Міністерство освіти і науки України
Харківський національний автомобільно-дорожній університет
Кафедра інформаційних технологій та мехатроніки

МЕТОДИЧНІ ВКАЗІВКИ

до лабораторних робіт з дисципліни

“Інформаційні технології”

“Програмування на мові С++ у середовищі Microsoft Visual Studio 2010”

для студентів напряму підготовки 6.050702 ” Електромеханіка ”,
галузь знань 0507 ” Електротехніка та електромеханіка ”

Розроблено та надруковано доц. Симбірським Г.Д.

Харків, 2015

Лабораторная работа № 2

РАЗРАБОТКА И ИССЛЕДОВАНИЕ РАЗВЕТВЛЯЮЩИХСЯ ПРОГРАММ В VISUAL C++ 2010

Цель работы: исследование разветвляющихся программ и получение практических навыков в их разработке.

1. Исследование условного оператора if…else

1.1. Любой алгоритм может быть записан на языке программирования с использованием только трех управляющих структур: последовательное выполнение, ветвление и повторение. Последовательное выполнение реализуется в линейных алгоритмах, исследованных в предыдущей лабораторной работе. В данной работе будут исследованы разветвленные алгоритмы, реализующие алгоритмы ветвления.

На рис. 2.1 приведен пример структуры ветвления. Операционный блок **Условие** состоит из выражения, содержащего логическое отношение, т. е. условие. Если условие выполняется, то реализуется **Действие 1** алгоритма, а в случае невыполнения - **Действие 2.**

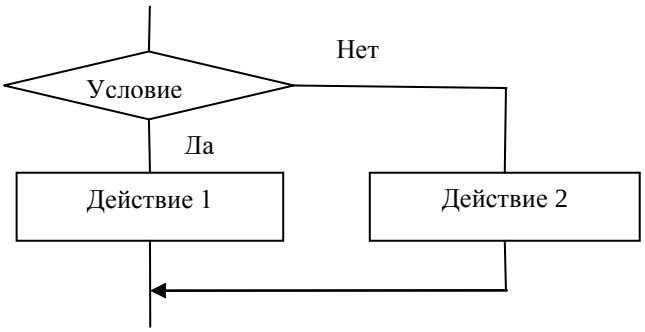

Рис.2.1. Структура ветвления

Структура ветвления описывается в языке C++ с помощью **условного оператора**:

```
if (выражение)
оператор_1;
else
оператор_2;
```

где часть **else оператор_2;** может отсутствовать.

Вначале вычисляется **выражение** в скобках. Если оно истинно, то выполняется **оператор_1**. Если **выражение** ложно, то **оператор_1** пропускается и выполняется **оператор_2.** Вместо единичных операторов могут использоваться группы из нескольких операторов (сложные операторы). При этом они заключаются в фигурные скобки - **{ }.**

Выражение в скобках представляет собой условие, заданное с помощью операций отношений и логических операций. В программировании постоянно приходится сравнивать числовые значения переменных, т. к. на этом построен процесс принятия решений при работе над различными проектами. Для реализации этого в языке C++ существует шесть базовых операторов для сравнения значений двух переменных:

< меньше; **<= меньше или равно**;
> больше; **>= больше или равно**;
== равно; **!= не равно**.

Сложные операторы. К сложным операторам относят собственно сложные операторы и блоки. В обоих случаях это последовательность операторов, помещенная в фигурные скобки. Блок отличается от сложного оператора наличием **объявлений переменных** в теле блока. Например,

```
{
a=b+c;          // сложный оператор
d=a+b;
}

{
int a,b,c,d;
a=b+c;          // блок
d=a+b;
}
```

Для лучшего понимания действия операторов в настоящей работе следует знать перевод следующих английских слов:

if – если;
then – тогда;
else – иначе;
case – случай;
switch – переключатель;
break – прерывать;
default – не выполнять.

Задание 2.1. Исследовать и выполнить программу для определения переменной **a** по следующему условию:

$$\mathbf{a} = \begin{cases} \mathbf{b + c + d}, & \text{если} \quad \mathbf{b > 10}; \\ \mathbf{b - c - d}, & \text{если} \quad \mathbf{b \le 10}. \end{cases}$$

Создайте проект **Lr2-1** в папке **d:\PE-11\Фамилия\Лр2** для разработки программы, производящей арифметические действия с тремя переменными **b**, **c** и **d** с использованием условного оператора **if…else** и операторов консольных ввода и вывода данных. Код (текст) программы с комментариями приведен ниже.

Блок-схема алгоритма решения задания 1 представлена на рис.2.2. Начертите ее в отчете по лабораторной работе.

