

## Практическое занятие 10

### Тема: «Условные операторы»

#### Теоретическая часть

На практике возможности программ, в которых операторы исполняются только в порядке их следования и исключительно по одному разу весьма ограничены. Слишком малое число проблем может быть решено подобным способом - большинство задач требует от программы принятия решения в зависимости от различных ситуаций. Для осуществления ветвления используются так называемые условные операторы.

#### Оператор `if`

Этот оператор обеспечивает ветвление программы в зависимости от результата проверки некоторого условия на истинность:

```
if (условие)  
    оператор1;  
    оператор2;
```

Проверяемое *условие* может быть любым выражением, которое чаще всего содержит операторы сравнения. Если условие принимает истинное значение (`true`), то выполняется оператор1. Если условие принимает ложное значение (`false`), то оператор1 не выполняется и выполнение программы переходит к оператору2.

В конструкциях языка C++ операторы могут быть блочными. Это означает, что в зависимости от принятого решения выполняется не один, а целый блок операторов. Блок начинается с открывающейся фигурной скобки (`{`) и заканчивается закрывающейся фигурной скобкой (`}`). Все содержимое блока рассматривается компилятором как единый оператор. В следующем примере показано использование блоков в операторе `if`.

```
#include <iostream>  
int main()  
{  
    int b;  
    if(b > 0)  
    { // Если условие b > 0 выполняется, то выполняется этот блок операторов  
    ... // Здесь могут размещаться другие операторы  
        cout << "b - положительное";  
    ... // Здесь могут размещаться другие операторы }  
    if(b < 0)  
        cout << "b — отрицательное";
```

```
return 0;  
}
```

В приведенном фрагменте объявляется целочисленная переменная  $b$ , а затем следует сравнение ее значения с нулем. Если величина  $b$  больше нуля, выводится сообщение " $b$  - положительное", а если  $b$  меньше нуля - сообщение " $b$  - отрицательное". В блоке могут быть размещены и другие операторы, которые необходимо выполнить при условии  $b > 0$ . Если необходимо добавить операторы при выполнении условия  $b < 0$ , то следует также создавать блок при помощи фигурных скобок `{ }`. После фигурных скобок блока точка с запятой не ставится.

## Операторы if-else

Оператор **if-else** применяется в тех случаях, когда по результату проверки некоторого условия принципиально возможно два пути развития алгоритма: первый — если условие выполняется, второй — если условие не выполняется. Оператор if с ключевым словом else имеет следующий вид:

```
if (условие)  
    оператор1;  
else  
    оператор2;  
оператор3;
```

Если проверяемое условие выполняется (принимает значение "True"), тогда осуществляется переход к оператору1 с последующим переходом к оператору3. В случае, когда проверяемое условие принимает ложное значение "False", тогда программа выполнит ветвь, которая содержит оператор2, а затем перейдет к оператору3.

## Практическая часть

### Оператор if

Приведенная ниже программа позволяет определить значения функции, которая представлена на рис.1. График этой функции представлен на рис.2.

```
#include <cstdlib>  
#include <iostream>  
  
using namespace std;  
  
int main(int argc, char *argv[])  
{
```

```

double x, y;
cout << endl << "Vvedite znachenie x = ";
cin >> x;
if(x < -2) y=0;
if((x >= -2)&&(x < -1)) y = -x - 2;
if((x >= -1)&&(x < 1)) y = x;
if((x >= 1)&&(x < 2)) y = -x + 2;
if(x >= 2) y = 0;
cout << endl << " x = " << x << "   y = " << y;
cout << endl << "Press the enter key to continue ...";
cin.get();
return EXIT_SUCCESS;
}

```

$$y = \begin{cases} 0, & x < -2 \\ -x - 2, & -2 \leq x < -1 \\ x, & -1 \leq x < 1 \\ -x + 2, & 1 \leq x < 2 \\ 0, & x \geq 2 \end{cases}$$

