

Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет информатики и
радиоэлектроники»
Филиал
«Минский радиотехнический колледж»

Учебная дисциплина
«Основы алгоритмизации и программирования»

Отчет
по выполнению лабораторной работы
«Разработка, отладка и испытание ветвящихся алгоритмов и программ»

Выполнил: Гуринович Роман Дмитриевич, гр. 1K9393
Проверил: Козел А. С.

Минск, 2022

Лабораторная работа № 2

Тема работы: «Разработка, отладка и испытание ветвящихся алгоритмов и программ»

Вариант 6

```
ConsoleApplication1.cpp  X
ConsoleApplication1      (Глобальная область)
1  #include<iostream>
2  #include<math.h>
3  using namespace std;
4  int main() {
5      setlocale(LC_ALL, "Rus");
6      double a, b, c, d;
7      cout << "Почередно введите значения коэффициентов уравнения: " << endl;
8      cin >> a >> b >> c;
9      d = pow(b, 2) - 4 * a * c;
10
11     if (sqrt(d) > 0) { cout << "Уравнение имеет два корня: x1 = " << (double)(-1 * b - sqrt(d)) / (2 * a) << ", x2 = " << (double)(-1 * b + sqrt(d)) / (2 * a) << endl; }
12     else if (sqrt(d) == 0.0) { cout << "Уравнение имеет один корень: " << (double)(-b) / (2 * a) << endl; }
13     else { cout << "Уравнение не имеет корней"; }
14
15
16     return 0;
17 }
18
```

```
Консоль отладки Microsoft Visual Studio
Почередно введите значения коэффициентов уравнения:
2
3
94
Уравнение не имеет корней
C:\Users\asus\source\repos\ConsoleApplication1\x64\Debug\ConsoleApplication1.exe (процесс 14920) завершил работу с кодом
0.
Нажмите любую клавишу, чтобы закрыть это окно:
```

```
Консоль отладки Microsoft Visual Studio
Почередно введите значения коэффициентов уравнения:
1
10
25
Уравнение имеет один корень: -5
C:\Users\asus\source\repos\ConsoleApplication1\x64\Debug\ConsoleApplication1.exe (процесс 11076) завершил работу с кодом
0.
Нажмите любую клавишу, чтобы закрыть это окно: _
```

```
Консоль отладки Microsoft Visual Studio
Почередно введите значения коэффициентов уравнения:
6
0
-54
Уравнение имеет два корня: x1 = -3, x2 = 3
C:\Users\asus\source\repos\ConsoleApplication1\x64\Debug\ConsoleApplication1.exe (процесс 4164) завершил работу с кодом
0.
Нажмите любую клавишу, чтобы закрыть это окно: _
```

1. Цель работы

Формирование первичных умений в разработке разветвляющихся программ.

2. Задание

6. Напишите программу, печатающую корни квадратного уравнения $ax^2 + bx + c = 0$. Если уравнение не имеет корней, то выведите соответствующее сообщение.

3. Ответы на контрольные вопросы

1. Основные структуры структурного программирования, их схемы:

Следование, ветвление и цикл называют базовыми конструкциями структурного программирования.

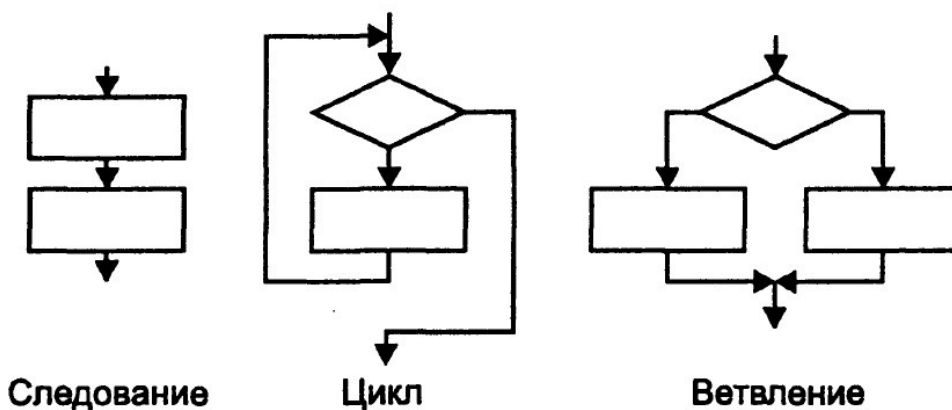


Рисунок 1- Базовые конструкции структурного программирования

2. Дополнительные структуры структурного программирования, их схемы.

