



МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет Информационных технологий
Кафедра Информатики и информационных технологий

направление подготовки

09.03.02 «Информационные системы и технологии»

ЛАБОРАТОРНАЯ РАБОТА № 2

Дисциплина: Основы алгоритмизации и программирования

Тема: "Основные сведения об алгоритмах"

Выполнил: студент группы 211-721

Дерендяев Дмитрий Сергеевич
(Фамилия И.О.)

Дата, подпись 02.10.2021 _____
(Дата) (Подпись)

Проверил: Новичков Иван Константинович _____
(Фамилия И.О., степень, звание) (Оценка)

Дата, подпись _____
(Дата) (Подпись)

Замечания:

Москва

2021

Лабораторная работа №2-3

"Основные сведения об алгоритмах"

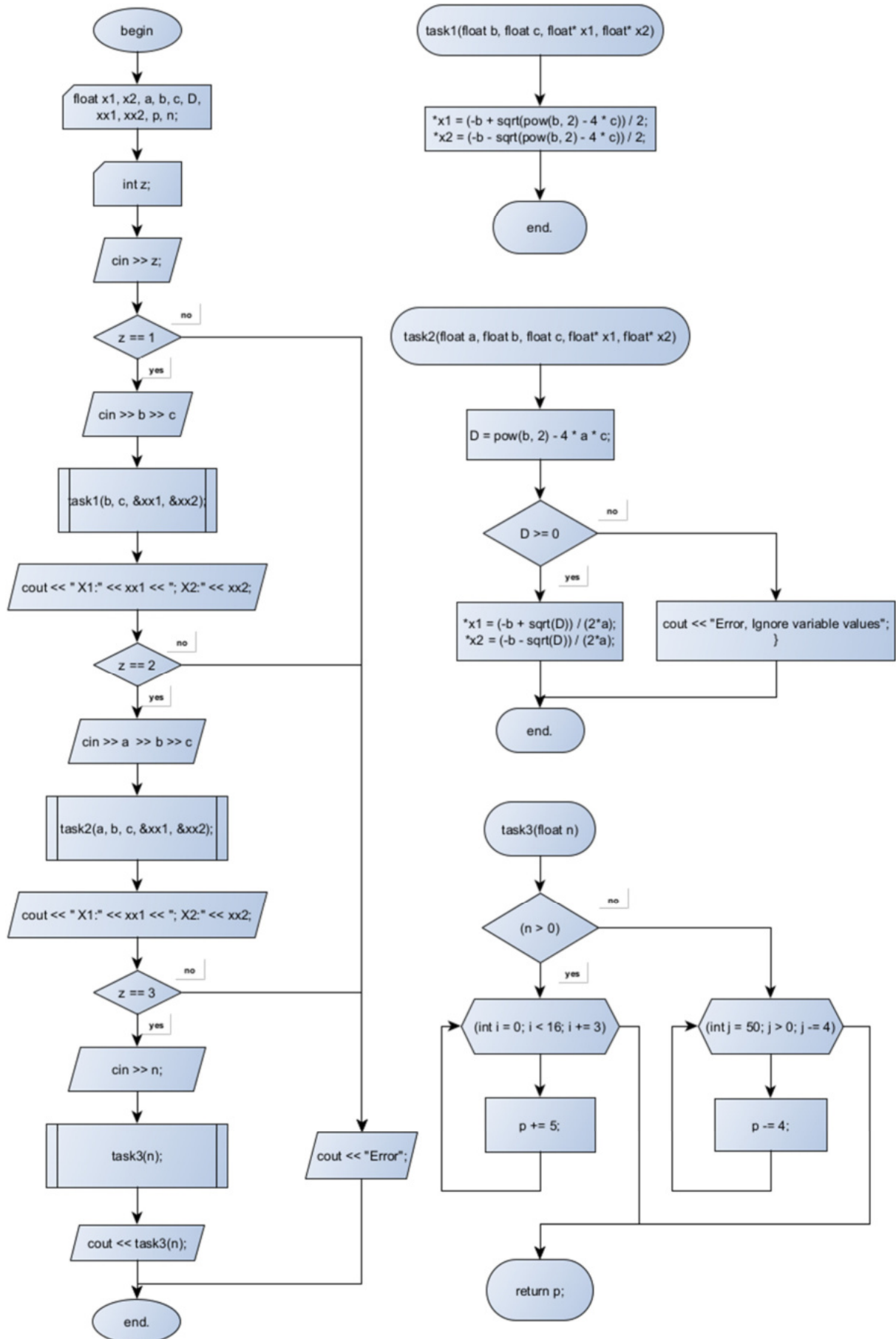
(продолжительность 4 часа)

Цель: Получить практические навыки разработке алгоритмов и их программной реализации.

