

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Харківський національний автомобільно-дорожній університет  
Кафедра інформаційних технологій і мехатроніки

**МЕТОДИЧНІ ВКАЗІВКИ  
до виконання контрольної роботи**

з дисципліни “Інформаційні технології ”  
(розділ 2 “Програмування на С++”)  
для студентів 2-го курсу ЦОП  
галузі знань 14 “Електрична інженерія”  
спеціальність 141 “Електроенергетика, електротехніка та  
електромеханіка” (для російськомовних студентів)

Методичні вказівки розроблені доцентом кафедри інформатики та прикладної математики кандидатом технічних наук Г. Д. Симбірським

## ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

### 1. Общие сведения о программе Visual Studio 2010

**Visual Studio 2010** — это полный набор средств для разработки программ и приложений для компьютеров, мобильных устройств и др. с использованием различных языков программирования. Среда разработки Visual Studio 2010 предназначена для написания программ и приложений на языках .NET, HTML, JavaScript, C# и C++. Возможности этого программного инструмента очень велики. Поэтому он получил название **ИСП - интегрированная среда разработки** (Integrated Development Environment – **IDE**). Не выходя за рамки этой среды, можно решать следующие задачи:

1. Формировать заготовки приложений без написания текстов программ;
2. Просматривать проекты несколькими разными способами;
3. Редактировать файлы заголовков и текстов программ;
4. Формировать визуальный графический интерфейс (меню и диалоговые окна);
5. Компилировать и компоновать программы;
6. Производить отладку разрабатываемых программ и приложений в процессе их выполнения.

**Visual C++ 2010** - одна из составных частей программы Visual Studio 2010 и является самодостаточной средой для разработки различных программ и приложений. Visual C++ 2010 включает следующие основные компоненты:

1. **Редактор** – для ввода, просмотра и проверки исходного кода;
2. **Компилятор** – для трансляции начального кода C++ в объектный код;
3. **Компоновщик** – создает выполняемые файлы, объединяя объектный код и библиотечные модули;
4. **Библиотеки** – сборники стандартных программ-модулей, которые можно применять в разрабатываемых программах и приложениях. Одна из самых важных библиотек - Microsoft Foundation Classes (базовые классы Microsoft или MFC), используемая при написании программ, работающих под управлением Microsoft Windows. Кроме этого, стандартные библиотеки C++ поддерживают операции ввода-вывода и другие стандартные возможности языка;
5. Прочие инструментальные средства, включая AppWizard (Мастер приложений), ClassWizard (Мастер классов) и Resource Editor (Редактор ресурсов).

### 2. Интерфейс программы Visual Studio 2010

Интерфейс Visual Studio 2010 нагляден и удобен для создания, редактирования, просмотра и отладки компонентов разрабатываемых программ и приложений – ресурсов, классов, файлов и др. Основной экран разделен на части с различными функциями, размеры которых можно изменять по своему усмотрению. Для облегчения решения стандартных задач есть контекстные меню, доступ к которым осуществляется с помощью нажатия правой кнопки мыши на различных компонентах изображения на экране.

С помощью Visual C++ 2010 можно работать с разрабатываемым приложением как с проектом.

**Проект** - это регламентированный (определенный протоколами Visual Studio 2010) набор файлов: заголовков, текстов программ, ресурсов, установок, конфигураций и др. Интерфейс Visual Studio 2010 дает возможность работать со всеми компонентами проекта одновременно, поэтому экран разделен на несколько зон (окон). Каждая разрабатываемая программа (приложение), даже самая простая, является проектом.

Набор папок и файлов, создаваемый средой Visual C++ 2010 при разработке проектов, называется **решением**.

При запуске программы Visual Studio 2010 открывается окно **Начальная страница – Microsoft Visual Studio (Администратор)**, состоящее из трех частей:

1. **Обозреватель решений** – окно в левой части экрана, в котором в дальнейшем будут представлены в виде дерева папки и файлы текущего проекта. В случае необходимости на месте данного окна можно открыть **Окно классов** данного проекта, его **Диспетчер свойств** или **Командный обозреватель**;

2. Окно **Начальная страница** в правой части экрана, в котором предложены на выбор возможные действия пользователя, в частности, **Создать проект** или **Открыть проект**, а также для открытия предложен список последних проектов;

3. Окно **Вывод** в нижней части экрана, в котором при построении решений проектов выводятся сообщения о предупреждениях (**warning**) и ошибках (**error**). На месте этого окна можно открыть **Окно определения кода** или окно **Результаты поиска**.

