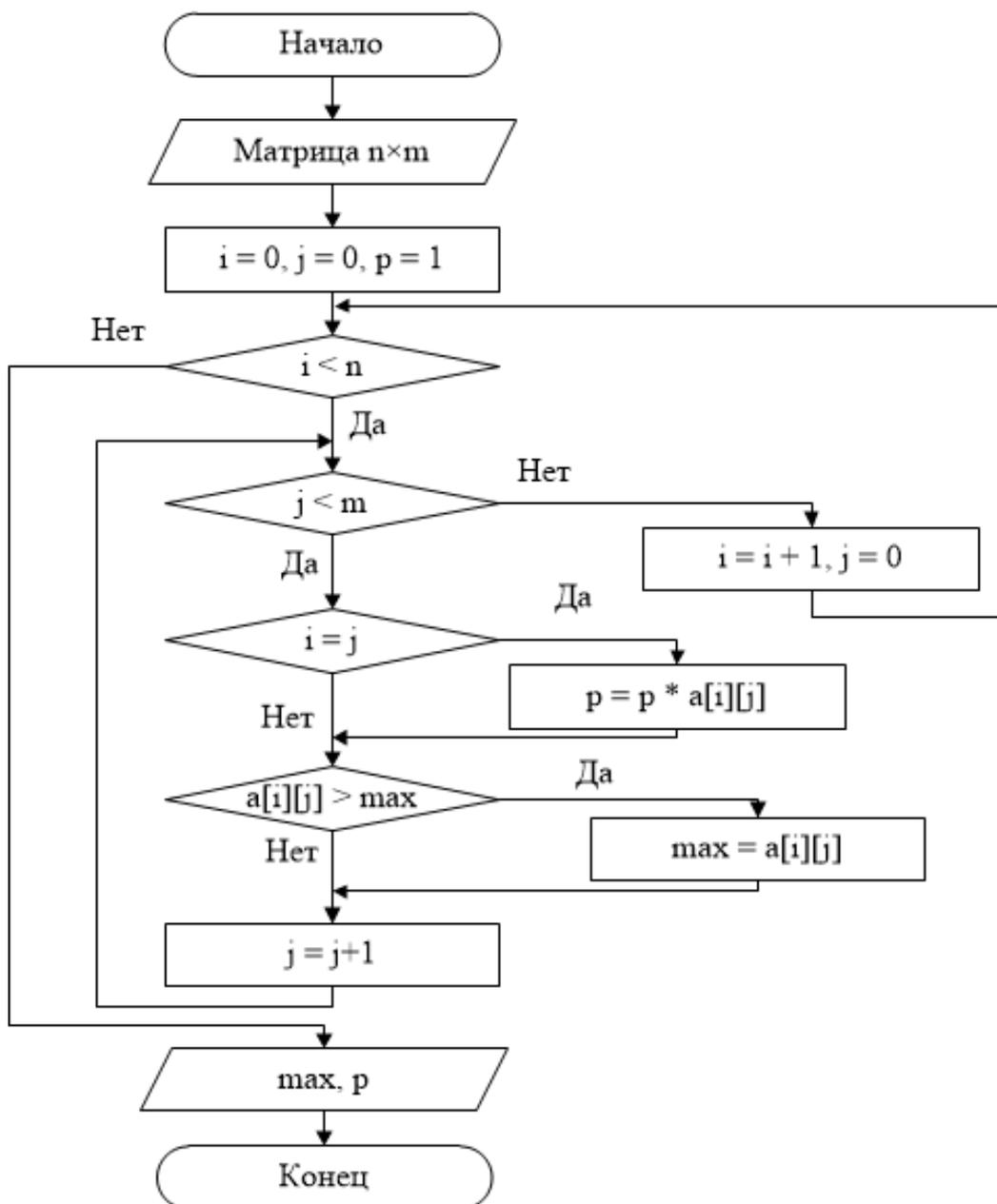


Рисунок 24 - Схема алгоритма программы реализации циклических вычислительных процессов

3) Код программы.

```
#include "pch.h"
```

```
#include <iostream>
```

```
#include <iomanip>
```

```
#include <cmath> // Подключение библиотеки для
```

использования математических функций

```
using namespace std;
```

```
int main()
```

```
{
```

```
setlocale(LC_ALL, "Rus");
```

```
double x, y, dx = 1.3, dy = 2, min = 1, max = -1, minpol = 10E10, maxotr = -  
10E10, proizv=1;
```

```
int count=0;
```

```
cout << setw(10) << "x" << setw(10) << "y" << setw(10) << "z" << endl; for (x  
= 3; x < 9.1; x = x + dx) // Внешний цикл. Счетчик цикла - x
```

```
{
```

```
for (y = 3; y < 6; y = y + dy) // Начало внутреннего цикла. Счетчик -
```

y

```
{
```

```
if (sin(x) > sin(y)) max = sin(x);
```

```
else max = sin(y);
```

```
if ((x < y) && (x < max)) min = x; //Поиск минимального
```

```
if ((y < x) && (y < max)) min = y; // значения
```

```
if ((max < x) && (max < y)) min = max; // функции Z(x,y)
```

```
if ((min < minpol) && (min > 0)) minpol = min; //Минимальное
```

положительное

```
if ((min > maxotr) && (min < 0)) maxotr = min; //Максимальное
```

отрицательное

```

        if (min != 0) proizv *= min;           //Произведение ненулевых
элементов
        if ((min > 1) || (min < -1)) count++; // Кол-во элементов от -1 до 1
        cout << setw(10) << x << setw(10) << y << setw(10) << min << endl; }
    }
    cout << "Наибольшее отрицательное значение функции: " << maxotr <<
endl;
    cout << "Наименьшее положительное значение функции: " << minpol <<
endl;
    cout << "Произведение ненулевых значений функции: " << proizv <<
endl;
    cout << "Количество значений функции > 1 или < -1: " << count << endl; }

```

4) Реализация программного кода. Реализация программного кода представлена на рисунке 25.

```

Консоль отладки Microsoft Visual Studio
    x      y      z
    3      3      0.14112
    3      5      0.14112
    4.3    3      0.14112
    4.3    5     -0.916166
    5.6    3      0.14112
    5.6    5     -0.631267
    6.9    3      0.57844
    6.9    5      0.57844
    8.2    3      0.940731
    8.2    5      0.940731
Наибольшее отрицательное значение функции: -0.631267
Наименьшее положительное значение функции: 0.14112
Произведение ненулевых значений функции: 6.79185e-05
Количество значений функции > 1 или < -1: 0
Чтобы закрыть это окно, нажмите любую клавишу...

```

Рисунок 25 - Результат работы программы реализации циклических алгоритмов

5 Практическое занятие №5. Работа с одномерными массивами в языке C++

Цель: Закрепление теоретических знаний и получение практических навыков написания программ по обработке одномерных массивов на языке C++

Задание. Разработать программу на языке C++ с использованием одномерных массивов. Размерности массивов следует задавать именованными константами.

Вариант 1. Необходимо составить программу, в которой должен быть задан одномерный массив, состоящей из n вещественных элементов. Требуется вычислить:

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

Упорядочить по возрастанию элементы массива. Результат работы программы вывести на экран.

Вариант 2. Необходимо составить программу, в которой должен быть задан одномерный массив, состоящей из n вещественных элементов. Требуется вычислить:

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

Упорядочить по убыванию элементы массива. Результат работы программы вывести на экран.

Вариант 3. Необходимо составить программу, в которой должен быть задан одномерный массив, состоящей из n вещественных элементов. Требуется вычислить:

1) произведение элементов массива с четными номерами;

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

Далее необходимо преобразовать массив. Сначала должны располагаться все положительные элементы массива, а затем все отрицательные элементы массива (элементы, равные 0, считать положительными). Результат работы программы вывести на экран.

Вариант 4. Необходимо составить программу, в которой должен быть задан одномерный массив, состоящей из n вещественных элементов. Требуется вычислить:

1) сумму элементов массива с нечетными номерами;

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

Далее необходимо сжать массив, удалив из него все элементы, модуль которых не превышает 1. Освободившиеся в конце массива элементы заполнить нулями. Результат работы программы вывести на экран.

Вариант 5. Необходимо составить программу, в которой должен быть задан одномерный массив, состоящей из n вещественных элементов. Требуется вычислить:

1) максимальный элемент массива;

2) сумму элементов массива, расположенных до последнего положительного элемента.

Далее необходимо сжать массив, удалив из него все элементы, модуль которых находится в интервале $[a,b]$. Освободившиеся в конце массива элементы заполнить нулями. Результат работы программы вывести на экран.

