**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное автономное образовательное

учреждение высшего образования

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Инженерная школа информационных технологий и робототехники

Отделение информационных технологий

Направление информатика и вычислительная техника

Отчет

по лабораторной работе №1

по дисциплине

**«**ПРОграммирование**»**

**Работа со средством автоматизации сборки ПО CMake и компилятором GCC в редакторе кода Visual Studio Code. Отладка программного кода на языке С++**

Выполнил:

Студент группы 8В31 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.А.Моцыборова

Проверил:

Ассистент ОИТ ИШИТР \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.В. Кузнецов

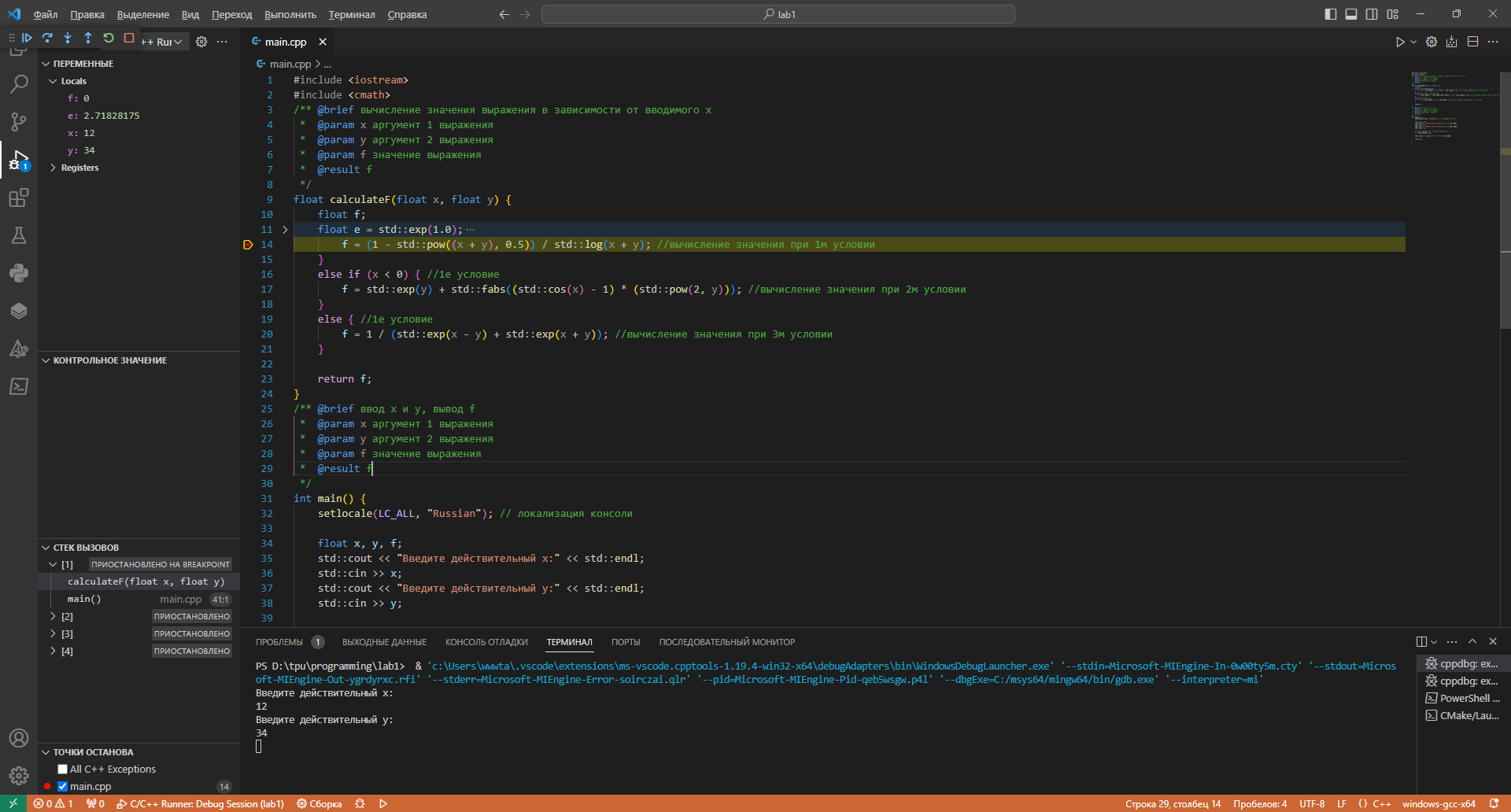
# Томск 2024

# Цель работы

# Получить навыки сборки проекта на языке программирования С++ с применением программного средства автоматизации сборки программного обеспечения CMake и компилятора GCC в редакторе кода Visual Studio Code. Получить навыки отладки программы, написанного на языке программирования С++.

