**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение

высшего образования

«Санкт-Петербургский политехнический университет Петра Великого»

(ФГАОУ ВО «СПбПУ»)

**Институт среднего профессионального образования**

**ОТЧЕТ**

**по учебной практике (по профилю специальности)**

по профессиональному модулю ПМ.04 «Сопровождение и обслуживание программного обеспечения компьютерных систем»

(код и наименование)

Специальность 09.02.07 «Информационные системы и программирование»

(код и наименование специальности)

Студент(ка) II курса 24290907/3091 группы

Жданов Иван Андреевич

(ФИО полностью)

Место прохождения практики: ФГАОУ ВО СПбПУ Петра Великого   
Институт СПО, учебно-вычислительный центр, пр. Энгельса д.23

(наименование и адрес организации)

Период прохождения практики

с «08» декабря 2025 г. по «27» декабря 2025 г.

Руководитель практики Курылева А. А.

от учебной организации (подпись) (расшифровка подписи)

Итоговая оценка по практике \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

М.П.

Санкт-Петербург

2025г.

СОГЛАСОВАНО

Председатель ПЦК

\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_ 202\_\_ г.

**ЗАДАНИЕ**

**на учебную практику (по профилю специальности)**

по профессиональному модулю ПМ.04 «Сопровождение и обслуживание программного обеспечения компьютерных систем»

(код и наименование)

Специальность 09.02.07 «Информационные системы и программирование»

(код и наименование специальности)

Студент(ка) II курса 24290907/3091 группы

Жданов Иван Андреевич

(ФИО полностью)

Место прохождения практики: ФГАОУ ВО СПбПУ Петра Великого  
Институт СПО, учебно-вычислительный центр, пр. Энгельса д.23

(наименование и адрес организации)

Период прохождения практики

с «08» декабря 2025 г. по «27» декабря 2025 г.

**Виды работ, обязательные для выполнения** (переносится из программы, соответствующего ПМ):

1. Персонализация интегрированной среды разработки Visual Studio Community 2022
2. Отладка в IDE Visual Studio Community 2022
3. Обеспечение качества кода
4. Упаковка приложения

**Индивидуальное задание:**

Задание выдал с «8» декабря 2025 г. Курылева А. А. (подпись) (Ф.И.О.)

Задание получил с «8» декабря 2025 г.

(подпись) (Ф.И.О.)

**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение

высшего образования

«Санкт-Петербургский политехнический университет Петра Великого»

(ФГАОУ ВО «СПбПУ»)

**Институт среднего профессионального образования**

**ДНЕВНИК**

**прохождения учебной практики**

**(по профилю специальности)**

по профессиональному модулю ПМ.04 «Сопровождение и обслуживание программного обеспечения компьютерных систем»

(код и наименование)

Специальность 09.02.07 «Информационные системы и программирование»

(код и наименование специальности)

Студент(ка) II курса 24290907/3091 группы

Жданов Иван Андреевич

(ФИО полностью)

Место прохождения практики: ФГАОУ ВО СПбПУ Петра Великого  
Институт СПО, учебно-вычислительный центр, пр. Энгельса д.23

(наименование и адрес организации)

Период прохождения практики

с «08» декабря 2025 г. по «27» декабря 2025 г.

Руководитель с места Курылева А. А.

прохождения практики (подпись) (расшифровка подписи)

Санкт-Петербург

2025 г.

**Содержание дневника**

|  |  |  |
| --- | --- | --- |
| Дата | Виды выполненных работ и заданий по программе практики | Подпись руководителя практики |
| **1** | **2** | **3** |
| 8.12.25 | Настройка меню и панели инструментов |  |
| 9.12.25 | Параметры текстового редактора |  |
| 10.12.25 | Создание кода и текстового шаблона |  |
| 11.12.25 | Навигация по коду с помощью отладчика |  |
| 12.12.25 | Использование точек останова |  |
| 13.12.25 | Управление исключениями с помощью отладчика |  |
| 15.12.25 | Использование файлов дампа  Использование средств профилирования |  |
| 16.12.25 | Тестирование в IDE Visual Studio Community 2022 |  |
| 17.12.25 | Документирование кода с помощью XML-комментариев |  |
| 18.12.25 | Изменение кода в соответствии с соглашением о кодировании |  |
| 19.12.25 | Анализ качества кода |  |
| 20.12.25 | Основы системы контроля версий Git |  |
| 22.12.25 | Технологические подходы программирования |  |
| 23.12.25 | Методология программирования |  |
| 24.12.25 | Работа с реестром ОС Windows |  |
| 25.12.25 | Упаковка классического приложения вручную |  |
| 26.12.25 | Упаковка приложения с помощью Visual Studio Package Installer |  |
| 27.12.25 | Создание виртуальной машины |  |

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Введение пишется ко всему отчету, не к каждой работе отдельно

# **Практическая работа №1 «Настройка внешнего вида и функциональности среды разработки Visual Studio»**

**Цель:** Освоить настройку интерфейса и функциональности Visual Studio для комфортной работы.

## **1.1. Параметры текстового редактора**

В настройках текстового редактора необходимо настроить автоматическое форматирование при вводе } и ;. Для этого необходимо в разделе «Средства → Параметры → Текстовый редактор» выбрать язык программирования, открыть общие настройки и установить флаги на соответствующих пунктах. Эти настройки позволят автоматически форматировать код во время ввода указанных символов (Рисунок 1).



Рисунок 1 *-* Автоматическое форматирование при вводе } и ;.

Для изменения интервалов в разделе «Средства → Параметры → Текстовый редактор» выбираем раздел «Форматирование» устанавливаем флаг на пункт «Вставлять пробел между скобками со списком параметров» (Рисунок 2).

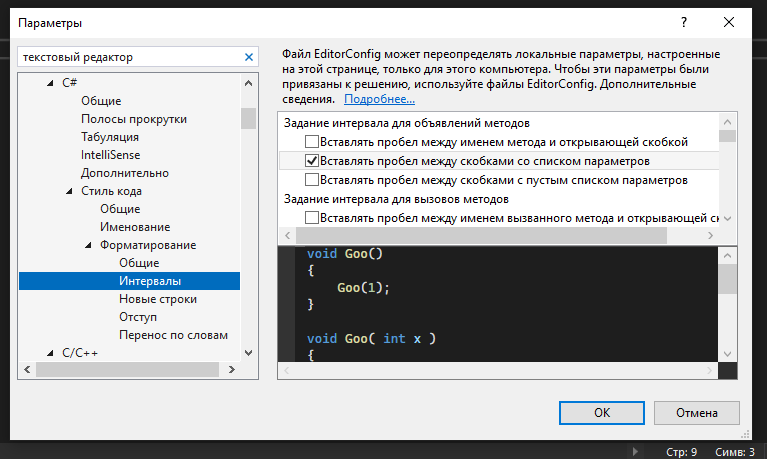


Рисунок 2 - Изменение интервалов

Для того, чтобы включить нумерацию строк, в разделе «Средства → Параметры → Текстовый редактор» выбираем раздел «Общие» устанавливаем флаг на пункт «Номера строк» (Рисунок 3).