Вариант 6. Необходимо составить программу, в которой должен быть задан одномерный массив, состоящей из n вещественных элементов. Требуется вычислить:

1) минимальный элемент массива;

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

Далее необходимо преобразовать массив. Сначала должны располагаться все элементы, равные нулю, а затем все остальные элементы массива. Результат работы программы вывести на экран.

Вариант 7. Необходимо составить программу, в которой должен быть задан одномерный массив, состоящей из n вещественных элементов. Требуется вычислить:

1) номер максимального элемента массива;

2) произведение элементов массива, расположенных между первым и вторым нулевыми элементами.

Далее необходимо преобразовать массив. В первой половине массива должны располагаться элементы, стоявшие в нечетных позициях, а во второй половине массива - элементы, стоявшие в четных позициях. Результат работы программы вывести на экран.

Вариант 8. Необходимо составить программу, в которой должен быть задан одномерный массив, состоящей из n вещественных элементов. Требуется вычислить:

1) номер минимального элемента массива;

2) сумму элементов массива, расположенных между первым и вторым отрицательными элементами.

Далее необходимо преобразовать массив. Сначала должны располагаться все элементы, модуль которых не превышает 1, а затем все остальные элементы массива. Результат работы программы вывести на экран.

Вариант 9. Необходимо составить программу, в которой должен быть задан одномерный массив, состоящей из n вещественных элементов. Требуется вычислить:

1) максимальный по модулю элемент массива;

2) сумму элементов массива, расположенных между первым и вторым положительными элементами.

Далее необходимо преобразовать массив. Элементы массива, равные нулю, должны располагаться после всех остальных элементов массива. Результат работы программы вывести на экран.

Вариант 10. Необходимо составить программу, в которой должен быть задан одномерный массив, состоящей из n вещественных элементов. Требуется вычислить:

1) минимальный по модулю элемент массива;

2) сумму модулей элементов массива, расположенных после первого элемента, равного нулю.

Далее необходимо преобразовать массив. В первой его половине должны располагаться элементы, стоявшие в четных позициях, а элементы, стоявшие в нечетных позициях во второй половине. Результат работы программы вывести на экран.

Вариант 11. Необходимо составить программу, в которой должен быть задан одномерный массив, состоящей из n вещественных элементов. Требуется вычислить:

1) номер минимального по модулю элемента массива;

2) сумму модулей элементов массива, расположенных после первого отрицательного элемента.

Далее необходимо сжать массив, удалив из него все элементы, величина которых находится в интервале $[a,b]$. Освободившиеся в конце массива элементы заполнить нулями. Результат работы программы вывести на экран.

Вариант 12. Необходимо составить программу, в которой должен быть задан одномерный массив, состоящей из n вещественных элементов. Требуется вычислить:

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

Далее необходимо преобразовать массив. Сначала должны располагаться все элементы, целая часть которых лежит в интервале $[a,b]$, а затем все остальные элементы массива. Результат работы программы вывести на экран.

Вариант 13. Необходимо составить программу, в которой должен быть задан одномерный массив, состоящей из n вещественных элементов. Требуется вычислить:

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

Далее необходимо упорядочить элементы массива по убыванию модулей элементов. Результат работы программы вывести на экран.

Вариант 14. Необходимо составить программу, в которой должен быть задан одномерный массив, состоящей из n вещественных элементов. Требуется вычислить:

- 1) количество элементов массива, равных 0;
- 2) сумму элементов массива, расположенных после минимального элемента.

Далее необходимо упорядочить элементы массива по возрастанию модулей элементов. Результат работы программы вывести на экран.

Вариант 15. Необходимо составить программу, в которой должен быть задан одномерный массив, состоящей из n вещественных элементов. Требуется вычислить:

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

Далее необходимо преобразовать массив. Сначала должны располагаться все отрицательные элементы, а затем все положительные (элементы, равные 0, считать положительными). Результат работы программы вывести на экран.

Вариант 16. Необходимо составить программу, в которой должен быть задан одномерный массив, состоящей из n вещественных элементов. Требуется вычислить:

1) количество отрицательных элементов массива;

2) сумму модулей элементов массива, расположенных после минимального по модулю элемента.

Далее необходимо все отрицательные элементы массива заменить их квадратами и упорядочить элементы массива по возрастанию. Результат работы программы вывести на экран.

Вариант 17. Необходимо составить программу, в которой должен быть задан одномерный массив, состоящей из n вещественных элементов. Требуется вычислить:

1) количество положительных элементов массива;

2) сумму элементов массива, расположенных после последнего элемента, равного нулю.

Далее необходимо преобразовать массив. Сначала должны располагаться все элементы, целая часть которых не превышает 1, а затем все остальные элементы массива. Результат работы программы вывести на экран.

Вариант 18. Необходимо составить программу, в которой должен быть задан одномерный массив, состоящей из n вещественных элементов. Требуется вычислить:

1) количество элементов массива, меньших K ;

2) сумму целых частей элементов массива, расположенных после последнего отрицательного элемента.

Далее необходимо преобразовать массив. Сначала должны располагаться все элементы, отличающиеся от максимального элемента не более чем на 19%, а затем все остальные элементы массива. Результат работы программы вывести на экран.

Вариант 19. Необходимо составить программу, в которой должен быть задан одномерный массив, состоящей из n вещественных элементов. Требуется вычислить:

1) произведение отрицательных элементов массива;

2) сумму положительных элементов массива, расположенных до максимального элемента.

Затем изменить порядок следования элементов в массиве, на обратный порядок. Результат работы программы вывести на экран.

Вариант 20. Необходимо составить программу, в которой должен быть задан одномерный массив, состоящей из n вещественных элементов. Требуется вычислить:

1) произведение положительных элементов массива;

2) сумму элементов массива, расположенных до минимального элемента.

Стоящие на четных местах элементы, упорядочить по возрастанию, а затем и элементы, стоящие на нечетных местах. Результат работы программы вывести на экран.

Пример выполнения практического задания №5.

Задание.

Необходимо составить программу, в которой должен быть задан одномерный массив, состоящей из пяти вещественных элементов. Требуется вычислить:

1) сумму четных элементов массива;

2) максимальный и минимальный элементы массива.

Решение.

1) Словесный алгоритм.

Для решения данной задачи использовался цикл for (нумерация элементов в массиве начинается с нуля номер первого элемента – 0, последнего – 4 значит счетчик цикла $i = 0$, а условие выхода цикла $i < 5$). Для поиска максимального и минимального элементов, а так же для проверки на четность было необходимо воспользоваться оператор if.

2) Схема алгоритма. Схема алгоритма программы представлена на рисунке 26.