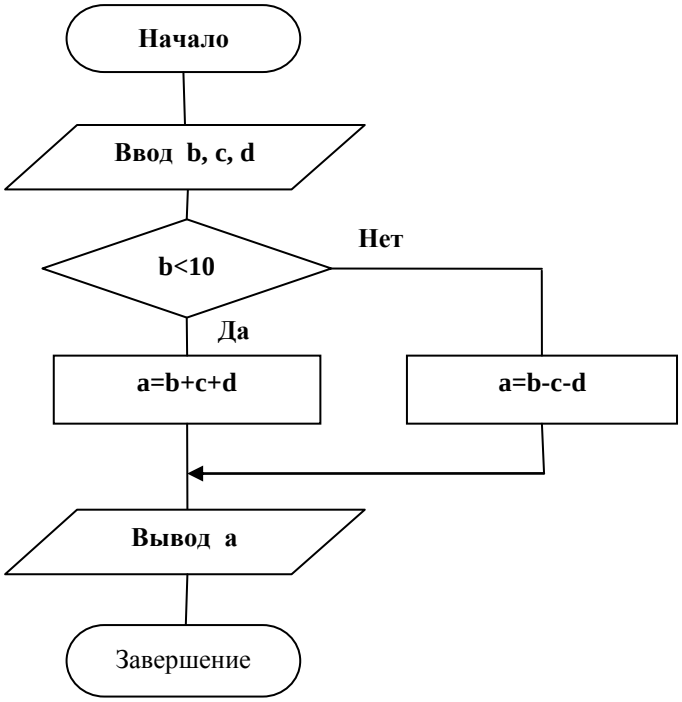

Рис. 2.2. Блок-схема алгоритма определения переменной **a** (простые операторы)

Введите и выполните программный код, реализующий алгоритм:

```
#include "stdafx.h"
#include <conio.h>                  //Файл conio.h обеспечивает задержку окна DOS на экране дисплея
#include <iostream>                 //Директива include подключает файл ввода–вывода iostream
using namespace std;                //Подключает все имена из пространства имен  std

int _tmain(int argc, _TCHAR* argv[])  //Объявление главной функции _tmain
{                                   //Начало главной функции
    int a, b;                       //Объявление переменных целого типа
    int c=2, d=3;                   //Объявление переменных целого типа  и их инициализация
        cout<<"Vvvedite b"<<endl;   //Вывод на экран надписи-приглашения  Vvedite b
        cin>>b;                     //Ввод значения переменной  b
        if (b<10)                   //Условный оператор if …else (1-я часть)
        a=b+c+d;                    //Вычисление переменной a при выполнении условия
        else                        //Условный оператор if …else (2-я часть)
        a=b-c-d;                    //Вычисление переменной a при невыполнении условия
        cout<<"a = "<<a;            //Вывод на экран значения переменной  a
    getch();                        //Функция задержки окна DOS на экране
    return 0;
}
```

Проведите вычисления для **b=8** и **b=12**, а результаты сохраните следующим образом.

Создайте в текстовом процессоре **Word** файл **Результат_Фамилия_Лр6**. Поля документа сделайте по 0,5 см.

Поместите окно DOS с результатами решения **Задания 2.1** в центральной части окна **Microsoft Visual Studio** ниже программного кода **Lr2-1.cpp** (см. рис. 1.2) и нажмите клавишу **<Prt Scr>**, после чего вставьте полученную копию экрана в файл **Результат_Фамилия_Лр2**. Над вставленным рисунком проставьте номер задания – **2-1**. Файл результатов не закрывайте до получения оценки за выполненную практическую часть работы в тетрадь с отчетом.

Закройте окно DOS, откройте пункт меню **Файл** и выполните команду **Закрыть решение**.

!Внимание! Результаты следующих заданий данной лабораторной работы сохраняйте строго в соответствии с приведенным выше порядком действий!

Задание 2.2. Исследовать и выполнить программу для определения переменной **a** по условию **Задания 1**. Отличие будет состоять в использовании сложных операторов в условном операторе **if…else**.

Блок-схема алгоритма решения данного задания представлена на рис.2.3. Обратите внимание на отличия данной блок-схемы от схемы, приведенной на рис. 2.2. Объясните их. Начертите блок-схему в отчете по лабораторной работе.