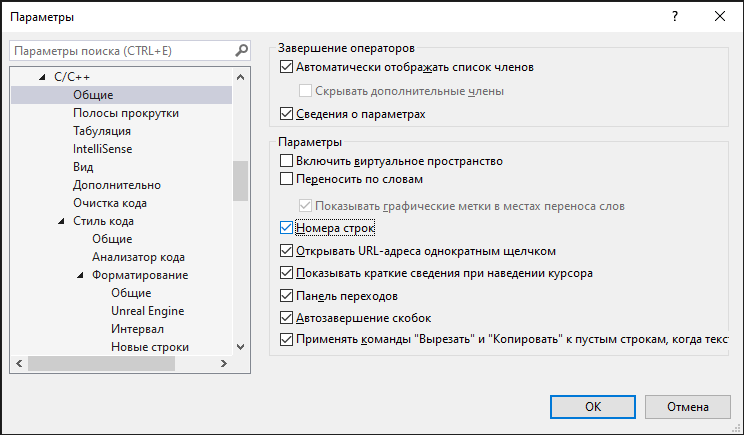


Рисунок 3 – Номера строк

## **1.2. Изменение шрифтов и цветов**

Для изменения шрифта и цвета всплывающих подсказок в разделе «Средства → Параметры → Окружение → Шрифты и цвета» в всплывающем окне выбрать нужный шрифт и цвет (Рисунок 4).

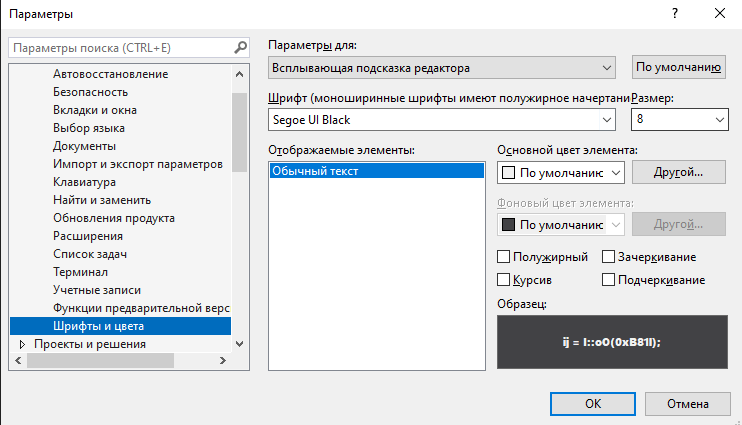


Рисунок 4 – Всплывающие подсказки

Для изменения шрифта и цвета текста среды в разделе «Средства → Параметры → Окружение → Шрифты и цвета» в всплывающем окне выбрать нужный шрифт (Рисунок 5).

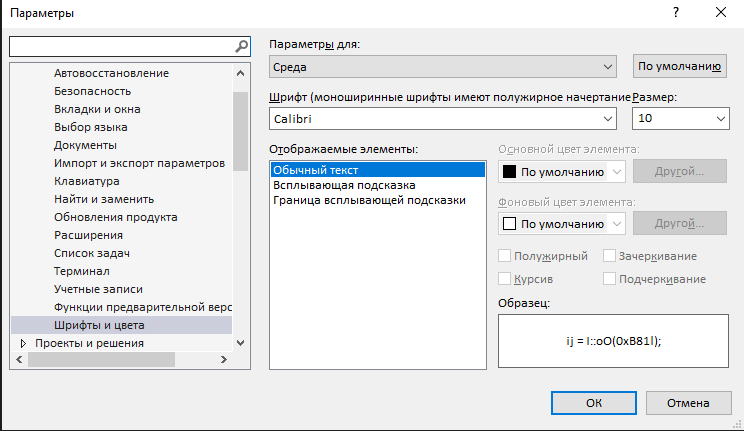


Рисунок 5 – Изменение шрифта и цвета всплывающих подсказок

Для изменения шрифта и цвета номера строки в разделе «Средства → Параметры → Окружение → Шрифты и цвета» в всплывающем окне выбрать нужный шрифт и цвет (Рисунок 6).

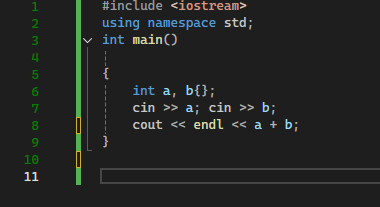


Рисунок 6 – Изменение шрифт и цвета номера строк

Для изменения шрифта и цвета комментариев в разделе «Средства → Параметры → Окружение → Шрифты и цвета» в всплывающем окне выбираем нужный шрифт и цвет (Рисунок 7).

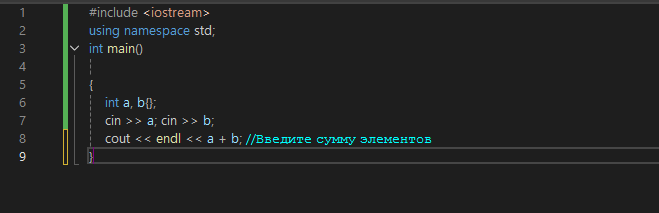


Рисунок 7 – Изменение шрифта и цвета комментария

Для изменения шрифта и цвета строковых литералов в разделе «Средства → Параметры → Окружение → Шрифты и цвета» в всплывающем окне выбираем нужный шрифт и цвет (Рисунок 8).

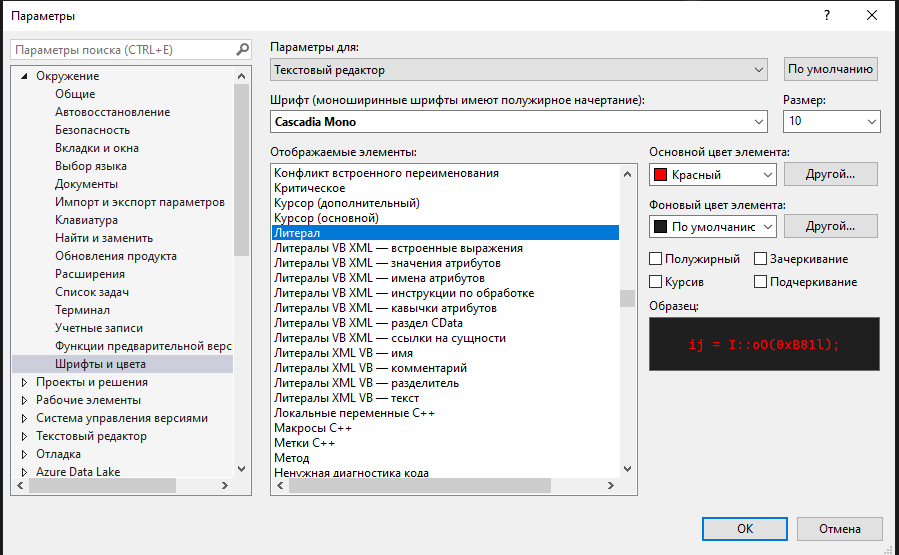


Рисунок 8 - Изменение шрифта и цвета строковых литералов

Для изменения шрифта и цвета чисел в разделе «Средства → Параметры → Окружение → Шрифты и цвета» в всплывающем окне выбираем нужный шрифт и цвет (Рисунок 9).