цикл с
предусловием цикл с
постусловием

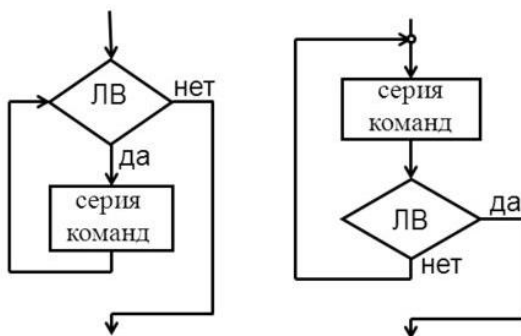


Рисунок 2 - Циклы

3. Логические операции, их приоритет, таблицы истинности.

1. В C++ существует три логические операции:
2. Логическая операция И &&;

3. Логическая операция ИЛИ \parallel ;
4. Логическая операция НЕ! или логическое отрицание.

Логические операции образуют сложное (составное) условие из нескольких простых (два или более) условий. Эти операции упрощают структуру программного кода в несколько раз.

Отрицание **Дизъюнкция (совокупность)** **Конъюнкция (система)**

A	\bar{A}
1	0
0	1

A	B	$A \cup B$	$A+B$	$\begin{cases} A \\ B \end{cases}$ или
1	1	1	1	1
1	0	1	1	1
0	1	1	1	1
0	0	0	0	0

A	B	$A \cap B$	$A \cdot B$	$\begin{cases} A \\ B \end{cases}$ и
1	1	1	1	1
1	0	0	0	0
0	1	0	0	0
0	0	0	0	0

Импликация (следование)

A	B	$A \Rightarrow B$	Если A, то B
1	1	1	1
1	0	0	0
0	1	1	1
0	0	1	1

Эквивалентность (равносильность)

A	B	$A \Leftrightarrow B$	A тогда и только тогда, когда B
1	1	1	1
1	0	0	0
0	1	0	0
0	0	1	1

Рисунок 3 - Таблицы истинности логических операций

4. Законы алгебры логики:

Выражение также называют логическим умножением. Его истинность сохраняется только в одном случае: при истинности простых выражений в его составе. В обратном случае выражение принимает значение ложного.

A	B	A & B
0	0	0
0	1	0
1	0	0
1	1	1

Рисунок 5 – Конъюнкция

Данное выражение также называют логическим сложением. Оно истинно почти всегда, за исключением ситуации, когда все подвыражения ложны.

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

Рисунок 6 – Дизъюнкция

Инверсия (также называемая логическим отрицанием) заключается в добавлении частицы «не» к выражению. В случае, когда изначальное выражение истинно, его отрицание – ложь. Если изначальное выражение ложь, его отрицание является истиной.

A	неA
1	0
0	1

Рисунок 7 – Инверсия

5. Формат условного оператора, правила выполнения:

Одна из основных конструкций алгоритма – разветвляющий процесс. Он реализован в языке C++ условным оператором **if (условие) оператор_1; else оператор_2;** Здесь *условие* — это логическое выражение, переменная или константа.

Работает условный оператор следующим образом. Сначала вычисляется значения выражения, записанного в виде *условия*. Если оно имеет значение истина (true), выполняется *оператор_1*. В противном случае (значение ложное (false)) *оператор_2*.

6. Формат оператора выбора, правила выполнения.

Оператор выбора switch позволяет выбрать один вариант хода решения задачи из нескольких в зависимости от значения выражения. Таким образом, обеспечивается многонаправленное разветвление в программе. Оператор switch работает следующим образом: сначала вычисляется значение выражения. Потом это значение последовательно сравнивается с константами из заданного списка. Как только найдено совпадение для

одного из условий сравнения, выполняется последовательность инструкций, которые связаны с этим сравнением.