Задачи:

Необходимо разработать алгоритмы и выполнить их программную реализацию для следующих задач:

1. Вычисление алгебраического выражения, не требующего сравнительных и циклических операций. Задается преподавателем
2. Вычисление алгебраического выражения, требующего сравнительных и не требующего циклических операций. Задается преподавателем
3. Вычисление алгебраического выражения, требующего сравнительных и циклических операций. Задается преподавателем



Листинг программы:

```
#include <iostream> //подключение ключевых библиотек: ввод/вывод, математических формул, ..
#include <cmath>
#include <stdio.h>

using namespace std;
float x1, x2, a, b, c, D; //объявление переменных
float xx1, xx2;
float p, n;

void task1(float b, float c, float* x1, float* x2) //инициализация функции task1
{
    *x1 = (-b + sqrt(pow(b, 2) - 4 * c)) / 2; //задача функции записать в адреса
    переменных X1 и X2 значения арифметических выражений
    *x2 = (-b - sqrt(pow(b, 2) - 4 * c)) / 2; //производится расчет корней квадратного
    уравнения (без проверки на отрицательны дискриминант)
}

void task2(float a, float b, float c, float* x1, float* x2) //инициализация функции task2
{
    D = pow(b, 2) - 4 * a * c; //расчет дискриминанта квадратного уравнения
    if (D >= 0) //проверка положительности дискриминанта
    {
        *x1 = (-b + sqrt(D)) / (2*a); //задача функции записать в адреса переменных
        X1 и X2 значения арифметических выражений
        *x2 = (-b - sqrt(D)) / (2*a); //
    }
    else
    {
        cout << "Error, Ignore variable values"; //в случае, если дискриминант равен
        нулю, вывести сообщение об ошибке
    }
}

int task3(float n) //инициализация функции task3
{
    if (n > 0) //проверка отрицательности переменной N, выполнение первого цикла, в
    случае положительности
    {
        for (int i = 0; i < 16; i += 3) //запуск цикла for с перебором значений i до
        16 и шагом 3
        {
            p += 5; //увеличение значения переменной на 5
        }
    }
    else //выполнение первого цикла, в случае отрицательности
    {
        for (int j = 50; j > 0; j -= 4) //запуск цикла for с перебором значений j
        до 0 и шагом -4
        {
            p -= 4; //уменьшение значения переменной на 4
        }
    }
    return p;
}

int main() //начало основаной программы
{
    int z; //объявление переменной z
    cin >> z; //ввод переменной z
    switch (z) //условный оператор множественного выбора
    {
        case 1:
```

