

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)
Кафедра экономической математики, информатики и статистики (ЭМИС)

Повторение основных конструкций языка программирования C/C++
Отчет по лабораторной работе по дисциплине «Объектно-ориентированное
программирование»

Студент группы 549



Баулин С. К.

«__» _____ 2020 г.

Кандидат физико-
математических наук,
доцент кафедры ЭМИС

_____ Шельмина Е. А.

_____ оценка

«__» _____ 2020 г.

Лабораторная работа №1

Цель работы: повторение основных конструкций языка программирования C/C++.

Краткие теоретические сведения.

Выполнение всех программ, написанных на языке C/C++, начинается с функции. У функции есть имя (`main`), после которого в круглых скобках перечисляются аргументы или параметры функции. Вслед за заголовком функции `main` в фигурных скобках размещается тело функции, которое представляет последовательность определений, описаний и исполняемых операторов. Как правило, определения и описания размещаются до исполняемых операторов. Каждое определение, описание и оператор завершается «;».

В языке C/C++ прежде чем использовать переменную, ее необходимо объявить. Объявить переменную с именем `x` можно так: `int x;`

Переменной можно присвоить какое-либо значение с помощью операции присваивания. Присвоить – это значит установить текущее значение переменной. По-другому можно объяснить, что операция присваивания запоминает новое значение в ячейке памяти, которая обозначена переменной.

Когда требуется не просто выполнять операции последовательно (линейный процесс), а изменять ход вычислительного процесса в зависимости от некоторых условий, используется ветвление по условию. В C/C++ ветвление по условию выполняется с помощью оператора: `if ... else`.

Для тех случаев, когда требуется выбрать действия не из двух вариантов, а из нескольких, предназначен оператор переключатель `switch`. Этот оператор позволяет обработать сразу несколько сравнений по условию «выражение равно».

Чтобы не повторять множество одинаковых фрагментов кода программы, предусмотрены специальные операторы циклического повторения выделенного фрагмента программы.

Существует несколько различных операторов, предназначенных для организации циклов.

1. Операторы цикла с предусловием.
2. Оператор цикла с постусловием.
3. Оператор цикла со счётчиком.

Массивы – структурированный тип данных с элементами одного и того же типа, имеющий одно имя и определённое количество элементов. Количество элементов определяет размер массива. Порядковый номер элемента массива называется его индексом.

Чтобы использовать массив, надо его объявить-выделить место в памяти компьютера, объём которой зависит от количества элементов и типа массива. Тип массива – это тип входящих в него элементов.

Каждый элемент массива имеет свой порядковый номер. Чтобы обратиться к элементу массива, надо написать имя массива и затем в квадратных скобках номер элемента. Элементы массивов в языке C/C++ нумеруются с нуля. Поэтому индекс последнего элемента массива на 1 меньше числа элемента в данном массиве.

Формулировка заданий с полученными результатами.

Задание 1. Вычислить $f(x)=kx+b$, при $x=1,2,...,100$ с использованием оператора while.

Код и результат выполненного задания представлен на рисунке 1.

```

13 int main()
14 {
15     SetConsoleCP(1251); SetConsoleOutputCP(1251); setlocale(LC_ALL, "Russian");
16     int x = 1;
17     double f = 0, k = 0, b = 0;
18
19     cout << "Введите k: "; cin >> k;
20     cout << "Введите b: "; cin >> b;
21
22     while (x <= 100) {
23         f = k * x + b;
24         cout << "\nf(" << x << ") = " << f << endl;
25         ++x;
26     }
27
28     return 1;
29 }
30
31

```

Консоль отладки Microsoft Visual Studio

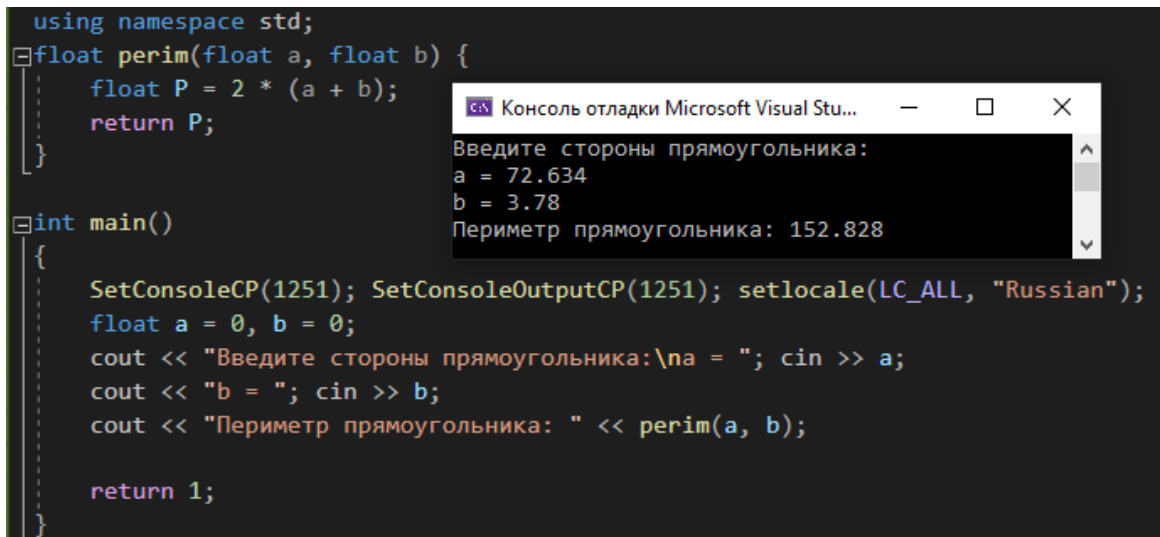
Введите k: 5.38
Введите b: 2.28

f(1) = 7.66
f(2) = 13.04
f(3) = 18.42
f(4) = 23.8
f(5) = 29.18
f(6) = 34.56
f(7) = 39.94
f(8) = 45.32
f(9) = 50.7

Рисунок 1 – Скриншот кода и результата выполненного задания 1

Задание 2. Написать функцию вычисления периметра прямоугольника.