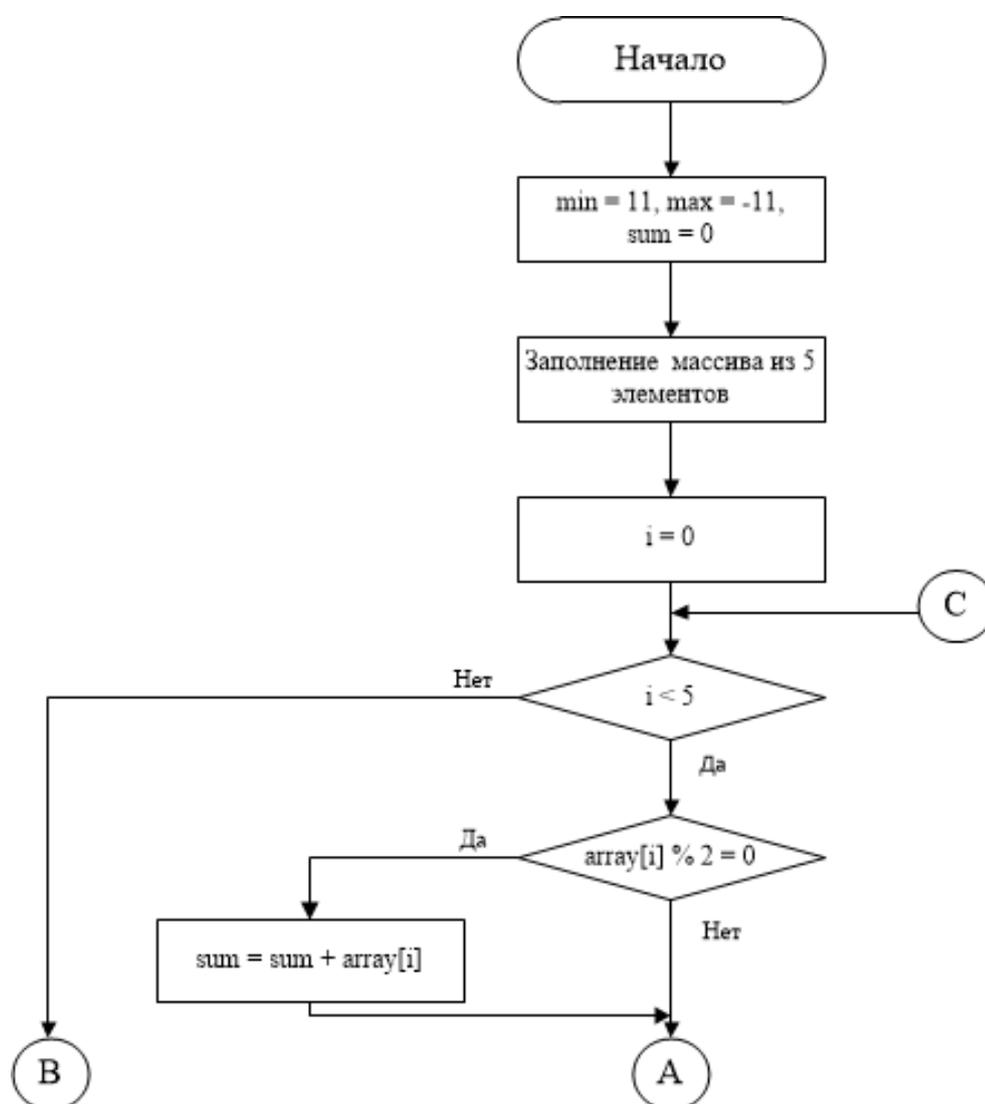


Рисунок 26 - Схема алгоритма программы работы с одномерными массивами

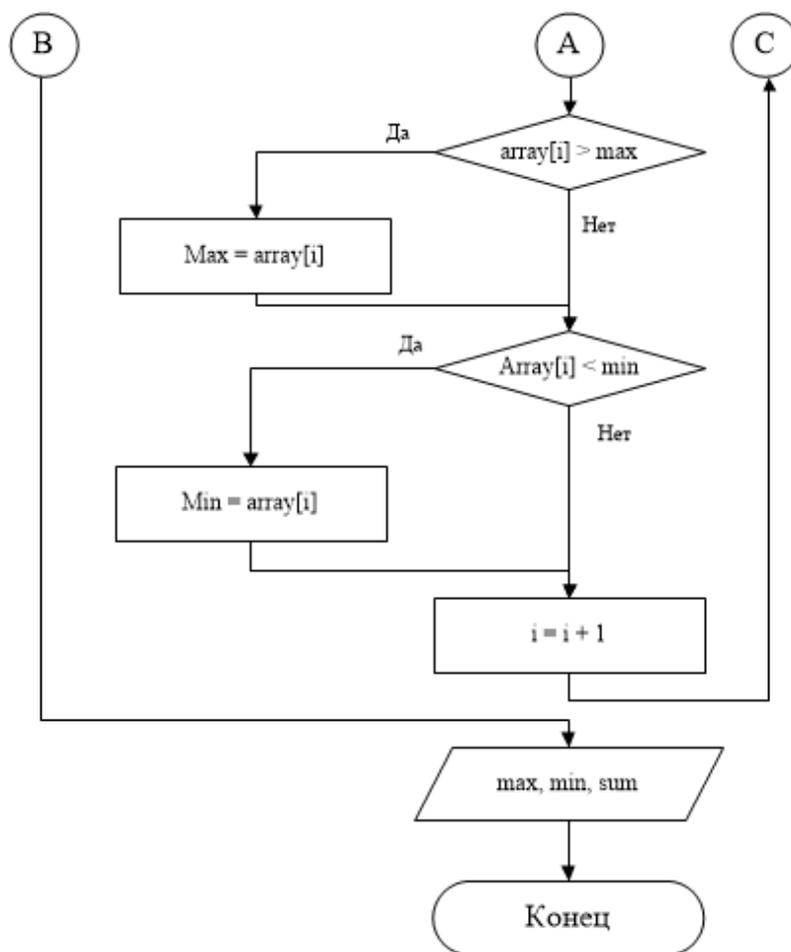


Рисунок 26, Лист 48.

3) Код программы.

```
#include "pch.h"
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    setlocale(LC_ALL, "Rus");
```

```
    int sum = 0, min = 11, max = -11;
```

```
    int array[5];
```

```
    // Инициализация массива из 5
```

элементов

```
    cout << "Введите массив: ";
```

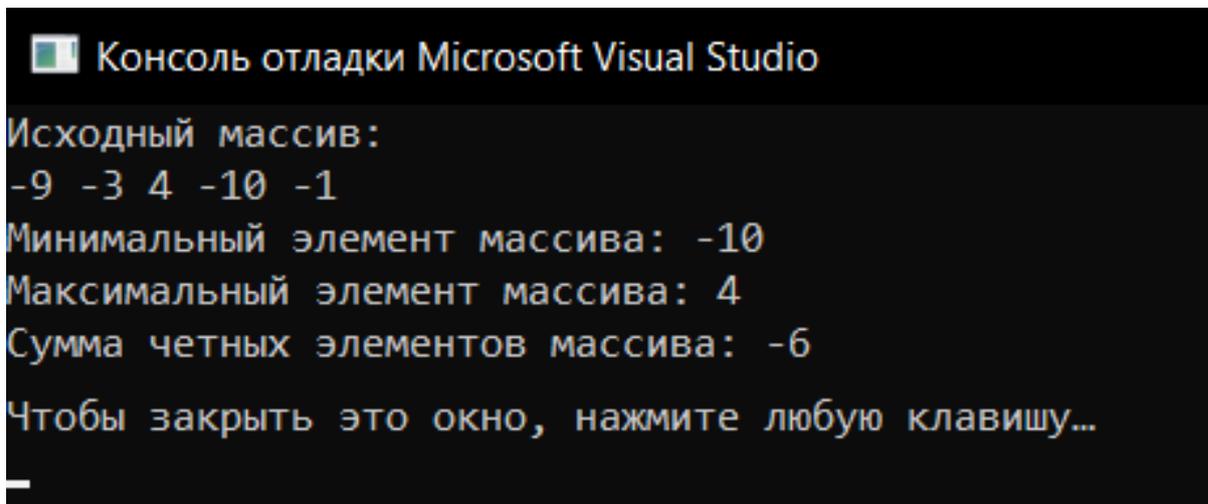
```
    for (int i = 0; i < 5; i++)
```

```
cin >> array[i]; // Ввод элементов массива с
```

клавиатуры

```
cout << "Исходный массив: " << endl;
for (int i = 0; i < 5; i++)
{
if ((array[i] % 2) == 0) sum += array[i]; // Сумма четных элементов
if (array[i] > max) max = array[i]; // Максимальный элемент
if (array[i] < min) min = array[i]; // Минимальный элемент
cout << array[i] << " "; }
cout << endl;
cout << "Минимальный элемент массива: " << min << endl;
cout << "Максимальный элемент массива: " << max << endl;
cout << "Сумма четных элементов массива: " << sum << endl; }
```

4) Реализация программного кода. Реализация программного кода представлена на рисунке 27.



```
Консоль отладки Microsoft Visual Studio
Исходный массив:
-9 -3 4 -10 -1
Минимальный элемент массива: -10
Максимальный элемент массива: 4
Сумма четных элементов массива: -6
Чтобы закрыть это окно, нажмите любую клавишу...
```

Рисунок 27 – Результат работы программы с одномерными массивами

6 Практическое задание №6. Обработка двумерных массивов с помощью инструментов языка C++

Цель: Закрепление теоретических знаний и получение практических навыков написания программ по обработке двумерных массивов на языке C++.

Задание. Разработать программу на языке C++ с использованием двумерных массивов.

Вариант 1. Необходимо составить программу, в которой задана целочисленная прямоугольная матрица. Требуется определить:

1) количество строк матрицы, не содержащих ни одного нулевого элемента матрицы;

2) максимальное из чисел, встречающихся в заданной матрице более одного раза.

Результат работы программы вывести на экран.

Вариант 2. Необходимо составить программу, в которой задана целочисленная прямоугольная матрица. Требуется определить количество столбцов матрицы, не содержащих ни одного нулевого элемента матрицы.

Характеристикой строки целочисленной матрицы называется сумма ее положительных четных элементов матрицы. Необходимо расположить строки заданной матрицы в соответствии с ростом характеристик. Результат работы программы вывести на экран.

Вариант 3. Необходимо составить программу, в которой задана целочисленная прямоугольная матрица. Требуется определить:

1) количество столбцов матрицы, содержащих хотя бы один нулевой элемент матрицы;

2) номер строки, в которой находится самая длинная серия одинаковых элементов матрицы.

Результат работы программы вывести на экран.

Вариант 4. Необходимо составить программу, в которой задана целочисленная квадратная матрица. Требуется определить:

1) произведение элементов в тех строках матрицы, которые не содержат отрицательных элементов матрицы;

2) максимум среди сумм элементов диагоналей матрицы, параллельных главной диагонали матрицы.