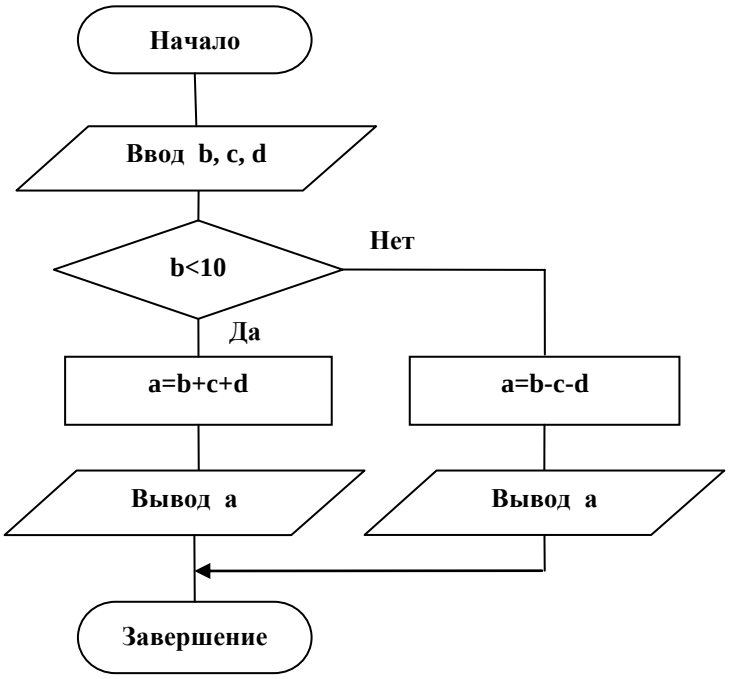

Рис. 2.3. Блок-схема алгоритма определения переменной **a** (сложные операторы)

Введите и выполните программный код, реализующий алгоритм. Обратите внимание на сдвиг фигурных скобок, ограничивающих сложные операторы, относительно скобок, охватывающих тело главной функции.

```
#include "stdafx.h"
#include <conio.h>                  //Файл conio.h обеспечивает задержку окна DOS на экране дисплея
#include <iostream>                 //Директива include подключает файл ввода–вывода iostream
using namespace std;                //Подключает все имена из пространства имен  std

int _tmain(int argc, _TCHAR* argv[])  //Объявление главной функции _tmain
{                                   //Начало главной функции
    int a, b;                       //Объявление переменных целого типа
    int c=2, d=3;                   //Объявление переменных целого типа  и их инициализация
        cout<<"Vvvedite b"<<endl;   //Вывод на экран надписи-приглашения  Vvedite b
        cin>>b;                     //Ввод значения переменной  b
        if (b<10)                   //Условный оператор if…else (1-я часть)
        {                           //Начало сложного оператора
        a=b+c+d;                    //Вычисление переменной a при выполнении условия
        cout<<"a=b+c+d= "<<a;       //Вывод на экран значения переменной  a
        }                           //Конец сложного оператора
        else                        //Условный оператор if …else (2-я часть)
        {                           //Начало сложного оператора
        a=b-c-d;                    //Вычисление переменной a при невыполнении условия
        cout<<"a=b-c-d= "<<a;       //Вывод на экран значения переменной  a
        }                           //Конец сложного оператора
getch();                            //Функция задержки окна DOS на экране
return 0;
}
```

Проведите вычисления для **b=8** и **b=12**, а результаты сохраните в файле **Результат_Фамилия_Лр2** в папке **d:\PE-11\Фамилия\Лр2** в подобном рис.1.2 виде. Над вставленным рисунком проставьте номер задания – **2-2**. При выполнении следующих заданий сохраняйте результаты таким же образом.

Задание 2.3. Самостоятельно разработайте и выполните программу для определения переменной **Y** по следующему условию:

$$\begin{cases} Y = \sqrt[3]{1 + \dfrac{\left(1 + x^3\right)^2}{4}}, & \text{если} \quad x \geq 5; \\ Y = \sin\dfrac{1-x}{1+x} + tg^4 5x, & \text{если} \quad x < 5. \end{cases}$$

Используйте условный оператор **if…else** и оператор консольных ввода и вывода данных аналогично **Заданию 1**. Обратите внимание на знаки отношений (>, >=,<=,…). Блок-схема алгоритма для решения данной задачи аналогична блок-схеме на рис. 2.1. Чертить ее необязательно. Результаты сохраните в **Результат_Фамилия_Лр2**.

Задание 2.4. Самостоятельно разработать алгоритм и программу для вычисления переменной **Y** (использовать оператор **if…else** и консольный ввод-вывод переменных). Определите свой номер варианта как номер компьютера.