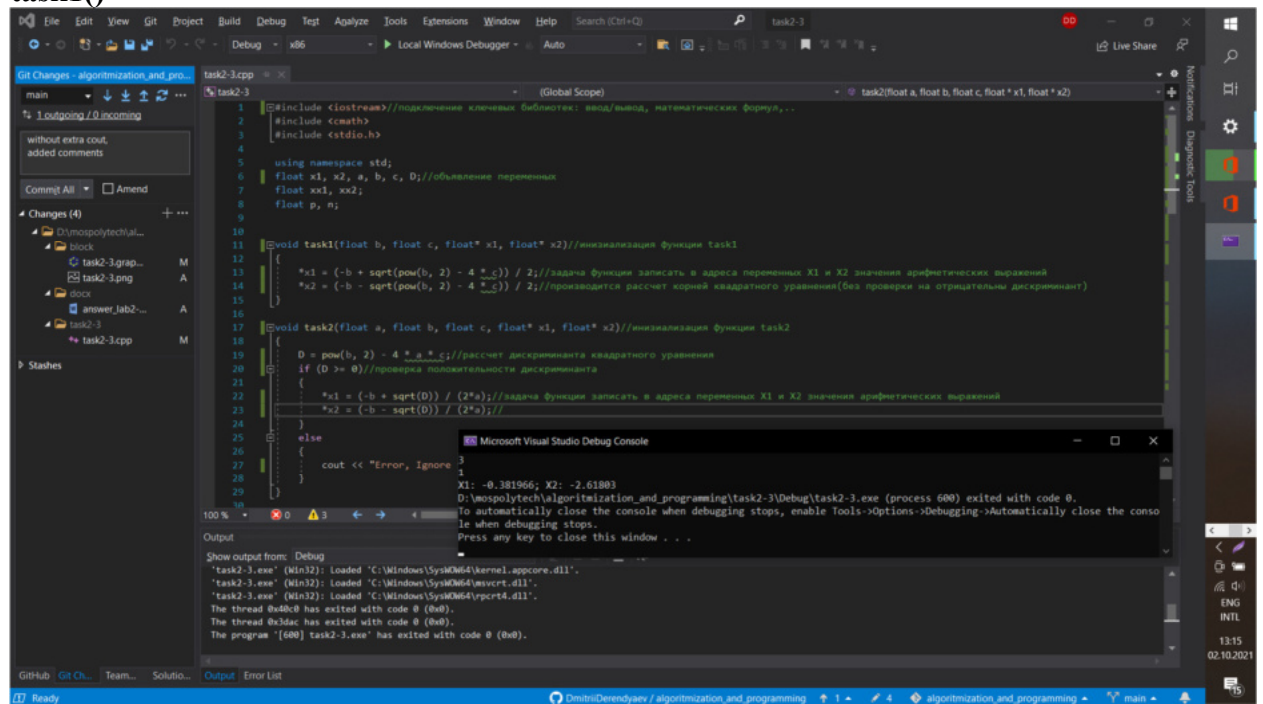
```

{
    cin >> b >> c; // ввод переменных b, c
    task1(b, c, &xx1, &xx2); // вызов процедуры task1 (внутри объявления процедуры
    // происходит запись значений адреса переменных x1 и x2 в переменные xx1 и xx2
    // соответственно)
    cout << "X1: " << xx1 << "; X2: " << xx2; // вывод значений переменных xx1,
    // xx2
    break; // игнорирование остальных условий внутри case
}
case 2:
{
    cin >> a >> b >> c; // ввод переменных a, b, c
    task2(a, b, c, &xx1, &xx2); // вызов процедуры task2 (внутри объявления