Результат работы программы вывести на экран.

Вариант 5. Необходимо составить программу, в которой задана целочисленная квадратная матрица. Требуется определить:

1) сумму элементов в тех столбцах матрицы, которые не содержат отрицательных элементов;

2) минимум среди сумм модулей элементов диагоналей матрицы, параллельных побочной диагонали матрицы.

Результат работы программы вывести на экран.

Вариант 6. Необходимо составить программу, в которой задана целочисленная прямоугольная матрица. Требуется определить:

1) сумму элементов в тех строках матрицы, которые содержат хотя бы один отрицательный элемент матрицы;

2) номера строк и столбцов всех седловых точек матрицы.

(Матрица A имеет седловую точку A_{ij} , если A_{ij} является минимальным элементом в i -й строке и максимальным в j -м столбце). Результат работы программы вывести на экран.

Вариант 7. Необходимо составить программу, в которой задана матрица размером 8 на 8. Требуется в данной матрице найти такие n , что n -я строка матрицы совпадает с n -м столбцом. Затем найти сумму элементов в тех строках,

которые содержат хотя бы один отрицательный элемент матрицы. Результат работы программы вывести на экран.

Вариант 8. Необходимо составить программу, в которой характеристикой столбца целочисленной матрицы называется сумма модулей его отрицательных нечетных элементов матрицы. Требуется расположить столбцы заданной матрицы в соответствии с ростом характеристик. Далее найти сумму элементов в тех столбцах, которые содержат хотя бы один отрицательный элемент матрицы. Результат работы программы вывести на экран.

Вариант 9. Необходимо составить программу, в которой соседями элемента M_{ij} в матрице, например, будут элементы M_{kl} с $i-1 \leq k \leq i+1, j-1 \leq l \leq j+1, (k, l) \neq (i, j)$. Операция сглаживания матрицы дает новую матрицу того же размера, каждый элемент которой получается как среднее арифметическое имеющихся соседей соответствующего элемента исходной матрицы. Требуется построить результат сглаживания заданной вещественной матрицы размером 10 на 10. Затем в сглаженной матрице найти сумму модулей элементов, расположенных ниже главной диагонали матрицы. Результат работы программы вывести на экран.

Вариант 10. Необходимо составить программу, в которой элемент матрицы называется локальным минимумом, если он строго меньше всех имеющихся у него соседей. Требуется подсчитать количество локальных минимумов заданной матрицы размером 10 на 10. Затем найти сумму модулей элементов, расположенных выше главной диагонали матрицы. Результат работы программы вывести на экран.

Вариант 11. Необходимо составить программу, в которой коэффициенты системы линейных уравнений должны быть заданы в виде прямоугольной матрицы. Далее привести систему к треугольному виду с помощью допустимых преобразований. Затем найти количество строк матрицы, среднее арифметическое элементов которых меньше заданной величины. Результат работы программы вывести на экран.

Вариант 12. Необходимо составить программу, в которой уплотнить заданную матрицу, удаляя из нее строки и столбцы, заполненные нулями. Далее найти номер первой из строк, содержащих хотя бы один положительный элемент матрицы. Результат работы программы вывести на экран.

Вариант 13. Необходимо составить программу, в которой осуществить циклический сдвиг элементов прямоугольной матрицы на n элементов вправо или вниз (в зависимости от введенного режима). n может быть больше количества элементов в строке или столбце матрицы. Результат работы программы вывести на экран.

Вариант 14. Необходимо составить программу, в которой осуществить циклический сдвиг элементов квадратной матрицы размерности $A \times B$ вправо на n элементов. Элементы первой строки должны сдвигаться в последний столбец сверху вниз, из него - в последнюю строку справа налево, из нее - в первый столбец снизу вверх, из него - в первую строку; для остальных элементов - аналогично. Результат работы программы вывести на экран.

Вариант 15. Необходимо составить программу, в которой должна быть задана целочисленная прямоугольная матрица. Далее найти номер первого из столбцов, содержащих хотя бы один нулевой элемент матрицы. Характеристика строки целочисленной матрицы – это сумма отрицательных четных элементов матрицы. Далее расположить строки в соответствии с убыванием характеристик. Результат работы программы вывести на экран.

Вариант 16. Необходимо составить программу, в которой упорядочить строки целочисленной прямоугольной матрицы по возрастанию количества одинаковых элементов в каждой строке матрицы. Далее найти номер первого из столбцов, не содержащих ни одного отрицательного элемента матрицы. Результат работы программы вывести на экран.

Вариант 17. Необходимо составить программу, в которой путем перестановки элементов квадратной вещественной матрицы добиться того, чтобы ее максимальный элемент находился в левом верхнем углу, а следующий по

величине элемент матрицы должен находиться в позиции (2,2), следующий по величине - в позиции (3,3) и т. д. Необходимо заполнить, таким образом, всю главную диагональ матрицы. Далее найти номер первой из строк, не содержащих ни одного положительного элемента матрицы. Результат работы программы вывести на экран.

Вариант 18. Необходимо составить программу, в которой должна быть задана целочисленная прямоугольная матрица. Требуется определить:

- 1) количество строк, содержащих хотя бы один нулевой элемент матрицы;
- 2) номер столбца, в котором находится самая длинная серия одинаковых элементов матрицы.

Результат работы программы вывести на экран.

Вариант 19. Необходимо составить программу, в которой должна быть задана целочисленная квадратная матрица. Требуется определить:

- 1) сумму элементов матрицы в тех строках, которые не содержат отрицательных элементов;
- 2) минимум среди сумм элементов диагоналей матрицы, параллельных главной диагонали матрицы.

Результат работы программы вывести на экран.

Вариант 20. Необходимо составить программу, в которой должна быть задана целочисленная прямоугольная матрица. Требуется определить:

- 1) количество отрицательных элементов матрицы в тех строках, которые содержат хотя бы один нулевой элемент матрицы;
- 2) номера строк и столбцов всех седловых точек матрицы.

(Матрица M имеет седловую точку M_{ij} , если M_{ij} является минимальным элементом в i -й строке и максимальным в j -м столбце). Результат работы программы вывести на экран.

Пример выполнения практического задания №6.

Задание.

Необходимо составить программу, в которой должна быть задана целочисленная квадратная матрица 4×4 . Требуется определить:

1) произведение элементов главной диагонали матрицы и найти максимальный элемент матрицы.

Решение.

1) Словесный алгоритм.

Для решения данной задачи использовались вложенные циклы. Внешний цикл рассматривал строки матрицы, а внутренний ее столбцы. Так же как и в случае одномерного массива, нумерация элементов в двумерном массиве начиналась с нуля. В квадратной матрице, если номер строки совпадал с номером столбца, рассматриваемая ячейка являлась элемент главной диагонали.

2) Схема алгоритма. Схема алгоритма программы представлена на рисунке 28.