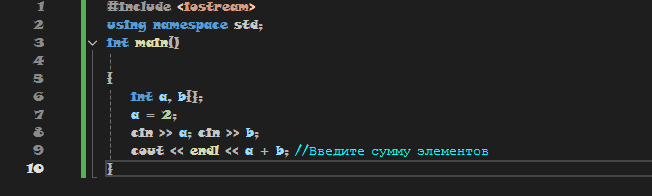


Рисунок 9 – Изменение шрифта и цвета числа

## **1.3. Настройка меню и панели инструментов**

Для создания своей панели инструментов «Средства → Настройка → Панель инструментов», добавить команды «Средства → Настройка → Команды» (Рисунок 10).

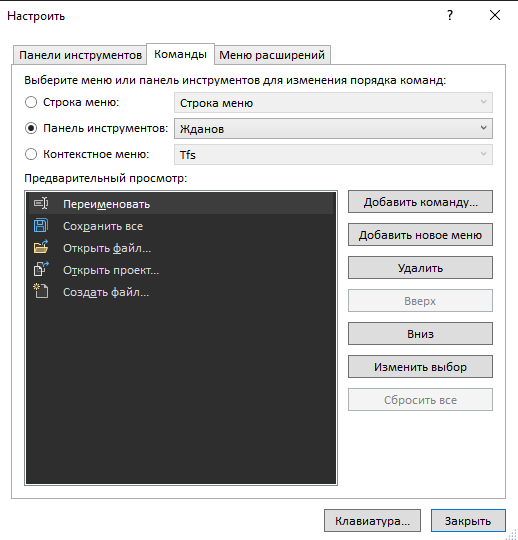


Рисунок 10 – Добавление команд в панель инструментов

Для назначения горячих клавиш через «Средства → Параметры → Окружение → Клавиатура» (Рисунок 11-15)

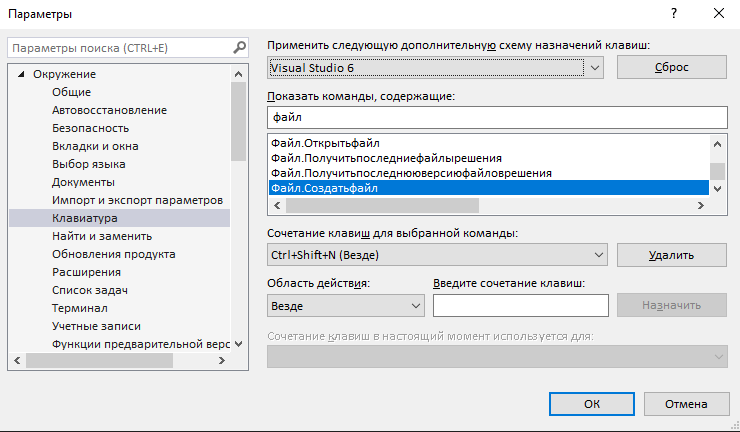


Рисунок 11 –Назначение горячей клавиши

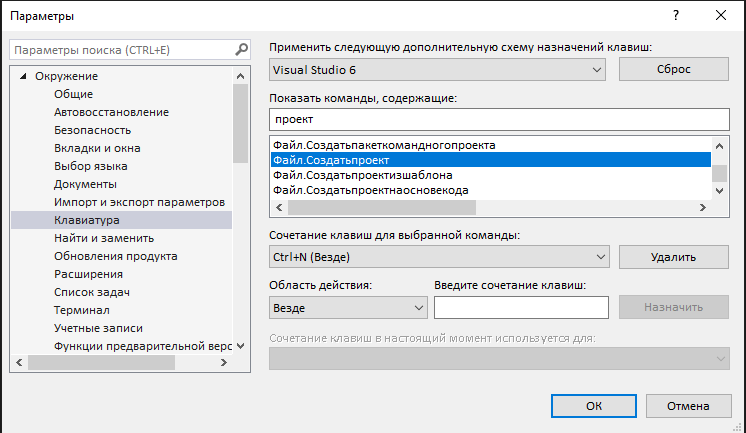


Рисунок 12 – Назначение горячей клавиши

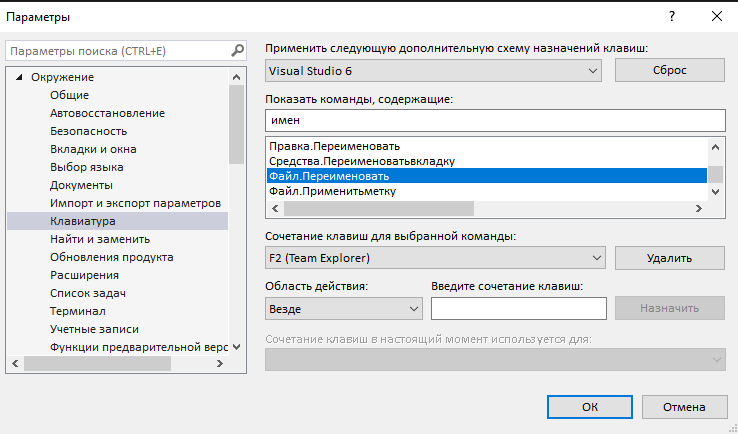


Рисунок 13 – Назначение горячей клавиши



Рисунок 14 – Назначение горячей клавиши

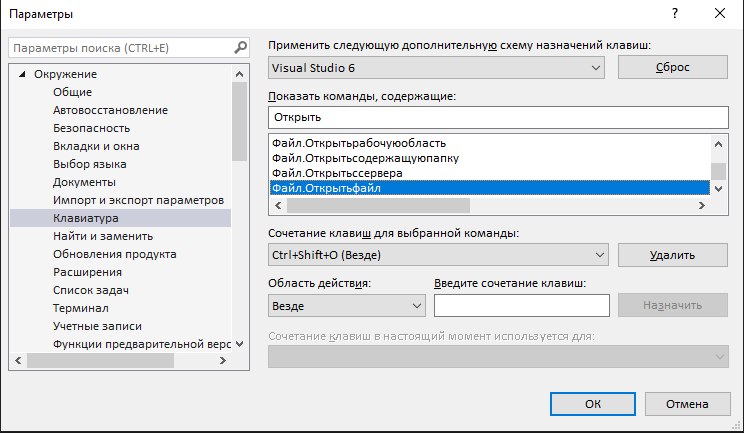


Рисунок 15 – Назначение горячей клавиши

## **1.4. Настройка макетов других окон**

Для добавления окон макета «Вид → Другие окна» (Рисунок 16).

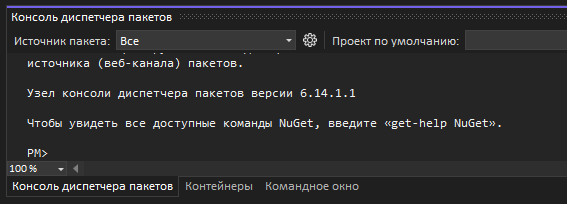


Рисунок 16 – Добавленные окна

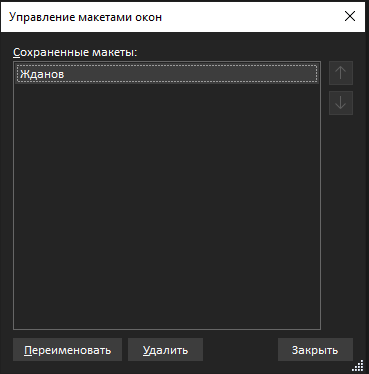
Для сохранения макета «Окно → Сохранить макет окна» (Рисунок 17). 