Таблица 2.1 Исходные данные и формулы для расчета Y (Задание 2.4)

№ варианта	Формула для расчета **Y**	Значение **a**
1	$\begin{cases} Y = tg\dfrac{1-x}{1+x} + \sin^2 5x, & \text{если} \quad x < 0{,}5; \\ Y = tg^3(x+a) - \arccos^2(x+a), & \text{если} \quad x \geq 0{,}5; \end{cases}$	**a=3,5**
2	$\begin{cases} Y = \cos^3(x+a) - 7(x+a), & \text{если} \quad x < 0{,}7; \\ Y = \sin^3(x+a) - \cos^2(x+a), & \text{если} \quad x \geq 0{,}7; \end{cases}$	**a=2,55**
3	$\begin{cases} Y = tg\dfrac{1-x}{1+x} + \sin^2 5x, & \text{если} \quad x < 0{,}5; \\ Y = tg^3(x+a) - \arccos^2(x+a), & \text{если} \quad x \geq 0{,}5; \end{cases}$	**a=1,5**
4	$\begin{cases} Y = ctg\dfrac{1-x}{1+x} + \cos^2 5x, & \text{если} \quad x < 0{,}3; \\ Y = \sin^3(x+a) - tg^2(x+a), & \text{если} \quad x \geq 0{,}3; \end{cases}$	**a=1,44**
5	$\begin{cases} Y = \sin\dfrac{1-x}{1+x} + ctg^2 5x, & \text{если} \quad x < 0{,}48; \\ Y = \cos^3(x+a) - tg^2(x+a), & \text{если} \quad x \geq 0{,}48; \end{cases}$	**a=2,4**
6	$\begin{cases} Y = ctg^3(x+a) - \sin^2(x+a), & \text{если} \quad x < 0{,}7; \\ Y = tg^3(x+a) - \cos^2(x+a), & \text{если} \quad x \geq 0{,}7; \end{cases}$	**a=1,1**
7	$\begin{cases} Y = tg\dfrac{1-x}{1+x} + \sin^2 5x, & \text{если} \quad x < 0{,}5; \\ Y = tg^3(x+a) - \arccos^2(x+a), & \text{если} \quad x \geq 0{,}5; \end{cases}$	**a=2,6**
8	$\begin{cases} Y = tg\dfrac{1-x}{1+x} + \sin^2 5x, & \text{если} \quad x < 0{,}5; \\ Y = tg^3(x+a) - \arccos^2(x+a), & \text{если} \quad x \geq 0{,}5; \end{cases}$	**a=1,9**
9	$\begin{cases} Y = \cos^3(x+a) - 7(x+a), & \text{если} \quad x < 0{,}7; \\ Y = \sin^3(x+a) - \cos^2(x+a), & \text{если} \quad x \geq 0{,}7; \end{cases}$	**a=3,1**
10	$\begin{cases} Y = ctg\dfrac{1-x}{1+x} + \cos^2 5x, & \text{если} \quad x < 0{,}3; \\ Y = \sin^3(x+a) - tg^2(x+a), & \text{если} \quad x \geq 0{,}3; \end{cases}$	**a=2,2**

11	$\begin{cases} Y = \sin^3(x+a) - \cos^2(x+a), & \text{если } x < 0{,}37; \\ Y = ctg^2 3x + tg^3(x+a)e^{3a}, & \text{если } x \geq 0{,}37; \end{cases}$	**a=1,1**
12	$\begin{cases} Y = ctg^2 3x + e^{3a}, & \text{если } x < 0{,}8; \\ Y = \sin^3(x+a) - tg^2(x+a), & \text{если } x \geq 0{,}8; \end{cases}$	**a=3,2**
13	$\begin{cases} Y = \cos^3(x+a) - 7(x+a), & \text{если } x < 0{,}7; \\ Y = \sin^3(x+a) - \cos^2(x+a), & \text{если } x \geq 0{,}7; \end{cases}$	**a=1,2**
14	$\begin{cases} Y = tg\frac{1-x}{1+x} + \sin^2 5x, & \text{если } x < 0{,}44; \\ Y = \cos^3(x+a) - tg^2(x+a), & \text{если } x \geq 0{,}44; \end{cases}$	**a=2,8**
15	$\begin{cases} Y = tg\frac{1-x}{1+x} + \sin^2 5x, & \text{если } x < 0{,}5; \\ Y = tg^3(x+a) - \arccos^2(x+a), & \text{если } x \geq 0{,}5; \end{cases}$	**a=3,5**
16	$\begin{cases} Y = ctg^2 3x + e^{3a}, & \text{если } x < 0{,}8; \\ Y = \sin^3(x+a) - tg^2(x+a), & \text{если } x \geq 0{,}8; \end{cases}$	**a=1,3**
17	$\begin{cases} Y = \cos^3(x+a) - 7(x+a), & \text{если } x < 0{,}7; \\ Y = \sin^3(x+a) - \cos^2(x+a), & \text{если } x \geq 0{,}7; \end{cases}$	**a=1,7**
18	$\begin{cases} Y = tg\frac{1-x}{1+x} + \sin^2 5x, & \text{если } x < 0{,}9; \\ Y = tg^3(x+a) - \arccos^2(x+a), & \text{если } x \geq 0{,}9; \end{cases}$	**a=2,6**
19	$\begin{cases} Y = ctg\frac{1-x}{1+x} + \cos^2 5x, & \text{если } x < 0{,}4; \\ Y = \sin^3(x+a) - tg^2(x+a), & \text{если } x \geq 0{,}4; \end{cases}$	**a=1,9**
20	$\begin{cases} Y = tg\frac{1-x}{1+x} + \sin^2 5x, & \text{если } x < 0{,}2; \\ Y = tg^3(x+a) - \arccos^2(x+a), & \text{если } x \geq 0{,}2; \end{cases}$	**a=2,0**