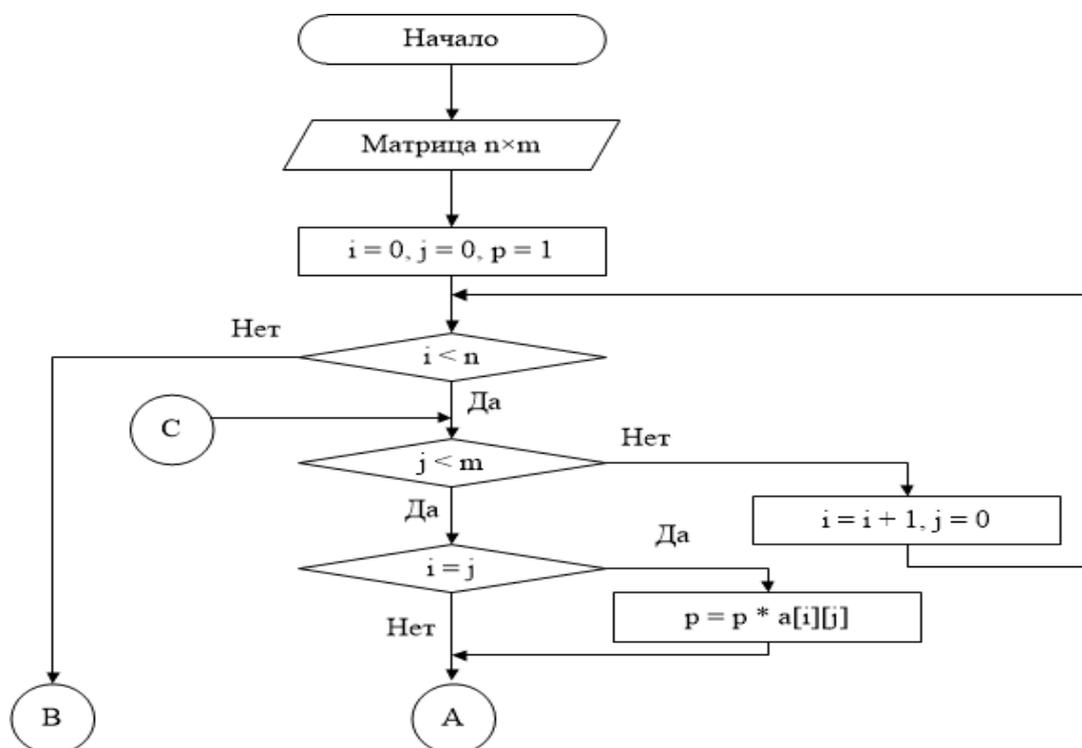


Рисунок 28 - Схема алгоритма программы работы с двумерными массивами

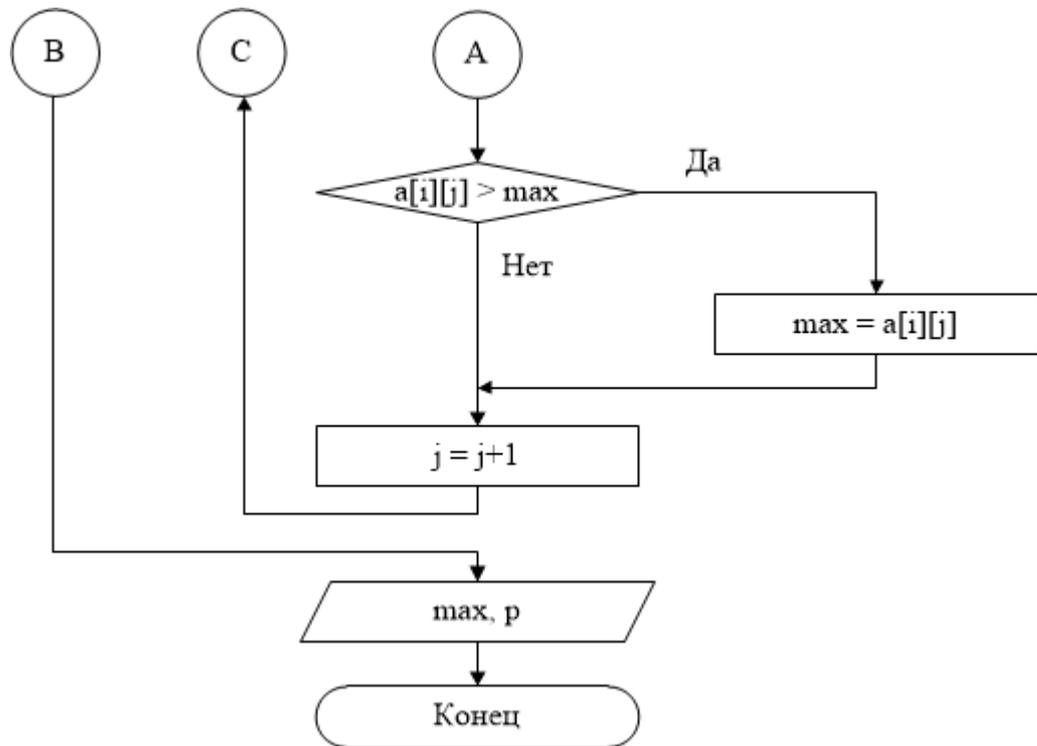


Рисунок 28, Лист 56.

3) Код программы.

```
#include "pch.h"
```

```
#include <iostream>
```

```
#include <iomanip>
```

```
using namespace std;
```

```
int main()
```

```
{ setlocale(LC_ALL, "Rus");
```

```
int n, m, p = 1, max = -11;
```

```
int a[4][4]; // Инициализация матрицы 4×4
```

```
cout << "Введите элементы матрицы: " << endl;
```

```
for (int i = 0; i < 4; i++) // Внешний цикл. Просматривает строки
```

матрицы

```
{ for (int j = 0; j < 4; j++)//Внутренний
```

цикл.

Просматривает столбцы

```
{ cin >> a[i][j]; } // Ввод элементов
```

```

    }
    cout << "Исходная матрица: " << endl;
    for (int i = 0; i < 4; i++)
    {
        for (int j = 0; j < 4; j++)
        {
            cout << setw(5) << a[i][j];    //      Вывод
элементов матрицы
            if (i == j) p *= a[i][j];    //      Произведение      элементов      главной
диагонали
            if (a[i][j] > max) max = a[i][j]; }    //Максимальный элемент
            cout << endl;    }
        cout << endl;
        cout << "Произведение элементов главной диагонали = " << p <<
endl;
        cout << "Максимальный элемент матрицы = " << max; }

```

4) Реализация программного кода. Реализация программного кода представлена на рисунке 28.

```

Консоль отладки Microsoft Visual Studio
Введите элементы матрицы:
2 1 1 1
1 2 1 1
1 1 2 1
1 1 1 2
Исходная матрица:
    2    1    1    1
    1    2    1    1
    1    1    2    1
    1    1    1    2
Произведение элементов главной диагонали = 16
Максимальный элемент матрицы = 2
Чтобы закрыть это окно, нажмите любую клавишу...

```

Рисунок 29 – Результат работы программы работы с двумерными массивами

7 Практическое занятие №7. Работа с динамической памятью.

Указатели

Цель: Закрепление теоретических знаний и получение практических навыков написания программ с использованием указателей в языке C++, работы с динамическими массивами. Приобретение навыков работы с указателями, освоить допустимые операции над указателями.

Задание. Разработать программу на языке C++ с использованием массивов динамической памяти.

Вариант 1. Необходимо составить программу, имеющую матрицу размера $A \times B$. Далее получить массив C , присвоив его n -му элементу значение 0, если все элементы n -го столбца матрицы нулевые, и значение 1 – в противном случае. Результат работы программы вывести на экран.

Вариант 2. Необходимо составить программу, имеющую матрицу размера $A \times B$. Далее получить массив C , присвоив его n -му элементу значение 1, если элементы n -ой строки матрицы упорядочены по убыванию, и значение 0 – в противном случае. Результат работы программы вывести на экран.

Вариант 3. Необходимо составить программу, имеющую матрицу размера $A \times B$. Далее получить массив C , присвоив его n -му элементу значение 1, если n -я строка матрицы симметрична, и значение 0 – в противном случае. Результат работы программы вывести на экран.

Вариант 4. Необходимо составить программу, имеющую матрицу размера $A \times B$. Далее определить количество «особых» элементов матрицы. Элемент считать «особым», если он больше суммы остальных элементов своего столбца. Результат работы программы вывести на экран.

Вариант 5. Необходимо составить программу, имеющую матрицу размера $A \times B$. Далее определить количество «особых» элементов матрицы. Элемент считать «особым», если в строке слева от него находятся элементы, меньшие его, а справа элементы большие. Результат работы программы вывести на экран.

Вариант 6. Необходимо составить программу, имеющую матрицу размера $A \times B$. Далее определить количество различных элементов матрицы (т.е. повторяющиеся элементы матрицы считать один раз). Результат работы программы вывести на экран.

Вариант 7. Необходимо составить программу, имеющую матрицу размера $A \times B$. Далее упорядочить строки матрицы по возрастанию и первых элементов. Результат работы программы вывести на экран.

Вариант 8. Необходимо составить программу, имеющую матрицу размера $A \times B$. Далее упорядочить строки матрицы по возрастанию суммы их элементов. Результат работы программы вывести на экран.

Вариант 9. Необходимо составить программу, имеющую матрицу размера $A \times B$. Далее упорядочить столбцы матрицы по возрастанию их наименьших элементов. Результат работы программы вывести на экран.

Вариант 10. Необходимо составить программу, и определить, является ли заданная квадратная матрица n -го порядка симметричной относительно побочной диагонали матрицы. Результат работы программы вывести на экран.

Вариант 11. Необходимо составить программу, имеющую матрицу размера $A \times B$. Далее вывести на экран все седловые точки матрицы. Элемент матрицы называется седловой точкой, если он является наименьшим в своей строке и одновременно наибольшим в своем столбце или наоборот. Результат работы программы вывести на экран.

Вариант 12. Необходимо составить программу, имеющую матрицу размера $A \times B$. Далее переставить строки матрицы так, чтобы на главной диагонали матрицы были расположены элементы, наибольшие по абсолютной величине. Результат работы программы вывести на экран.

Вариант 13. Необходимо составить программу, имеющую матрицу размером $A \times B$. Далее найти максимальный элемент, лежащий ниже побочной диагонали матрицы, и минимальный элемент, лежащий выше главной диагонали матрицы. Результат работы программы вывести на экран.