Рисунок 17 – Сохраненный макет

## **1.5. Экспорт настроек**

Для сохранения настроек «Средства → Импорт и экспорт параметров → Экспорт» (Рисунок 18).

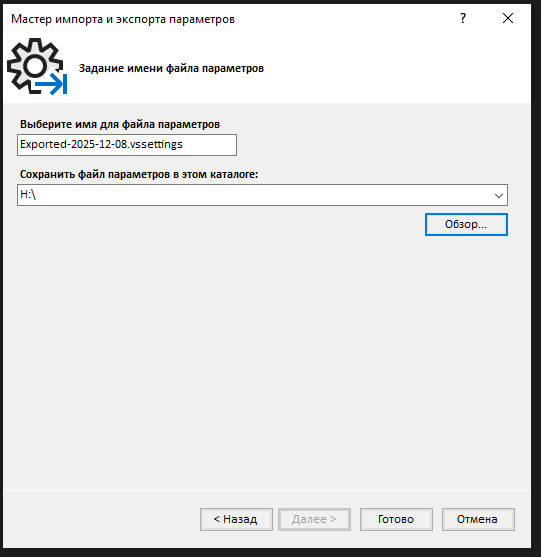


Рисунок 18 – Сохранение макета

Для импорта сохранённого макета «Средства → Импорт и экспорт параметров → Импорт» (Рисунок 19).

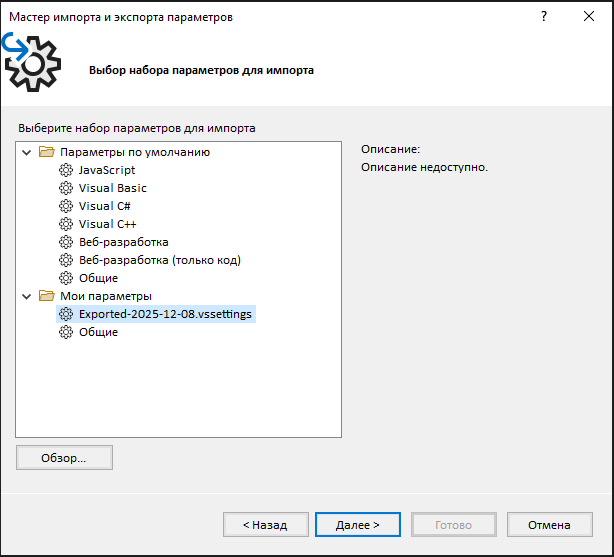


Рисунок 19 – Импорт макета

**Задача.** 1 вариант (нечетный номер в журнале) - Вам даны два целых числа. Напишите программу, выводящую произведение целых чисел.

Код задачи

#include <iostream>

using namespace std;

int main() {

int a{}, b{};

cin >> a; cin >> b;

cout << a \* b;

}

Запуски (Рисунок 20).

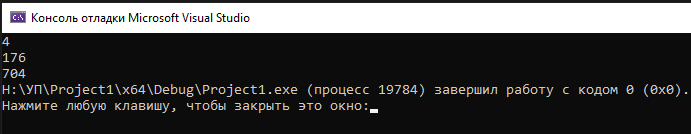


Рисунок 20 – a = 4, b = 176

# **Практическая работа №2 «Отладка в Visual Studio»**

**Цель:** освоить базовые и продвинутые техники отладки.

## **Навигация по коду с помощью отладчика**

В текстовом редакторе необходимо написать консольное приложение на C++ содержащее цикл и рекурсивный вызов функции. (Рисунок 21).



Рисунок 21 – Написание кода задачи

Для того чтобы поставить точку остановы нужно поставить красную точку слева от номера, там остановится код при выполнении в режиме debug. (Рисунок 22)

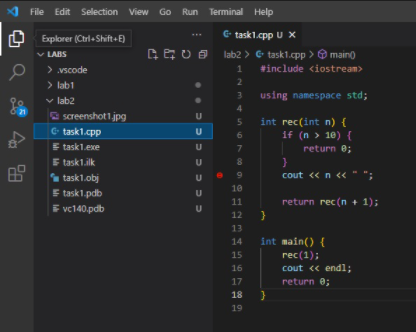


Рисунок 22 – Точка останова

Для того чтобы запустить отладку нужно нажать кнопку F5, используя кнопки F10, F11, Shift+F11 можно выполнить текущую строку, войти в метод и выйти из метода. (Рисунок 23).