Рис.1

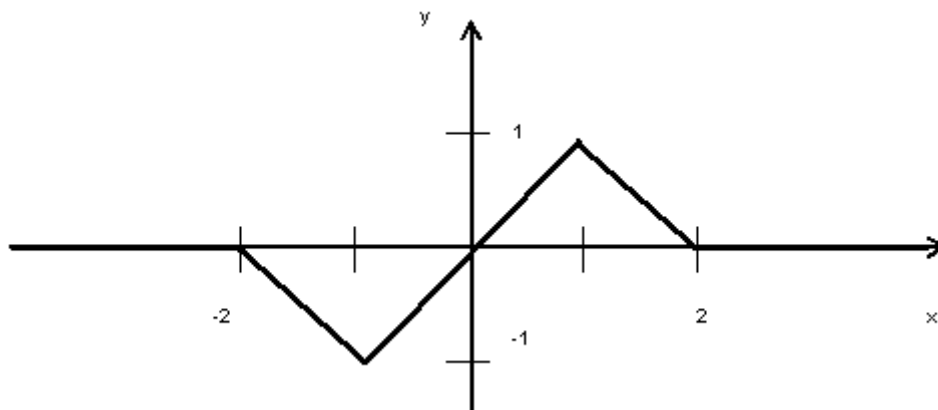


Рис.2

**Задание 1.** Вычислите 5 значений функции на разных участках интервала от -2 до 2. Результат вычислений представьте в виде таблицы, с указанием значений аргумента и полученных значений функции. В отчете представьте код программы, полученную таблицу значений функции и один скриншот ее работы.

### Оператор if-else

В качестве примера применения оператора **if-else** рассмотрим решение квадратного уравнения. Общий вид квадратного уравнения

$$a \cdot x^2 + b \cdot x + c = 0 .$$

Для его решения будем использовать метод вычисления детерминанта с последующим

сравнением полученного результата с нулем. Детерминант вычисляется по формуле

$$D = b^2 - 4 \cdot a \cdot c .$$

Если детерминант  $D < 0$  , то уравнение не имеет корней в области действительных чисел. Если детерминант  $D = 0$  , то уравнение имеет один корень, а если детерминант  $D > 0$  , то уравнение имеет два корня. Ниже представлена программа решения квадратного уравнения описанным методом, в которой используется оператор **if-else**.

```
#include <cstdlib>
#include <iostream>
#include <math.h> // в программе используется функция квадратного корня

using namespace std;

int main(int argc, char *argv[])
{
    double a, b, c, D, x1, x2; // объявление переменных
    cout << endl << "Введите a, b, c";
    cout << endl << "a = " ;
        cin >> a;
    cout << endl << "b = " ;
        cin >> b;
    cout << endl << "c = " ;
        cin >> c;

    D=b*b-4*a*c; // вычисление детерминанта
    if(D < 0) // проверка выполнения первого условия
    {
        cout << endl << "Уравнение не имеет корней" ;
        return EXIT_SUCCESS;
    }

    if(D==0) // проверка выполнения второго условия
    {
        x1 = -b/(2*a) ;
        cout << endl << "Уравнение имеет 1 корень:";
        cout << endl << " x = " << x1;
        return EXIT_SUCCESS;
    }
    else // если первые два условия не выполняются, то выполняется третье
    {
        x1=(-b+sqrt(D))/(2*a);
        x2=(-b-sqrt(D))/(2*a);
        cout << endl << " Уравнение имеет 2 корня:";
        cout << endl << " x1 = " << x1;
        cout << endl << " x2 = " << x2;
    }
    cin.get();
    return EXIT_SUCCESS;
}
```

Задание 2. Наберите и откомпилируйте код приведенной программы. Проверьте работу программы для следующих значений a, b и c:

- 1) a = 1; b = 2; c = 3;
- 2) a = 2; b = - 4; c = 2;
- 3) a = 1; b = 4; c = 3;
- 4) a = 2.3; b = -5.61; c = 1.091.

Задание 3.

Функция задана следующим образом:

$$y(x) = \begin{cases} x^2 - x - 1/2, & x \leq -1/2 \\ -x^2 - x + 2, & -1/2 < x \leq 1 \\ -x^2 + x + 1/2, & x > 1 \end{cases}$$

Напишите программу вычисления значений функции и вычислите 5 значений функции на разных участках интервала от -2 до 2. Результат вычислений представьте в виде таблицы, с указанием значений аргумента и полученных значений функции. В отчете представьте код программы, полученную таблицу значений функции и один скриншот ее работы.