2. Использование условного оператора if…else для 3-х интервалов значений переменой

Выше исследовались случаи применения условного оператора **if…else** для 2-х интервалов значений переменой, т. е. рассматривались простые условия (логические выражения) типа **x<a**. При этом число **a** делит числовую ось **x** на два интервала (рис. 2.4).

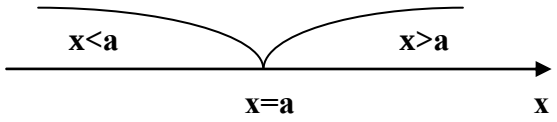

Рис. 2.5. Графическая интерпретация логического выражения для двух интервалов значений **x**

Язык C++ и среда Visual C++2010 позволяют в случае необходимости применять в условном операторе **if…else** и более сложные логические выражения с использованием различных логических операций.

Рассмотрим случай, когда в условии оператора **if…else** указан некоторый интервал значений, с которым сравнивается переменная. Например, переменная **x** должна находиться в интервале значений от **a** до **b** (**a<b**) (рис. 2.6). В этом случае условие запишется следующим образом:

if (x>=a && x<=b) ,

где **&&** - операция логического **И**.

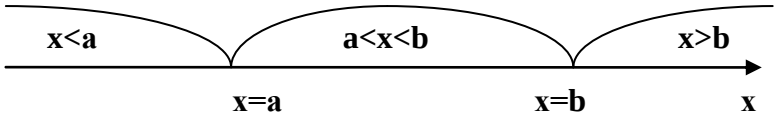

Рис. 2.6. Графическая интерпретация логического выражения для трех интервалов значений **x**

3. Исследование вложенного условного оператора if…else

Во многих случаях использование простого (одинарного) оператора **if…else** не обеспечивает решение поставленной задачи. Очевидно, что данный оператор, например, не позволяет вычислительному процессу принять решение в случае, когда для какой-то переменной существует четыре и более интервала ее значений. В этом случае применяют вложенный условный оператор **if…else**.

На рис. 2.5 приведен пример структуры вложенного условного оператора **if…else** (в качестве базовой использована структура, изображенная на рис. 2.1).

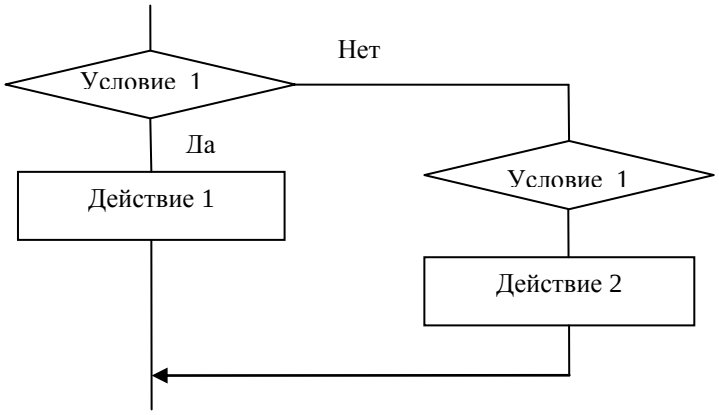

Рис.2.1. Структура вложенного условного оператора **if…else**

4. Оператор выбора switch

Оператор **switch** (переключатель) предназначен для разветвления процесса вычислений на несколько направлений. Он обеспечивает выбор из нескольких вариантов на основании некоторого фиксированного параметра. Формат оператора следующий:

```
switch (выражение)
{
    case константа_1: оператор_1;
    case константа_2: оператор_2;
    case константа_3: оператор_3;
    ...
    case константа_n: оператор_n;
    [default: оператор;]
}
```

Блок-сема алгоритма, реализующего работу оператора выбора switch, приведена на рис. 2.4.