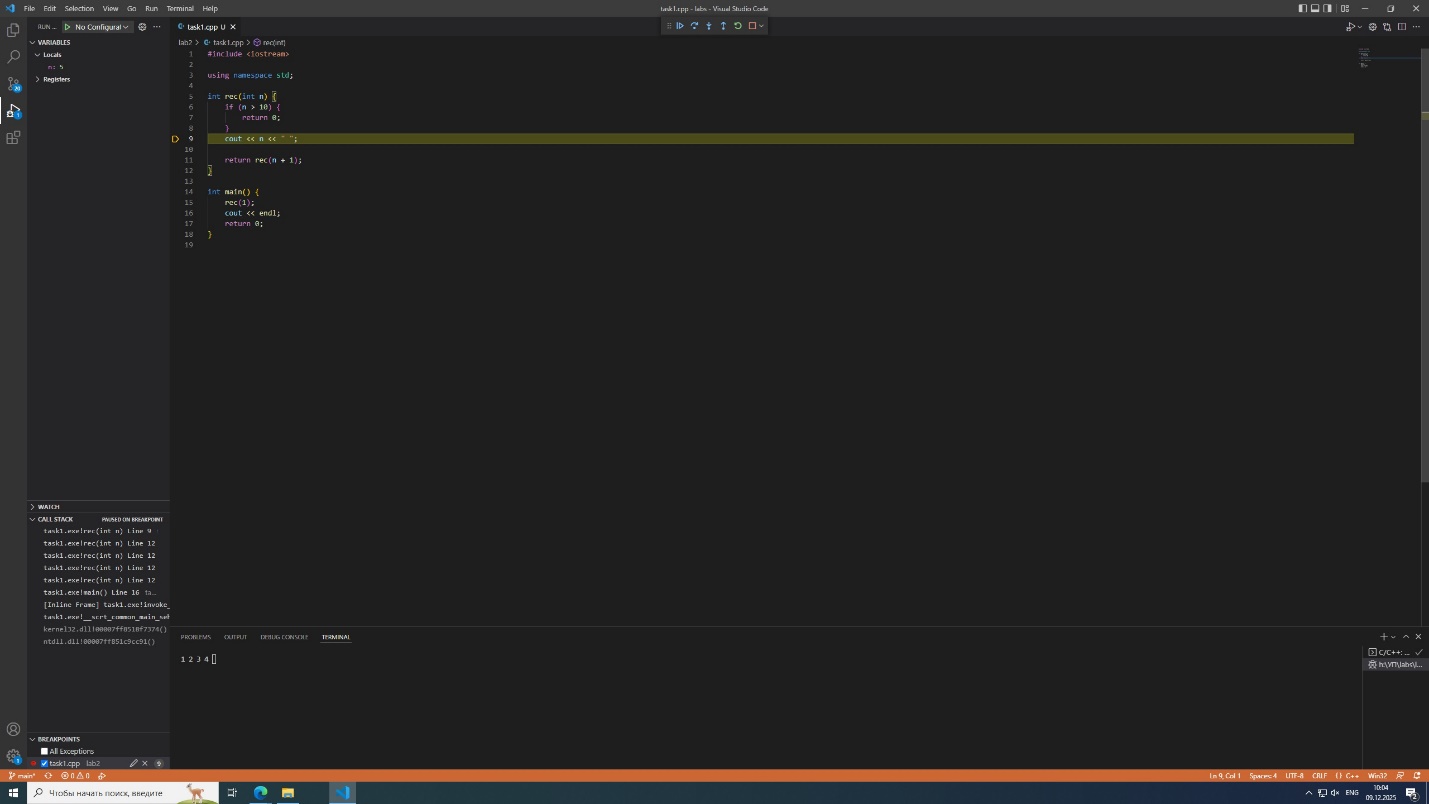


Рисунок 23 – Запуск отладки

Для установки условной точки остановы, нужно нажать правую кнопку мыши по красной точке и нажать условие, ввести, например n == 5 (Рисунок 24).

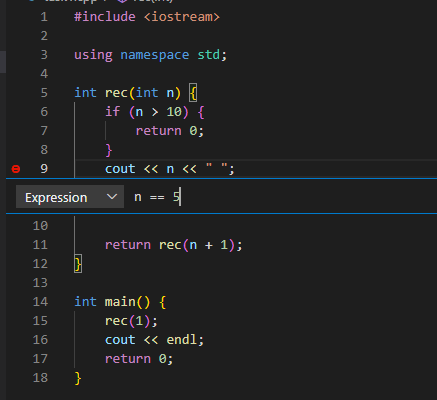


Рисунок 24 – Условная точка останова

Значения переменных можно посмотреть в окнах: variables, watch, quick watch (Рисунок 25).

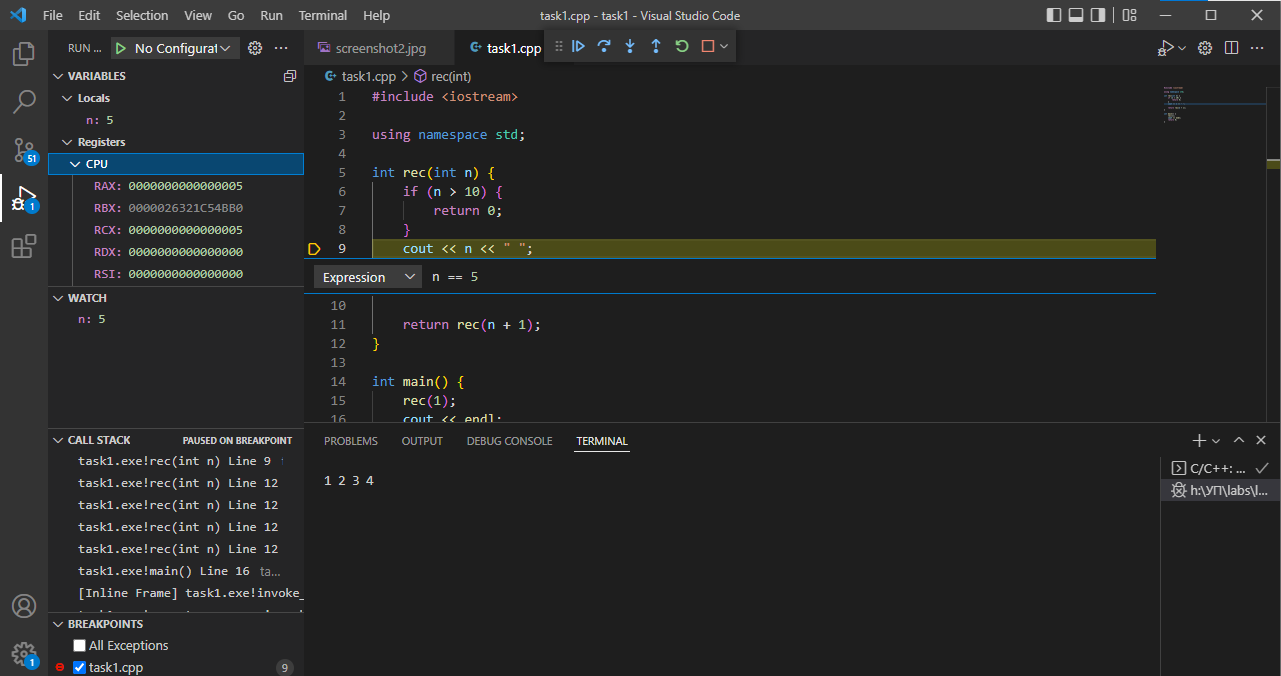


Рисунок 25 – Мониторинг переменных

## **Управление исключениями**

Для того, чтобы специально вызвать исключение, нужно добавить функцию, код которой будет вызывать исключение и вызвать ее в main, обернув в try-catch (Рисунок 26).

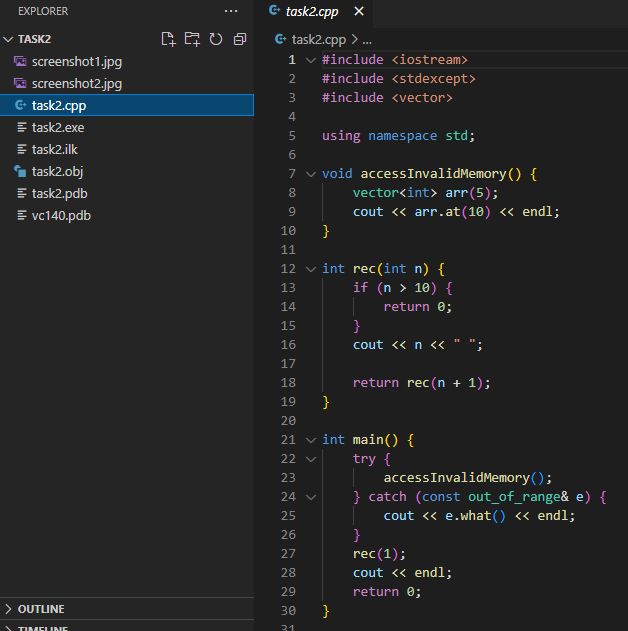


Рисунок 26 – Код с исключением

Для того, чтобы код прерывался при определенных исключениях, в окне “Breakpoints” напротив “All exceptions” вводим через запятую виды ошибок (Рисунок 27).