Вариант 14. Необходимо составить программу, в которой в матрице размером $A \times B$ поменять местами строку, содержащую элемент с наибольшим значением матрицы, со строкой, содержащей элемент с наименьшим значением матрицы. Результат работы программы вывести на экран.

Вариант 15. Необходимо составить программу, в которой из матрицы размером $A \times B$ получить матрицу размером $(A-1) \times (B-1)$ путем удаления из исходной матрицы строки и столбца, на пересечении которых расположен элемент с наибольшим по модулю значением. Результат работы программы вывести на экран.

Вариант 16. Необходимо составить программу, в которой сформировать одномерный массив. Затем из массива удалить элемент с заданным номером. После, добавить элемент с заданным номером. Результат работы программы вывести на экран.

Вариант 17. Необходимо составить программу, в которой сформировать одномерный массив. Затем из массива удалить элемент с заданным ключом. После, добавить элемент с заданным ключом. Результат работы программы вывести на экран.

Вариант 18. Необходимо составить программу, в которой сформировать одномерный массив. Затем из массива удалить N элементов, начиная с заданного номера, номер задавать с клавиатуры. После, добавить элемент с заданным ключом. Результат работы программы вывести на экран.

Вариант 19. Необходимо составить программу, в которой сформировать одномерный массив. Затем из массива удалить элемент с заданным номером, номер задавать с клавиатуры. После, добавить N элементов, начиная с заданного номера. Результат работы программы вывести на экран.

Вариант 20. Необходимо составить программу, в которой сформировать одномерный массив. Затем из массива удалить N элементов, начиная с заданного номера, номер задавать с клавиатуры. После, добавить N элементов, начиная с заданного номера. Результат работы программы вывести на экран.

Пример выполнения практического задания №7.

Задание.

Необходимо составить программу, имеющую матрицу размером $m \times n$. Далее получить массив В, элементами которого являются максимальные элементы каждой строки матрицы. Результат работы программы вывести на экран.

Решение.

1) Словесный алгоритм.

Так как, количество строк и столбцов в матрице заранее не известно, для решения данной задачи использовались указатели. Был объявлены массивы с помощью указателей. Для нахождения максимального элемента в строке применялся оператор `if`.

2) Схема алгоритма программы представлена на рисунке 30.