Выполнение оператора **switch** начинается с вычисления выражения в скобках (оно должно быть целочисленным). Его значение последовательно сравнивается с константами, которые записаны следом за каждым оператором **case**. Затем управление передается тому оператору, который помечен константным выражением (меткой), значение которого совпало с вычисленным выражением в условии. После этого последовательно выполняются все остальные ветви, если выход из переключателя явно не указан.

Все константные выражения должны иметь разные значения, но быть одного и того же целочисленного типа. Несколько меток могут следовать подряд. Если совпадения не произошло, выполняются операторы, расположенные после слова **default** (а при его отсутствии управление передается следующему за **switch** оператору).

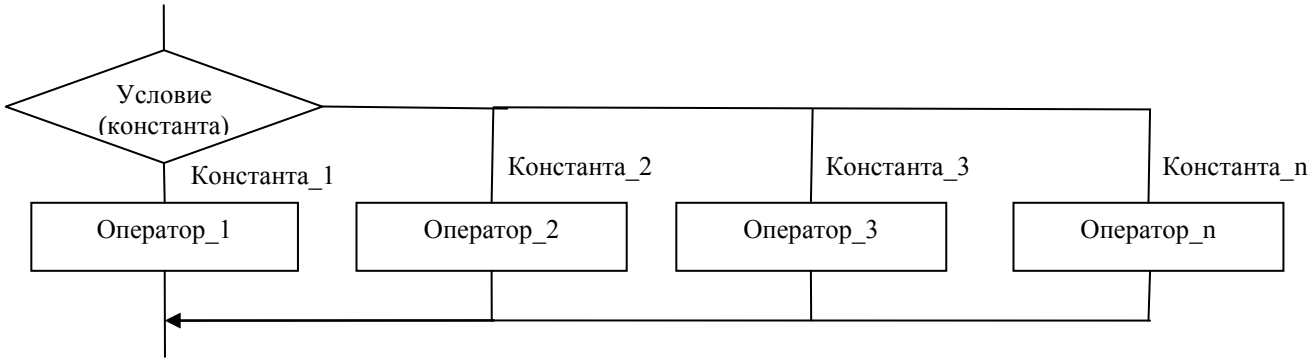

Рис. 2.4. Блок-схема алгоритма работы оператора **switch**

Задание 2.5. Проанализируем работу оператора **switch** на примере программы, реализующей простейший калькулятор на 4 действия. Здесь консольно вводятся две переменные **a** и **b** и знак операции, которую необходимо совершить с этими переменными. Оператор **switch** сравнивает введенный знак операции со знаком, содержащемся в четырех метках **case**, и производит необходимую арифметическую операцию. Результат выводится на экран. Если знак операции не совпадает с содержащимися в метках, то на экран выводится сообщение о неизвестной операции.

```
#include "stdafx.h"
#include <conio.h>                          //Файл conio.h обеспечивает задержку окна DOS на экране дисплея
#include <iostream>                         //Директива include подключает файл ввода–вывода iostream
using namespace std;                        //Подключает все имена из пространства имен std

int _tmain(int argc, _TCHAR* argv[])        //Объявление главной функции _tmain
{                                           //Начало главной функции
double a, b, res;                           //Объявление вещественных переменных
char operation;                             //Объявление символьной переменной
cout << "Vvedite a : "<<endl;               //Вывод на экран надписи-приглашения Vvedite a :
cin >> a;                                   //Ввод значения переменной a
cout << "Vvedite operation : "<<endl;       //Вывод на экран надписи-приглашения Vvedite operation :
cin >> operation;                           //Ввод знака операции
cout << "Vvedite b : "<<endl;               //Вывод на экран надписи-приглашения Vvedite b :
cin >> b;                                   //Ввод значения переменной b
    switch (operation)                      //Условие (с чем будет сравнение) в операторе switch
    {                                       //Начало составного оператора
        case '+': res = a + b; break;       //Метка ”+” и вычисление переменной res для этой метки
        case '-': res = a - b; break;       //Метка ”-” и вычисление переменной res для этой метки
        case '*': res = a * b; break;       //Метка ”*” и вычисление переменной res для этой метки
        case '/': res = a / b; break;       //Метка ”/” и вычисление переменной res для этой метки
        default : cout <<"Unknown operation"<<endl;   //Вывод сообщения о неизвестной операции
    }                                       //Конец сложного оператора
cout << "Result : " << res;                 //Вывод на экран результата вычислений
getch();                                    //Функция задержки окна DOS на экране
return 0;
}                                           //Конец главной функции
```