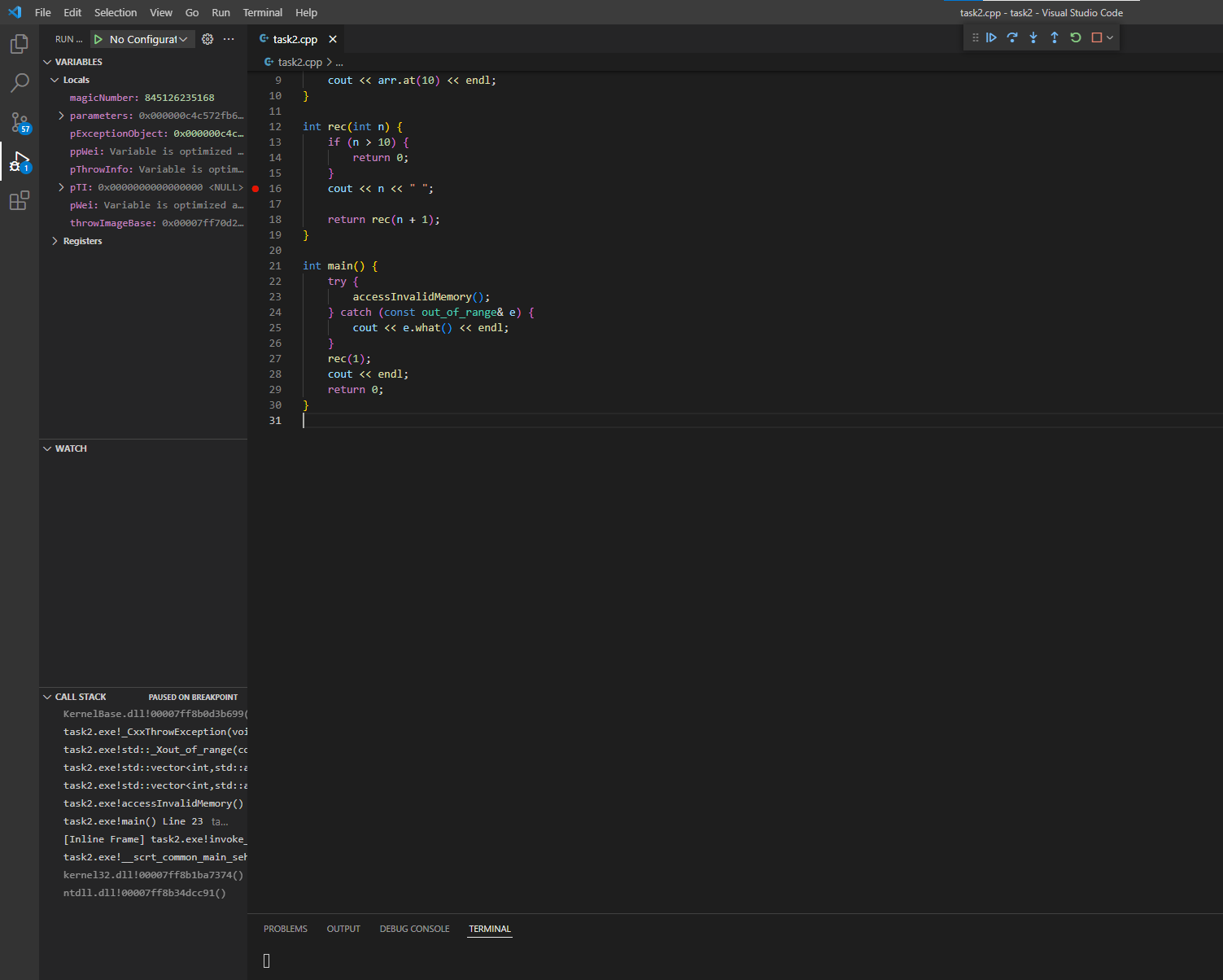


Рисунок 27 – Отладка при исключении

* 1. **Профилирование**

Для написания кода с интенсивными вычислениями, используй вычисление случайных чисел и добавление их значения к результату (Рисунок 28).

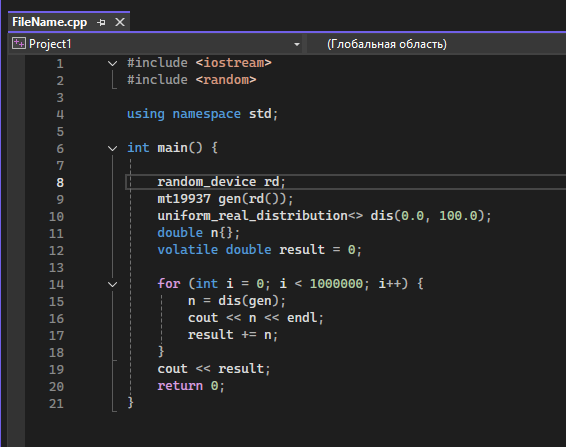


Рисунок 28 – Код с интенсивными вычислениями

Используя «**Отладка → Профилировщик производительности**», запускаем анализ «**Использование ЦП**» и «**Использование памяти**» и сохраняем отчет профилирования (Рисунок 29).