    // процедуры происходит запись значений адреса переменных x1 и x2 в переменные xx1 и xx2
    // соответственно)
    cout << " X1: " << xx1 << "; X2: " << xx2; // вывод значений переменных xx1,
    // xx2
    break; // игнорирование остальных условий внутри case
}
case 3:
{
    cin >> n; // ввод переменной n
    cout << task3(n); // вывод результата работы функции task3
    break; // игнорирование остальных условий внутри case
}
default:
    cout << "Error"; // при вводе в переменную (выбора) значения иного, чем 1-3,
    // вывод ошибки
}

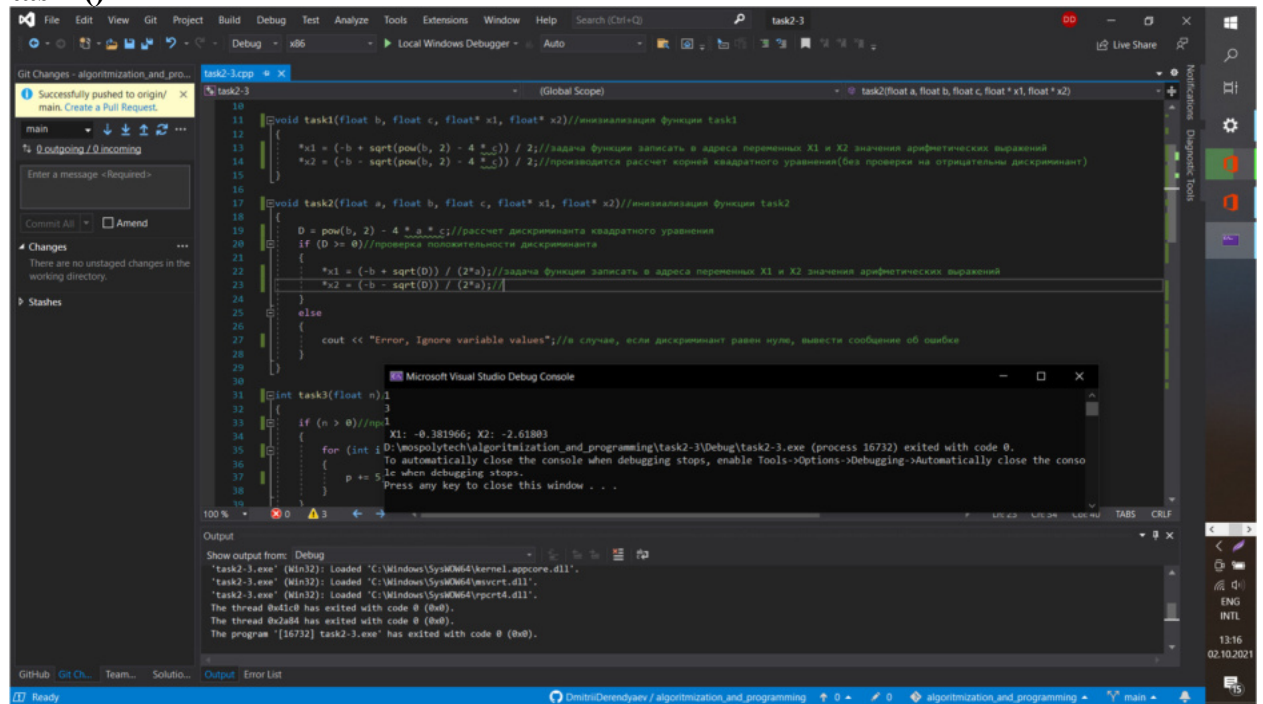
return 0; // завершение работы основной программы
}