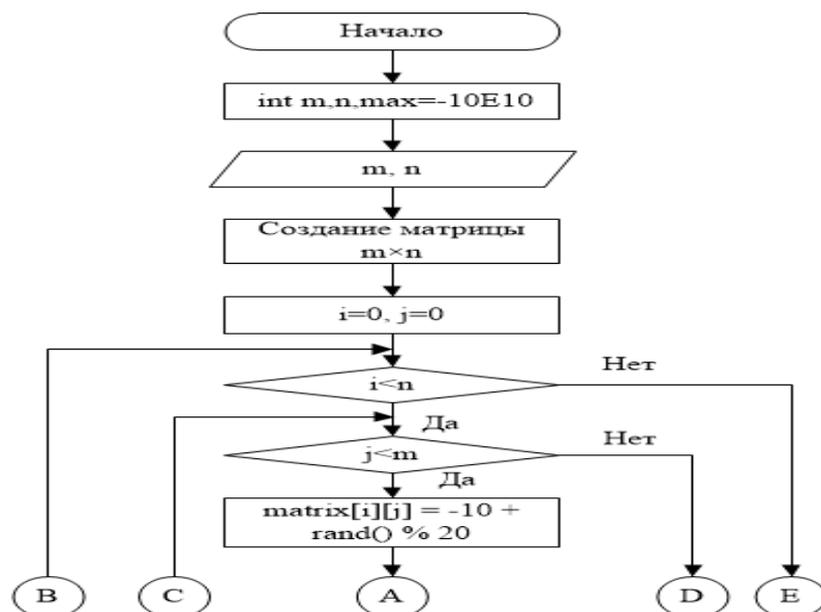


Рисунок 30, Лист 1 - Схема алгоритма программы работы с динамической памятью

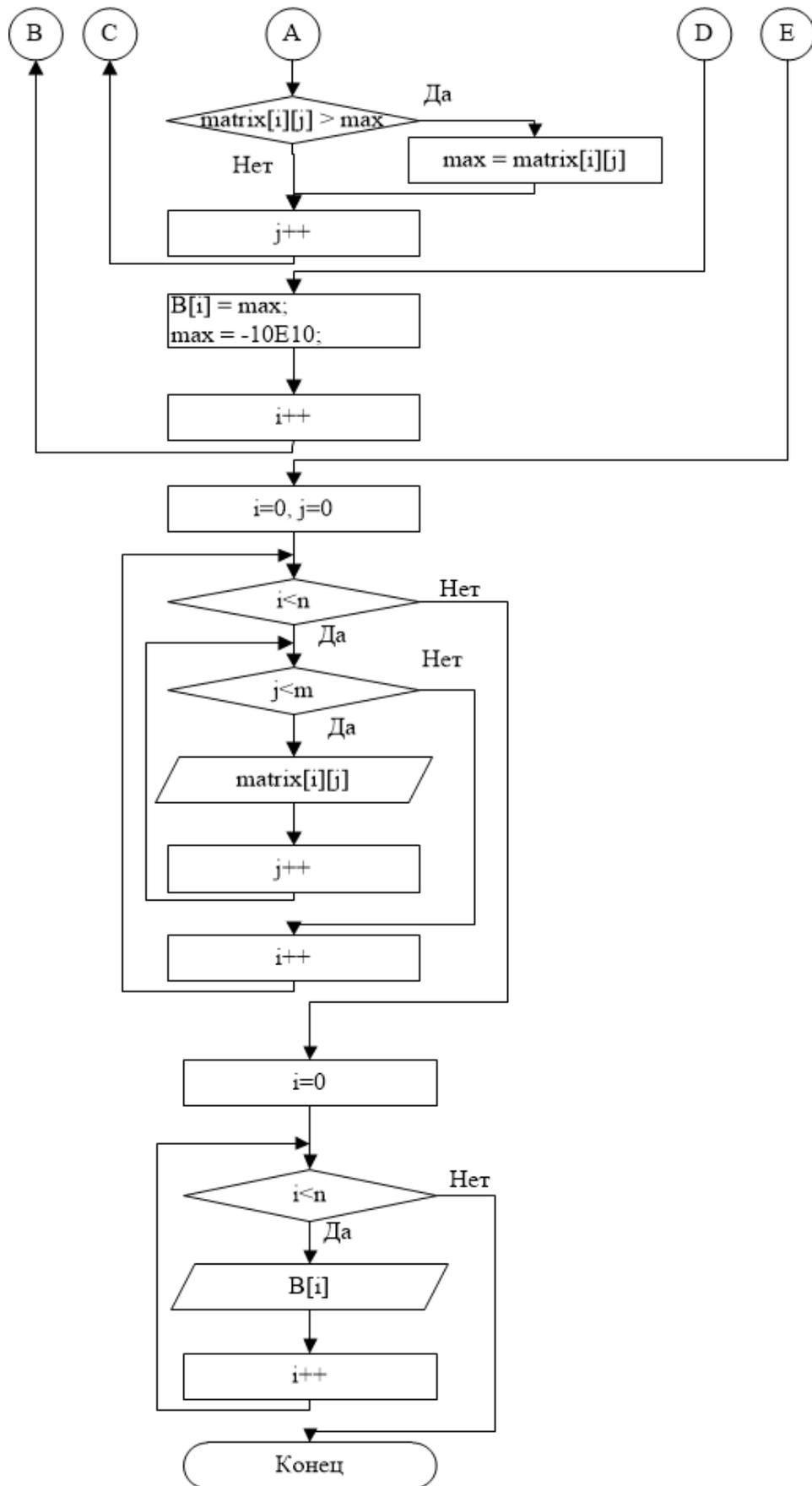


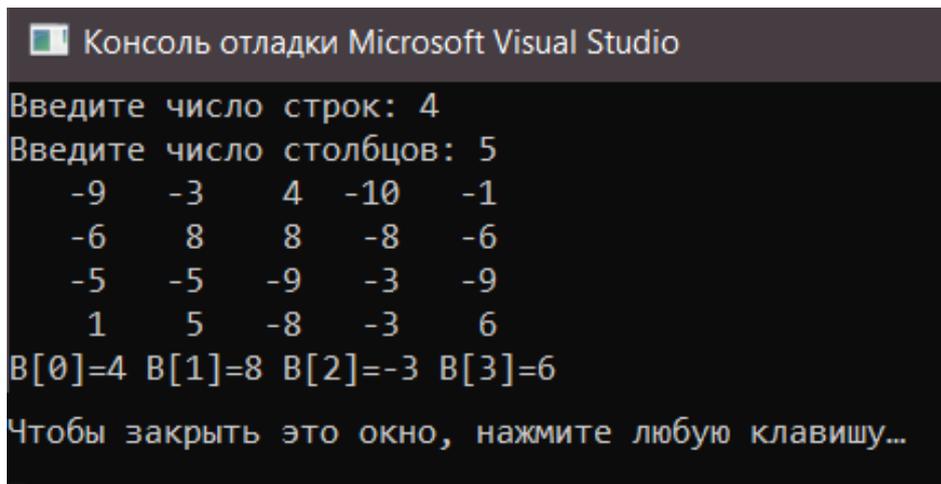
Рисунок 30, Лист 2

3) Код программы.

```
#include "pch.h"
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{
    setlocale(LC_ALL, "Rus"); // Установка русского языка в консоли
    int n, m, max = -10E10;
    cout << "Введите число строк: ";
    cin >> n;
    cout << "Введите число столбцов: ";
    cin >> m;
    int* B = new int[n]; // Объявление массива B из n элементов
    int** matrix = new int*[n]; //Объявление матрицы
    for (int i = 0; i < n; i++) matrix[i] = new int[m]; // размером n×m
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            matrix[i][j] = -10 + rand() % 20; //Заполнение матрицы
случайными значениями от -10 до 10
        }
        if (matrix[i][j] > max) max = matrix[i][j]; } //Поиск максимального
элемента в строке
        B[i] = max; // Заполнение массива B
        max = -10E10; } // Возвращение значения переменной
max
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++) {
            cout << setw(5) << matrix[i][j]; } // Вывод матрицы
        cout << endl; }
}
```

```
for (int i =0;i<n;i++)  
cout << "B[" << i << "]= " << B[i] << " "; }  
// Вывод массива B
```

4) Реализация программного кода. Реализация программного кода представлена на рисунке 31.



The screenshot shows the 'Консоль отладки Microsoft Visual Studio' window. The text displayed is as follows:

```
Введите число строк: 4  
Введите число столбцов: 5  
-9 -3 4 -10 -1  
-6 8 8 -8 -6  
-5 -5 -9 -3 -9  
1 5 -8 -3 6  
B[0]=4 B[1]=8 B[2]=-3 B[3]=6  
Чтобы закрыть это окно, нажмите любую клавишу...
```

Рисунок 31 – Результат работы программы с использованием динамической памяти

8 Практическое занятие №8. Работа со строками символов в языке C++

Цель: Закрепление теоретических знаний и получение практических навыков написания программ по обработке строк в языке C++. Изучение приёмов объявления и обработки строковых величин средствами языка C++.

Задание. Разработать программу на языке C++ с использованием массивов символов (строк).

Вариант 1. Необходимо составить программу, в которой проверить является ли строка палиндромом. Палиндром – это выражение, которое читается, одинаково слева направо и справа налево.

Вариант 2. Необходимо составить программу, в которой напечатать самое длинное и самое короткое слово в строке.

Вариант 3. Необходимо составить программу, в которой напечатать все слова в строке, которые не содержат гласных букв.

Вариант 4. Необходимо составить программу, в которой напечатать все слова в строке, которые содержат по одной цифре.

Вариант 5. Необходимо составить программу, в которой напечатать все слова в строке, которые совпадают с ее первым словом.

Вариант 6. Необходимо составить программу, в которой преобразовать строку так, чтобы сначала в строке были напечатаны только буквы, а затем только цифры, при этом не менять порядок следования символов в строке.

Вариант 7. Необходимо составить программу, в которой преобразовать строку так, чтобы все буквы строки были отсортированы по возрастанию.

Вариант 8. Необходимо составить программу, в которой преобразовать строку так, чтобы все цифры строки были отсортированы по убыванию.

Вариант 9. Необходимо составить программу, в которой преобразовать строку так, чтобы все слова строки стали идентификаторами, а слова, которые состоят только из цифр необходимо удалить.

Вариант 10. Необходимо составить программу, в которой напечатать все слова-палиндромы, которые есть в строке.

Вариант 11. Необходимо составить программу, в которой преобразовать строку так, чтобы в ее начале были записаны слова, содержащие только цифры, а затем слова, которые содержат только буквы. А после слова, которые содержат и буквы и цифры.

Вариант 12. Необходимо составить программу, в которой преобразовать строку так, чтобы все слова строки были напечатаны наоборот.

Вариант 13. Необходимо составить программу, в которой преобразовать строку так, чтобы буквы каждого слова строки были отсортированы по возрастанию.

Вариант 14. Необходимо составить программу, в которой преобразовать строку так, чтобы цифры каждого слова строки были отсортированы по убыванию.

Вариант 15. Необходимо составить программу, в которой преобразовать строку так, чтобы в строке остались только слова, содержащие буквы и цифры, остальные слова необходимо удалить.

Вариант 16. Необходимо составить программу, в которой определить какое слово встречается в строке чаще всего.

Вариант 17. Необходимо составить программу, в которой определить какие слова встречаются в строке по одному или два раза.

Вариант 18. Необходимо составить программу, в которой все слова строки должны начинаться с буквы. Затем отсортировать слова строки в алфавитном порядке.

Вариант 19. Необходимо составить программу, в которой все слова строки должны начинаться с цифры. Затем отсортировать слова строки по убыванию.

Вариант 20. Необходимо составить программу, в которой удалить из строки все слова, не являющиеся идентификаторами.

Пример выполнения практического задания №8.

Задание.

Необходимо составить программу, в которой проверить является ли введенная строка палиндромом. Палиндром – это выражение, которое читается одинаково слева направо и справа налево.

Решение.

1) Словесный алгоритм.

Палиндром – это выражение, которое читается одинаково слева направо и справа налево, поэтому сравнивались начальные и конечные буквы в слове. Затем вторая и предпоследняя буквы и т.д.

Для ввода строки был использован массив символов. Тип этого массива – char. Чтобы узнать является ли полученная строка палиндромом сравнивались первый и последний элементы в массиве, второй и предпоследний и т.д. Для этого применялся цикл for.

2) Схема алгоритма программы представлена на рисунке 32.