# Задание

# На основе программного кода, написанного в рамках выполнения задания №1 лабораторной работы № 7 по дисциплине ”Информатика 1.2” выполнить сборку проекта с применением CMake и GCC в редакторе кода VS Code. Продемонстрировать (не менее двух скриншотов для каждого из пунктов (1.1, 1.2, 1.3) задания к лабораторной работе №7 по дисциплине ”Информатика 1.2”) построчную отладку кода.



#include <iostream>

#include <cmath>

/\*\* @brief вычисление значения выражения в зависимости от вводимого x

 \*  @param x аргумент 1 выражения

 \*  @param y аргумент 2 выражения

 \*  @param f значение выражения

 \*  @result f

 \*/

float calculateF(float x, float y) {

    float f;

    float e = std::exp(1.0);

    if ((x >= 0) && (y > -x)) { //1е условие

        f = (1 - std::pow((x + y), 0.5)) / std::log(x + y); //вычисление значения при 1м условии

    }

    else if (x < 0) { //1е условие

        f = std::exp(y) + std::fabs((std::cos(x) - 1) \* (std::pow(2, y))); //вычисление значения при 2м условии

    }

    else { //1е условие

        f = 1 / (std::exp(x - y) + std::exp(x + y)); //вычисление значения при 3м условии

    }

    return f;

}

/\*\* @brief ввод x и y, вывод f

 \*  @param x аргумент 1 выражения

 \*  @param y аргумент 2 выражения

 \*  @param f значение выражения

 \*  @result f

 \*/

int main() {

    setlocale(LC\_ALL, "Russian"); // локализация консоли

    float x, y, f;

    std::cout << "Введите действительный x:" << std::endl;

    std::cin >> x;

    std::cout << "Введите действительный y:" << std::endl;

    std::cin >> y;

    // Вызов функции для вычисления значения f

    f = calculateF(x, y);

    std::cout << "Результат = " << f << std::endl;

    return 0;

}

Лаб 5 задание 1

#include <iostream>

// Функция increaseMinBy2 принимает указатели на переменные a и b и увеличивает значение минимальной переменной в 2 раза

void increaseMinBy2(int\* aPtr, int\* bPtr) {

    if (\*aPtr < \*bPtr) {

        \*aPtr \*= 2;

    } else {

        \*bPtr \*= 2;

    }

}

// Функция swapValues принимает указатели на переменные a и b и меняет их значения местами

void swapValues(int\* aPtr, int\* bPtr) {

    int temp = \*aPtr;

    \*aPtr = \*bPtr;

    \*bPtr = temp;

}

int main() {

    int a, b;

    std::cout << "Введите значение a: ";

    std::cin >> a;

    std::cout << "Введите значение b: ";

    std::cin >> b;

    int\* aPtr = &a; // Создание указателя aPtr и привязка его к переменной a

    int\* bPtr = &b; // Создание указателя bPtr и привязка его к переменной b

    increaseMinBy2(aPtr, bPtr); // Вызов функции increaseMinBy2 с передачей указателей на a и b

    swapValues(aPtr, bPtr); // Вызов функции swapValues с передачей указателей на a и b

    std::cout << "a = " << a << std::endl; // Вывод значения переменной a

    std::cout << "b = " << b << std::endl; // Вывод значения переменной b

    return 0;

}

Лаб 6 задание 1 вариант 1

#include <iostream>

#include <vector>

#include <random>

// Генерирует N случайных целых чисел от 0 до 20 и сохраняет их в массиве Z

void generateNumbers(std::vector<int>& Z, int N) {

std::random\_device rd;

std::mt19937 gen(rd());

std::uniform\_int\_distribution<int> dist(0, 20);

for (int i = 0; i < N; i++) {

int num = dist(gen);

Z.push\_back(num);

}

}

// Проверяет существование элементов, равных x, в массиве Z

// Возвращает количество найденных элементов и массив с их позициями

int checkElements(const std::vector<int>& Z, int x, std::vector<int>& positions) {

int count = 0;

for (int i = 0; i < Z.size(); i++) {

if (Z[i] == x) {

count++;

positions.push\_back(i);

}

}

return count;

}

int main() {

int N;

std::cout << "Введите число N (N >= 3): ";

std::cin >> N;

std::vector<int> Z;

generateNumbers(Z, N);

std::cout << "Массив Z(N): ";

for (int i = 0; i < Z.size(); i++) {

std::cout << Z[i] << " ";

}

std::cout << std::endl;

int x;

std::cout << "Введите число x: ";

std::cin >> x;

std::vector<int> positions;

int count = checkElements(Z, x, positions);

if (count > 0) {

std::cout << "Количество элементов " << x << " в массиве Z(N): " << count << std::endl;

std::cout << "Позиции элементов " << x << " в массиве Z(N): ";

for (int i = 0; i < positions.size(); i++) {

std::cout << positions[i] << " ";

}

std::cout << std::endl;

} else {

std::cout << "Нет элементов, равных " << x << " в массиве Z(N)" << std::endl;

}

return 0;

}

# Вывод