Код и результат выполненного задания представлен на рисунке 2.



```
using namespace std;
float perim(float a, float b) {
    float P = 2 * (a + b);
    return P;
}

int main()
{
    SetConsoleCP(1251); SetConsoleOutputCP(1251); setlocale(LC_ALL, "Russian");
    float a = 0, b = 0;
    cout << "Введите стороны прямоугольника:\na = "; cin >> a;
    cout << "b = "; cin >> b;
    cout << "Периметр прямоугольника: " << perim(a, b);

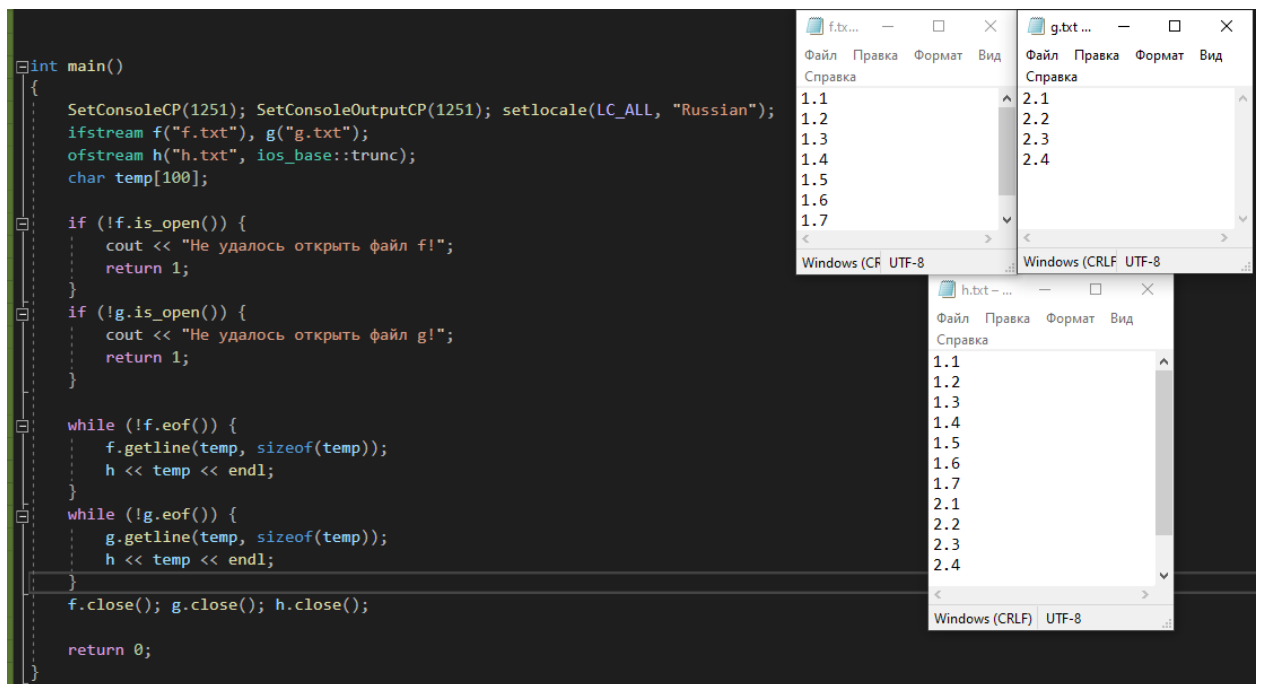
    return 1;
}
```

Консоль отладки Microsoft Visual Stu...
 Введите стороны прямоугольника:
 a = 72.634
 b = 3.78
 Периметр прямоугольника: 152.828

Рисунок 2 – Скриншот кода и результата выполненного задания 2

Задание 3. Даны текстовые файлы f и g. Записать в файл h сначала компоненты файла f, затем - компоненты файла g с сохранением порядка.

Код и результат выполненного задания представлен на рисунке 3.



```
int main()
{
    SetConsoleCP(1251); SetConsoleOutputCP(1251); setlocale(LC_ALL, "Russian");
    ifstream f("f.txt"), g("g.txt");
    ofstream h("h.txt", ios_base::trunc);
    char temp[100];

    if (!f.is_open()) {
        cout << "Не удалось открыть файл f!";
        return 1;
    }

    if (!g.is_open()) {
        cout << "Не удалось открыть файл g!";
        return 1;
    }

    while (!f.eof()) {
        f.getline(temp, sizeof(temp));
        h << temp << endl;
    }

    while (!g.eof()) {
        g.getline(temp, sizeof(temp));
        h << temp << endl;
    }

    f.close(); g.close(); h.close();

    return 0;
}
```

Windows (CF UTF-8) f.txt...
 1.1
 1.2
 1.3
 1.4
 1.5
 1.6
 1.7

Windows (CRLF UTF-8) g.txt...
 2.1
 2.2
 2.3
 2.4

Windows (CRLF UTF-8) h.txt...
 1.1
 1.2
 1.3
 1.4
 1.5
 1.6
 1.7
 2.1
 2.2
 2.3
 2.4

Рисунок 3 - Скриншот кода и результата выполненного задания 3

Вывод к лабораторной работе: повторены основные конструкции языка C/C++.