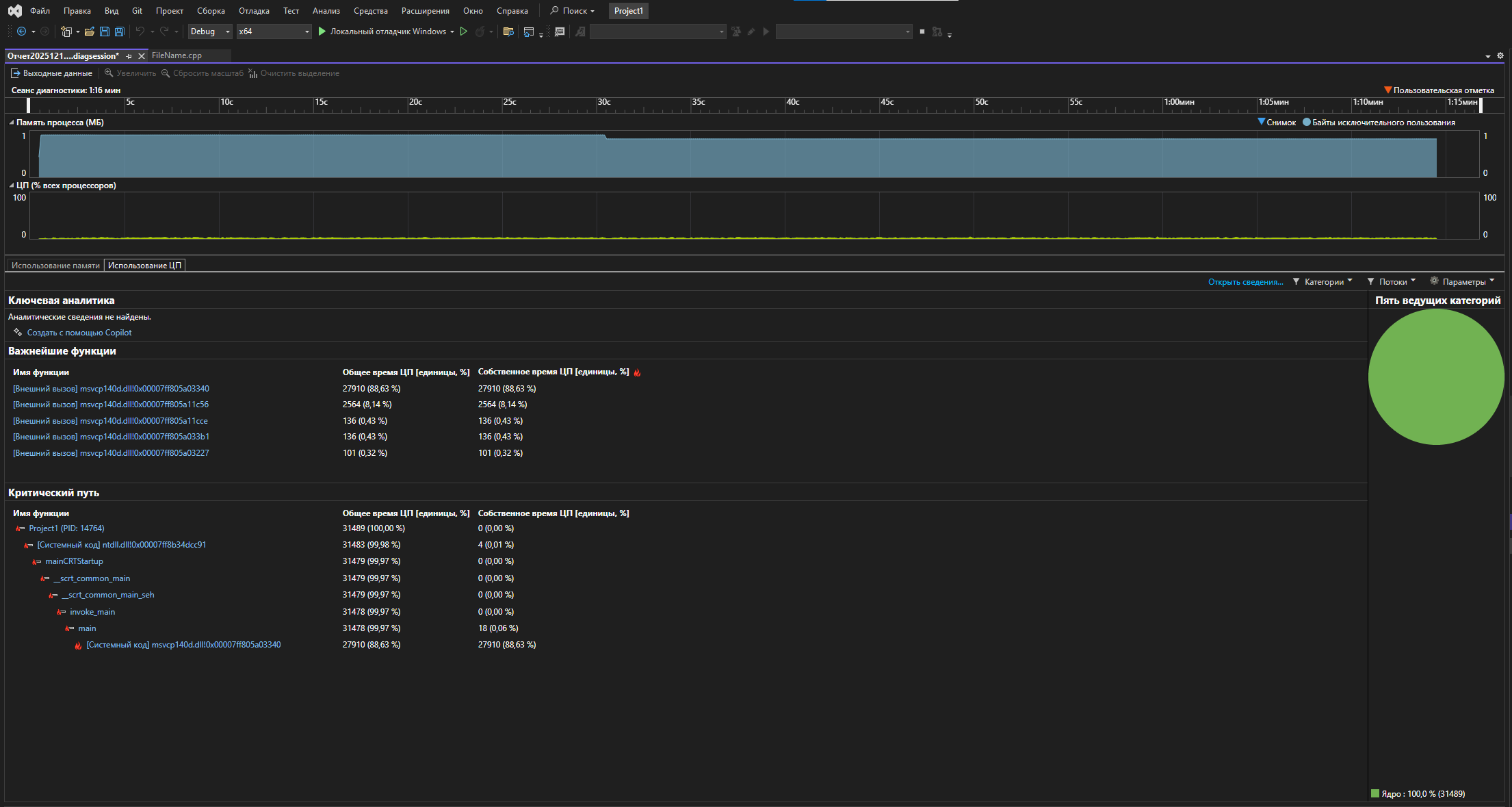


Рисунок 29 – Отчет профилирования

# **Практическая работа №3 «Обеспечение качества кода на C++»**

**Цель:** освоить базовые и продвинутые техники отладки.

## **Самодокументирующийся код**

Для того чтобы добавить XML-комментарии напишем код, состоящий из двух функций для вычисления времени пути исходя из расстояния и скорости, вариант 9 (Рисунок 30).

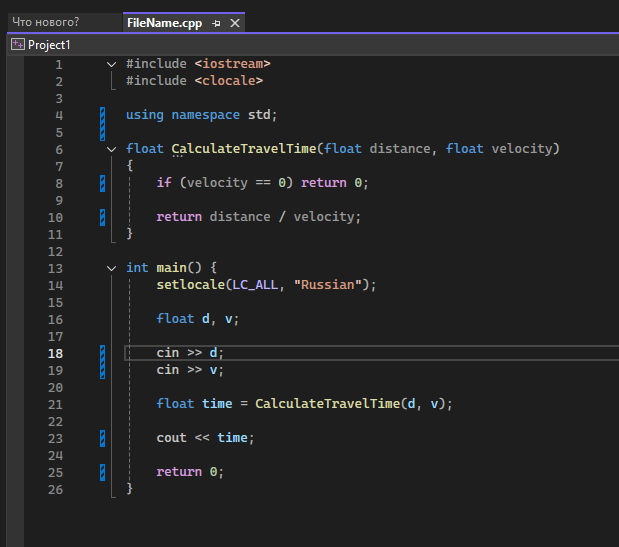


Рисунок 30 – Код для выполнения задачи

Для того чтобы в коде появлялись подсказки, нужно добавить XML-комментарии перед функцией (Рисунок 31).

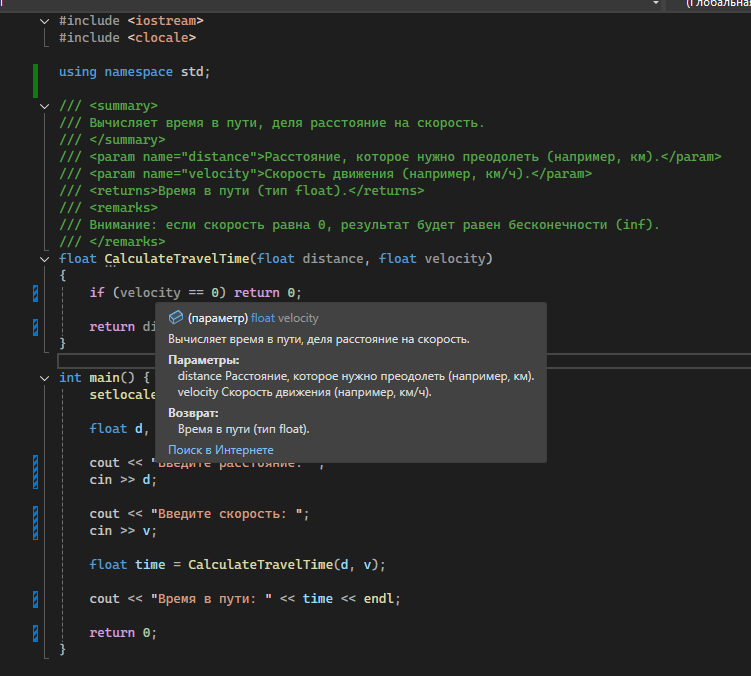


Рисунок 31 – Код с XML-комментариями и подсказками

## **Соглашение о кодировании**

Цель: Оформить свой код согласно правилам С++

Для того чтобы код соответствовал правилам C++, необходимо расставить правильные отступы, использовать правильные имена переменных и функций, иметь пробелы вокруг операторов, правильно расставленные фигурные скобки, убрать “using namespace std;” (Рисунок 32).

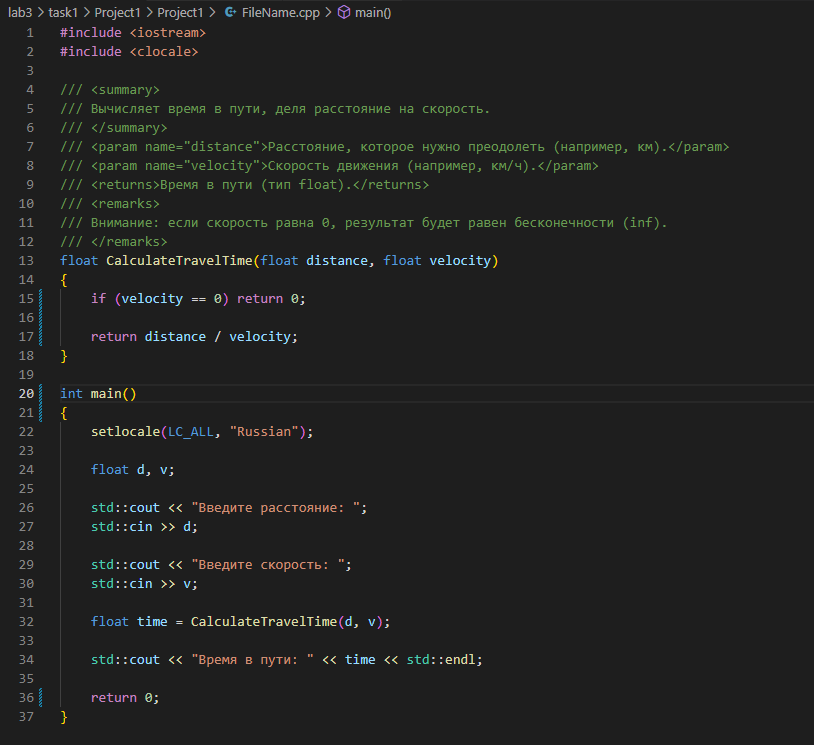


Рисунок 32 – Оформленный код

## **Метрика Джилба (упрощенный вариант)**

**Цель:** Для вашей функции из Задания 1, рассчитать сложность по метрике Джилба:

Для вычисления метрики Джилба необходимо посчитать количество исполняемых строк кода и прибавить количество логических переходов.  
Исходя из моего кода, можно посчитать:

Gillb = η1 + η2 = 2 + 1= **3** (Рисунок 33).

Следовательно код относится к первому уровню сложности (Gillb ≤ 10).