5. Безусловный оператор goto

Переход к любому оператору программы без условия (директивный переход) в среде Visual C++2010 осуществляется с помощью оператора безусловного перехода **go to**. В отличие от оператора **if…else** этот оператор осуществляет переход в нужное место программы без выполнения каких-либо условий. Оператор имеет следующий вид:

goto Метка;

Меткой обозначают какой-либо оператор, на который должен быть осуществлен переходи з места установки оператора **goto**. В качестве метки выступает идентификатор с расположенным за ним символом двоеточия:

Метка: оператор;

Как только выполнение программы достигает оператора **goto**, управление передается оператору, помеченному меткой А.

Задание 2.6. Проанализируем работу оператора **goto** на примере программы из задания 2. Расположим оператор **goto** с меткой **A** перед условным оператором, а метку **A** – перед концом программы. Теперь условный оператор не выполнится, т. к. компьютер выполнит оператор **goto A** и перейдет сразу к выводу надписи **"Perehod po metke ";**

```
#include "stdafx.h"
#include <conio.h>                  //Файл conio.h обеспечивает задержку окна DOS на экране дисплея
#include <iostream>                 //Директива include подключает файл ввода–вывода iostream
using namespace std;                //Подключает все имена из пространства имен  std

int _tmain(int argc, _TCHAR* argv[])  //Объявление главной функции _tmain
{                                   //Начало главной функции
    int a, b;                       //Объявление переменных целого типа
    int c=2, d=3;                   //Объявление переменных целого типа  и их инициализация
        cout<<"Vvvedite b"<<endl;   //Вывод на экран надписи-приглашения  Vvedite b
        cin>>b;                     //Ввод значения переменной  b
            goto A;                 //Оператор goto
        if (b<10)
        {
        a=b+c+d;
        cout<<"a=b+c+d= "<<a;
        }
        else
        {
        a=b-c-d;
        cout<<"a=b-c-d= "<<a;
        }
    A: cout<<"Perehod po metke ";   //Метка A и оператор вывода
    getch();                        //Функция задержки окна DOS на экране
    return 0;
}
```

Задание 2.7. Самостоятельно разработать алгоритм и программу решения задачи (использовать оператор **if…else**). Определите свой номер варианта как номер компьютера.

Варианты заданий

Вариант № 1. Написать программу, которая вычисляет частное двух чисел. Программа должна проверять правильность введенных данных и, если они неверные (делитель равен нулю), выдавать сообщение об ошибке. Рекомендованный вид экрана:

Вычисление частного
Введите в одной строке делимое и делитель. Нажмите <Enter>
12 0
Вы ошиблись. Делитель не должен быть равен нулю

Вариант № 2. Написать программу вычисления площади кольца. Программа должна проверять правильность введенных данных. Рекомендованный вид экрана:

Вычисление площади кольца.
Введите начальные данные:
Радиус кольца (см) 3.5
Радиус отверстия (см) 7
Ошибка. Радиус отверстия не может быть больше радиуса кольца.

Вариант № 3. Написать программу вычисления стоимости покупки с учетом скидки. Скидка в 10% предоставляется, если сумма покупки более 1000 грн. Рекомендованный вид экрана:

Вычисление стоимости покупки с учетом скидки 10%.
Введите сумму покупки: 1200
Сумма покупки: 1080.00 грн.

Вариант № 4. Написать программу вычисления стоимости покупки с учетом скидки. Скидка в 3% предоставляется, если сумма покупки более 500 грн., в 5% -- если сумма более 800 грн. Рекомендованный вид экрана:

Вычисление стоимости покупки с учетом скидки.
Введите сумму покупки: 640
Вам предоставляется скидка 3%
Сумма покупки: 620.80 грн.

Вариант №5. Написать программу для проверки знания даты начала второй мировой войны. В случае неправильного ответа, программа должна выводить правильный ответ. Рекомендованный вид экрана:

В каком году началась вторая мировая война?
Введите число и нажмите <Enter>
1939

Правильно

Вариант № 6. Написать программу, которая сравнивает два введенных с клавиатуры числа. Программа должна указать, какое число больше, или, если числа равны, вывести соответствующее сообщение. Рекомендованный вид экрана:

Введите в одной строке два целых числа: 34 67
34 меньше 67

Вариант № 7. Написать программу, которая проверяет, является ли введенное целое число четное. Рекомендованный вид экрана:

Введите целое число: 23
Число 23 – нечетное.