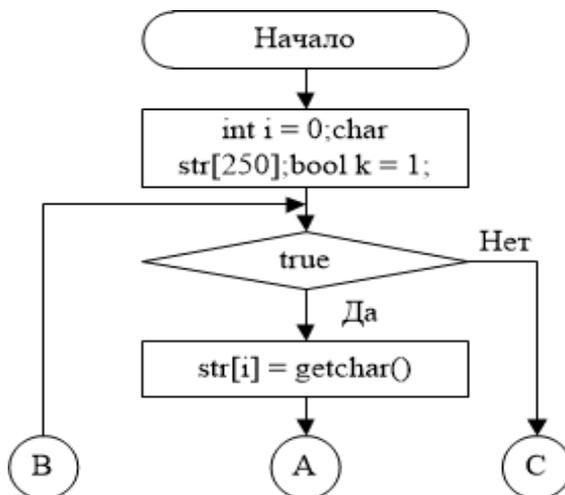


Рисунок 32, Лист 1 - Схема алгоритма программы работы со строками

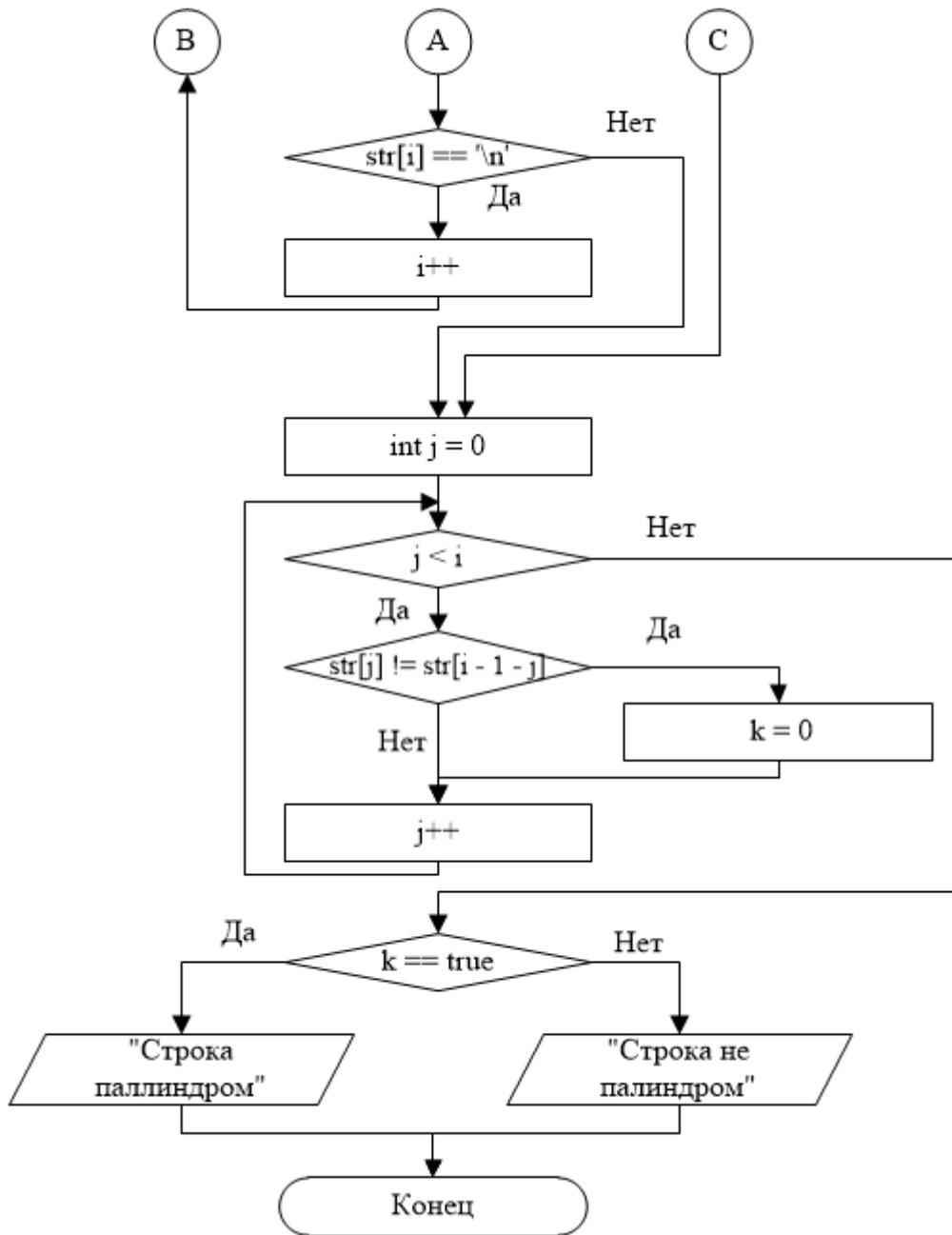


Рисунок 32, Лист 2

3) Код программы.

```
#include "pch.h"
```

```
#include <iostream>
```

```
using namespace std;
```

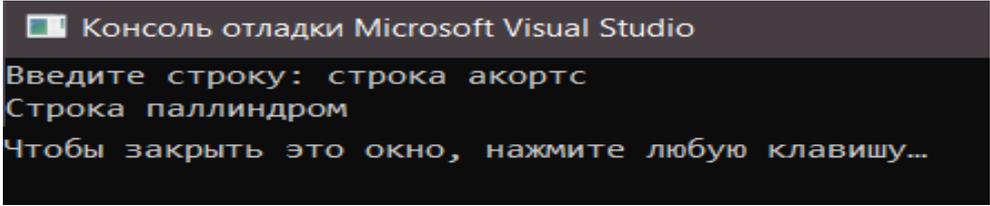
```
int main()
```

```

{   bool k = 1;
    setlocale(LC_ALL, "Rus"); // Установка русского языка в консоли
    int i = 0;
    char str[250];           // Массив типа char
    cout << "Введите строку: ";
    while (true)
    {   str[i] = getchar();   // Ввод символов
        if (str[i] == '\n') break; // Прекращение ввода после нажатия
enter
        i++; }
    for (int j = 0; j < i ; j++)
    {   if (str[j] != str[i -1- j]) k = 0; } // Проверка на палиндромность
    if (k == true ) cout << "Строка палиндром";
    else cout << "Строка не палиндром"; }

```

4) Реализация программного кода. Реализация программного кода представлена на рисунках 33, 34.

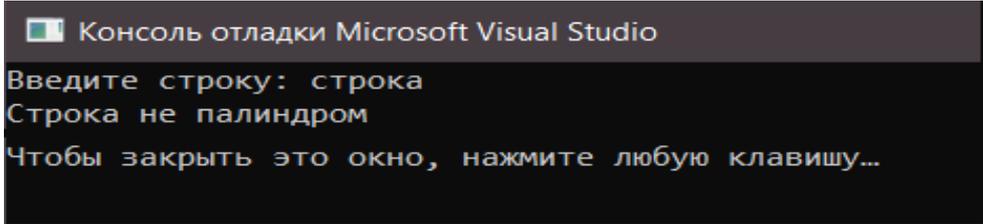


```

Консоль отладки Microsoft Visual Studio
Введите строку: строка акорты
Строка паллиндром
Чтобы закрыть это окно, нажмите любую клавишу...

```

Рисунок 33 – Результат работы программы. Строка является палиндромом



```

Консоль отладки Microsoft Visual Studio
Введите строку: строка
Строка не палиндром
Чтобы закрыть это окно, нажмите любую клавишу...

```

Рисунок 34 – Результат работы со строками. Строка не является палиндромом

Заключение

Таким образом, в представленных методических указаниях по практическим занятиям изложены задания для выполнения работ по разработке программ на языке C++. Представлены задания для создания приложений в инструментальной среде MS Visual Studio, методика работы с алгоритмами языка C++, реализацией циклических вычислительных процессов, одномерными и многомерными массивами, с динамической памятью, указателями, со строками символов.

В результате изучения методических указаний студенты смогут самостоятельно разрабатывать программы на языке высокого уровня C++.

Список использованных источников

- 1 Павловская, Т.А. С/С++. Программирование на языке высокого уровня / Т. А. Павловская. - СПб.: Питер, 2004. – 461 с.
- 2 Павловская, Т.А. С/С ++. Структурное программирование: Практикум / Т.А. Павловская, Ю.А. Щупак. - СПб.: Питер, 2007. - 239 с.: ил.
- 3 Павловская, Т. А., С++. Объектно-ориентированное программирование: Практикум. / Т.А. Павловская, Ю.А. Щупак - СПб.: Питер, 2006. - 265 с: ил.
- 4 Кольцов, Д.М. 100 примеров на Си. / Д.М. Кольцов. СПб.: “Наука и техника”, 2017 - 256 с.
- 5 Доусон, М. Изучаем С++ через программирование игр. / М. Доусон. - СПб.: “Питер”, 2016. - 352.
- 6 Седжвик, Р. Фундаментальные алгоритмы на С++. Анализ/Структуры данных/Сортировка/Поиск: Пер. с англ. / Роберт Седжвик. - Киев: Издательство “Диасофт”, 2001. - 688с.
- 7 Сиддхартха, Р. Освой самостоятельно С++ за 21 день. / Сиддхартха, Р. - М.: SAMS, 2013. - 651 с.
- 8 Стивен, П. Язык программирования С++. Лекции и упражнения: пер. с англ. / П. Стивен. 6-е изд. - М.: ООО "И.Д. Вильямс", 2012. - 1248 с.
- 9 Черносивитов, А. Visual С++: руководство по практическому изучению / А. Черносивитов . - СПб. : Питер, 2002. - 528 с. : ил.
- 10 Страуструп, Б. Дизайн и эволюция языка С++. / Б. Страуструп - М.: ДМК, 2000. - 448 с.
- 11 Мейерс, С. Эффективное использование С++. / С. Мейерс. - М.: ДМК, 2000. - 240 с.
- 12 Бадд Т. Объектно-ориентированное программирование в действии. - СПб: Питер, 1997. - 464 с.
- 13 Лаптев В.В. С ++. Объектно-ориентированное программирование: Учебное пособие.- СПб.: Питер, 2008. - 464 с.: ил.

14 Страуструп, Б. Язык программирования С++. / Б. Страуструп. Режим доступа: http://8361.ru/6sem/books/Straustrup-Yazyk_programmirovaniya_c.pdf.

15 Керниган, Б., Язык программирования Си. / Б. Керниган, Д. Ритчи. Режим доступа: http://cpp.com.ru/kr_cbook/index.html.

16 Герберт, Шилдт: С++ базовый курс. / Герберт Шилдт. Режим доступа: https://www.bsuir.by/m/12_100229_1_98220.pdf.

17 Богуславский, А.А. Основы программирования на языке Си++. / А.А. Богуславский, С.М. Соколов. Режим доступа: http://www.ict.edu.ru/ft/004246/cpp_p1.pdf.

18 Линский, Е. Основы С++. / Е. Линский. Режим доступа: <https://www.lektorium.tv/lecture/13373>.

Приложение А

Структура отчета

Отчет по практическому заданию должен был выполнен на листах формата А4 и оформлен в соответствии с требованиями СТО.

Отчет по практическому заданию должен содержать следующие пункты:

- 1) Задание на практическое занятие.
- 2) Этапы решения задачи.
- 3) Словесный алгоритм.
- 4) Схема алгоритма.
- 5) Код программы.
- 6) Реализация программного кода.