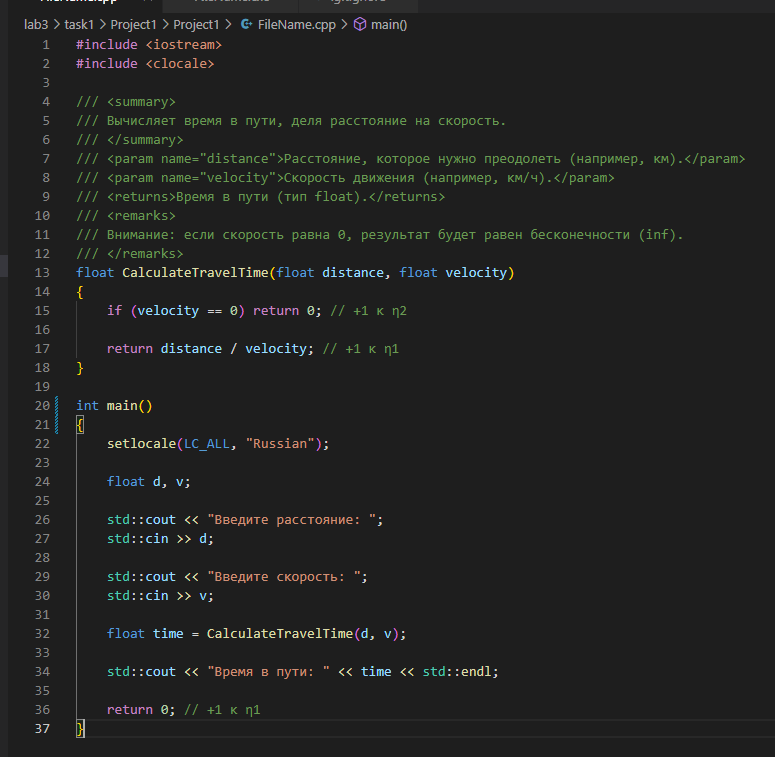


Рисунок 33 – Код с вычислением метрики Джилба

# **Практическая работа №4 «Работа с реестром ОС Windows»**

**Цель:** понять, что такое реестр, для чего нужен. Научиться работать с файлами.

Для того чтобы понять, что такое реестр и научиться с ним работать, необходимо создать консольное приложение и реализовать функционал для сохранения и восстановления из файла, обеспечить корректный вывод сохраненных данных при его перезапуске (Рисунок 34).

Код задачи, вариант 9:

#include <iostream>

#include <fstream>

#include <string>

#include <Windows.h>

using namespace std;

// функция для установки цвета по названию

void setConsoleColor(const string& color) {

    if (color == "red") {

        system("color 4F"); // красный фон + белый текст

    }

    else if (color == "blue") {

        system("color 1F"); // синий фон + белый текст

    }

    else if (color == "green") {

        system("color 2F"); // зелёный фон + белый текст

    }

    else if (color == "yellow") {

        system("color 6F"); // жёлтый фон + белый текст

    }

    else if (color == "white") {

        system("color 7F"); // белый фон + чёрный текст

    }

    else {

        system("color 0F"); // по умолчанию: чёрный фон + белый текст

    }

}

int main() {

    SetConsoleCP(1251);

    SetConsoleOutputCP(1251);

    string pet, color;

    ifstream fin("settings.txt"); //Создает объект fin для чтения из файла и открывает файл

    if (fin.is\_open()) {// Проверка открытия файла

        // читаем сохраненные данные

        fin >> pet >> color;// Читает данные из файла (аналогично cin)

        fin.close(); //Закрывает файл и освобождает ресурсы

        setConsoleColor(color);

        cout << "Любимое животное: " << pet << endl;

        char choice;

        cout << "\nХотите изменить поменять выбор? (y/n): ";

        cin >> choice;

        if (choice == 'y' || choice == 'Y') {

            cout << "Введите новое любимое животное (dog, cat, parrot): ";

            cin >> pet;

            ofstream fout("settings.txt"); //Создает объект fout для записи в файл и открывает файл

            //ofstream fout("settings.txt", ios::app); // Открываем для добавления, если нужно внести еще какие-то параметры в файл ios::app, если не указывать по умолчанию очищает

            fout << pet << " " << color; //Записывает данные в файл (аналогично cout)

            fout.close();

            cout << "Выбор изменен" << endl;

        }

        cout << "\nХотите изменить поменять цвет консоли? (y/n): ";

        cin >> choice;

        if (choice == 'y' || choice == 'Y') {

            cout << "Введите новый любимый цвет (red, blue, green, yellow, white): ";

            cin >> color;

            ofstream fout("settings.txt"); //Создает объект fout для записи в файл и открывает файл

            //ofstream fout("settings.txt", ios::app); // Открываем для добавления, если нужно внести еще какие-то параметры в файл ios::app, если не указывать по умолчанию очищает

            fout << pet << " " << color; //Записывает данные в файл (аналогично cout)

            fout.close();

            cout << "Выбор изменен" << endl;

        }

    }

    else {

        // файла нет – вводим новые настройки

        cout << "Введите ваше любимое животное (dog, cat, parrot): ";

        cin >> pet;

        cout << "Введите ваш любимый цвет (red, blue, green, yellow, white): ";

        cin >> color;

        ofstream fout("settings.txt");

        fout << pet << " " << color;

        fout.close();

        cout << "Настройки сохранены!" << endl;

    }

    return 0;

}

ЗАКЛЮЧЕНИЕ

Заключение пишем ко всему отчету сразу, не к отдельным работам

**АТТЕСТАЦИОННЫЙ ЛИСТ**

**ПО УЧЕБНОЙ ПРАКТИКЕ (ПО ПРОФИЛЮ СПЕЦИАЛЬНОСТИ)**

по профессиональному модулю ПМ.04 «Сопровождение и обслуживание программного обеспечения компьютерных систем»

(код и наименование)

Специальность 09.02.07 «Информационные системы и программирование»

(код и наименование специальности)

Студент(ка) II курса 24290907/3091 группы

Жданов Иван Андреевич

(ФИО полностью)

Место прохождения практики: ФГАОУ ВО СПбПУ Петра Великого  
Институт СПО, учебно-вычислительный центр, пр. Энгельса д.23

(наименование и адрес организации)

Период прохождения практики

с «8» декабря 2025 г. по «27» декабря 2025 г.

**Виды и качество выполнения работ**

|  |  |  |
| --- | --- | --- |
| **Виды выполненных работ обучающимся**  **во время практики** | **Объем работ, час.** | **Качество выполнения работ (оценка по пятибалльной системе)** |
| **Тема 1** Персонализация интегрированной среды разработки Visual Studio Community 2022 | 18 |  |
| **Тема 2** Отладка в IDE Visual Studio Community 2022 | 30 |  |
| **Тема 3** Обеспечение качества кода | 36 |  |
| **Тема 4** Упаковка приложения | 24 |  |

Характеристика учебной/профессиональной деятельности обучающегося во время учебной практики (по профилю специальности):

Общие и профессиональные компетенции, предусмотренные программой практики, *освоены* ***/*** *не освоены*.

(нужное подчеркнуть)

Итоговая оценка по практике \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель практики     Курылева А. А. \_\_\_\_\_\_\_\_\_\_\_\_\_

(Ф.И.О.) (подпись)