Вариант № 8. Написать программу, которая проверяет, делится ли на три введенное с клавиатуры целое число. Рекомендованный вид экрана:

Введите целое число: 451
Число 451 нацело на три не делится.

Вариант № 9. Написать программу, которая вычисляет частное двух чисел. Программа должна проверять правильность введенных данных и, если они неверные (делитель равен нулю), выдавать сообщение об ошибке. Рекомендованный вид экрана:

Вычисление частного
Введите в одной строке делимое и делитель. Нажмите <Enter>
12 0
Вы ошиблись. Делитель не должен быть равен нулю

Вариант № 10. Написать программу, которая вычисляет оптимальный вес, сравнивает его с реальным и выдает рекомендацию о необходимости пополнеть или похудеть. Оптимальный вес вычисляется по формуле: **рост (см) – 100.** Рекомендованный вид экрана:

Введите в одной строке через пропуск рост (см) и вес (кг): 170 68
Вам нужно пополнеть на 2,00 кг

Вариант № 11. Написать программу, которая вычисляет частное двух чисел. Программа должна проверять правильность введенных данных и, если они неверные (делитель равен нулю), выдавать сообщение об ошибке. Рекомендованный вид экрана:

Вычисление частного
Введите в одной строке делимое и делитель. Нажмите <Enter>
12 0
Вы ошиблись. Делитель не должен быть равен нулю

Вариант № 12. Написать программу вычисления площади кольца. Программа должна проверять правильность введенных данных. Рекомендованный вид экрана:

Вычисление площади кольца.
Введите начальные данные:
Радиус кольца (с) 3.5
Радиус отверстия (см) 7
Ошибка. Радиус отверстия не может быть больше радиуса кольца.

Вариант № 13. Написать программу вычисления стоимости покупки с учетом скидки. Скидка в 10% предоставляется, если сумма покупки более 1000 грн. Рекомендованный вид экрана:

Вычисление стоимости покупки с учетом скидки 10%.
Введите сумму покупки: 1200
Сумма покупки: 1080.00 грн.

Вариант № 14. Написать программу вычисления стоимости покупки с учетом скидки. Скидка в 3% предоставляется, если сумма покупки более 500 грн., в 5% -- если сумма более 800 грн. Рекомендованный вид экрана:

Вычисление стоимости покупки с учетом скидки.
Введите сумму покупки: 640
Вам предоставляется скидка 3%
Сумма покупки: 620.80 грн.

Вариант № 15. Написать программу для проверки знания даты начала второй мировой войны. В случае неправильного ответа, программа должна выводить правильный ответ. Рекомендованный вид экрана:

В каком году началась вторая мировая война?
Введите число и нажмите <Enter>
1939
Правильно

Вариант № 16. Написать программу, которая сравнивает два введенных с клавиатуры числа. Программа должна указать, какое число больше, или, если числа равны, вывести соответствующее сообщение. Рекомендованный вид экрана:

Введите в одной строке два целых числа: 34 67
34 меньше 67

Вариант № 17. Написать программу, которая проверяет, является ли введенное целое число четное. Рекомендованный вид экрана:

Введите целое число: 23
Число 23 – нечетное.

Вариант № 18. Написать программу, которая проверяет, делится ли на три введенное с клавиатуры целое число. Рекомендованный вид экрана:

Введите целое число: 451
Число 451 нацело на три не делится.

Вариант № 19. Написать программу, которая вычисляет частное двух чисел. Программа должна проверять правильность введенных данных и, если они неверные (делитель равен нулю), выдавать сообщение об ошибке. Рекомендованный вид экрана:

Вычисление частного
Введите в одной строке делимое и делитель. Нажмите <Enter>
12 0
Вы ошиблись. Делитель не должен быть равен нулю

Вариант № 20. Написать программу, которая вычисляет оптимальный вес, сравнивает его с реальным и выдает рекомендацию о необходимости пополнеть или похудеть. Оптимальный вес вычисляется по формуле: **рост (см) – 100.** Рекомендованный вид экрана:

Введите в одной строке через пропуск рост (см) и вес (кг): 170 68
Вам нужно набрать 2,00 кг

Контрольные вопросы

1. Какие операторы языка С++ используются для реализации разветвляющихся вычислительных процессов?
2. В каких случаях необходимо применение условных операторов?
3. Запишите условный оператор **if…else** в общем виде и опишите порядок его выполнения.
4. Чем отличается блок операторов от составного оператора?
5. Запишите в общем виде оператор **switch** и опишите порядок его выполнения.
6. Для чего необходим оператор **break**?
7. В каких случаях необходимо применение оператора **goto**?
8. Запишите в общем виде оператор **goto** и опишите порядок его выполнения