```

Демонстрация работоспособности программы: task1()

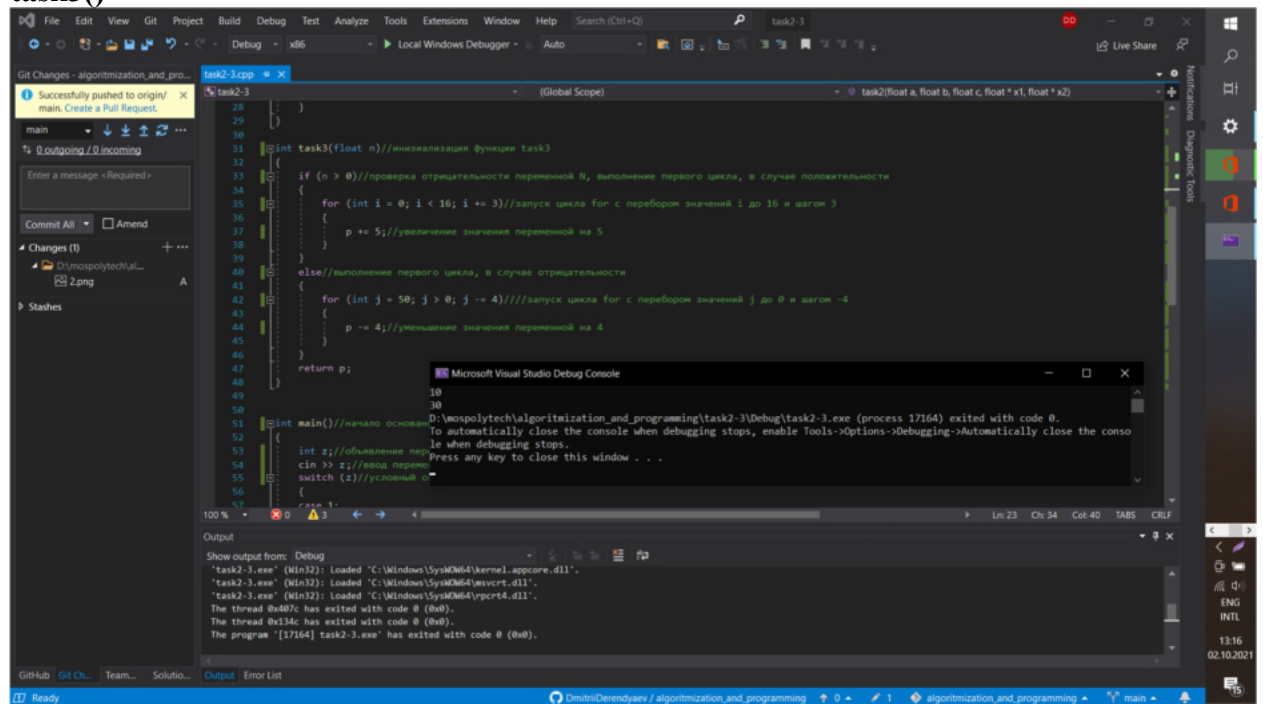


task2()



```
11 void task1(float b, float c, float* x1, float* x2) //инициализация функции task1
12 {
13     *x1 = (-b + sqrt(pow(b, 2) - 4 * c)) / 2; //задача функции записать в адреса переменных X1 и X2 значения арифметических выражений
14     *x2 = (-b - sqrt(pow(b, 2) - 4 * c)) / 2; //производится расчет корней квадратного уравнения (без проверки на отрицательный дискриминант)
15 }
16
17 void task2(float a, float b, float c, float* x1, float* x2) //инициализация функции task2
18 {
19     D = pow(b, 2) - 4 * a * c; //расчет дискриминанта квадратного уравнения
20     if (D >= 0) //проверка положительности дискриминанта
21     {
22         *x1 = (-b + sqrt(D)) / (2*a); //задача функции записать в адреса переменных X1 и X2 значения арифметических выражений
23         *x2 = (-b - sqrt(D)) / (2*a); //
24     }
25     else
26     {
27         cout << "Error, ignore variable values" //в случае, если дискриминант равен нулю, вывести сообщение об ошибке
28     }
29 }
30
31 int task3(float n)
32 {
33     if (n > 0) //n>0
34     {
35         for (int i = 0; i < 16; i += 3) //запуск цикла for с перебором значений i до 16 и шагом 3
36         {
37             p += 5; //увеличение значения переменной на 5
38         }
39     }
40     else //выполнение первого цикла, в случае отрицательности
41     {
42         for (int j = 50; j > 0; j -= 4) //запуск цикла for с перебором значений j до 0 и шагом -4
43         {
44             p -= 4; //уменьшение значения переменной на 4
45         }
46     }
47     return p;
48 }
49
50 int main() //начало функции
51 {
52     int z; //объявление переменной z
53     cin >> z; //ввод переменной z
54     switch (z) //условный оператор
55     {
56         case 1:
57             task1(1, 2, &z, &z);
58             task2(1, 2, 3, &z, &z);
59             task3(1);
60     }
```

task3()



```
28 }
29
30
31 int task3(float n) //инициализация функции task3
32 {
33     if (n > 0) //проверка отрицательности переменной N, выполнение первого цикла, в случае положительности
34     {
35         for (int i = 0; i < 16; i += 3) //запуск цикла for с перебором значений i до 16 и шагом 3
36         {
37             p += 5; //увеличение значения переменной на 5
38         }
39     }
40     else //выполнение первого цикла, в случае отрицательности
41     {
42         for (int j = 50; j > 0; j -= 4) //запуск цикла for с перебором значений j до 0 и шагом -4
43         {
44             p -= 4; //уменьшение значения переменной на 4
45         }
46     }
47     return p;
48 }
49
50 int main() //начало функции
51 {
52     int z; //объявление переменной z
53     cin >> z; //ввод переменной z
54     switch (z) //условный оператор
55     {
56         case 1:
57             task1(1, 2, &z, &z);
58             task2(1, 2, 3, &z, &z);
59             task3(1);
60     }
```

При необходимости, вы можете найти всю историю разработки программы на моем GitHub:

https://github.com/DmitriiDerendyaev/algoritmization_and_programming