При создании консольных приложений на языке C++ работа осуществляется в окне **<Имя проекта> – Microsoft Visual Studio (Администратор)**, состоящем из трех частей:

1. **Обозреватель решений** (описание см. выше);
2. Окно для ввода программного кода и для работы с программой в правой части экрана;
3. Окно **Вывод** (описание см. выше).

### 3. Редактирование и отладка программ в Visual C++ 2010

Код (текст) программы вводится в окне редактора с использованием клавиатуры и основных приемов работы с текстом в операционной системе (ОС) Windows. При отображении текста программы используется синтаксическое раскрашивание. По умолчанию текст программы черного цвета с комментариями зеленого цвета и ключевыми и служебными словами синего цвета.

После того, как набран программный код, следует приступить к отладке программе. Для этого следует открыть меню **Построение** и выбрать пункт **Построить решение**. После чего все сообщения о предупреждениях и ошибках отображаются в окне **Вывод**. Ошибки следует исправлять обязательно, т. к. программа с ошибками выполняться не будет. С предупреждениями программа будет выполняться, однако их необходимо проанализировать и, если возможно, исправить или учесть. Описание ошибки находится в строке сообщения об ошибке (для получения подробного описания ошибки следует нажать **<F1>**). Чтобы исправить ошибку необходимо дважды щелкнуть в окне отладчика на сообщении о данной ошибке, после чего в окне редактора появится указатель на строке с ошибкой. После исправления всех ошибок

необходимо открыть пункт меню **Отладка**. Программа запускается при выборе пункта меню **Начать отладку** или при нажатии клавиши <F5>.

#### 4. Данные и переменные в языке C++

##### 4.1. Типы данных в языке C++.

Основные или базовые типы данных в языке C++ следующие:

**int** – целочисленные данные (4 байта), диапазон значений: целые от -2 147 483 647 до 2 147 483 647;

**char** – символьные данные (1 байт), диапазон значений: от 0 до 255 или от -127 до 128;

**float** – числа с плавающей запятой (4 байта), диапазон значений: от 3.4E-38 до 3.4E+38;

**double** – число с плавающей запятой двойной точности (8 байт), диапазон значений: от 1.7E-308 до 1.7E+308;

**bool** – логические переменные (**true** и **false**).

Кроме вышеприведенных основных типов данных, в языке C++ используются еще несколько типов данных.

##### 4.2. Объявление переменных и констант в языке C++

Фрагмент памяти, в котором хранится элемент данных и к которому можно обращаться по некоторому имени, называется **переменной**. Имена переменных могут включать буквы латинского алфавита **A – z** (в верхнем или нижнем регистре), цифры от 0 до 9 и знак подчеркивания. Имена переменных должны начинаться либо с буквы, либо со знака подчеркивания. В C++ принято назначать имена переменных с прописных букв, а классов – с заглавных. Компилятор C++ различает прописные и строчные буквы, например, **Sum** и **sum** означают разные переменные.

**Объявление** переменной с одновременным заданием типа хранимого под ее именем элемента данных осуществляется с использованием следующего синтаксиса:

##### ТипПеременной ИмяПеременной;

Например, строка **int arg;** объявляет целочисленную переменную с именем **arg**.

В языке C++ есть зарезервированные слова, имеющие специальное значение – **ключевые слова**. Это названия типов данных и некоторых операторов и др. Редактор среды разработки Visual C++ 2010 подсвечивает их **синим** цветом. Имена переменных не должны совпадать с ключевыми словами.

Объявляя переменную, можно сразу присвоить ей начальное значение. Например

**int sum=0;** или **int sum(0);**

**float a=2.7;** или **float a(2.7);**

Переменная объявляется **перед** тем местом, где она будет впервые задействована. Подробнее о месте объявления переменных в C++ будет сказано ниже.

Переменную, не меняющую своего значения в программе, можно использовать как константу. В C++ объявление константы выглядит следующим образом:

**const Тип ИмяКонстанты = Значение; .**

Например, объявить постоянную  $\pi$  можно следующим образом:

**const float pi = 3.14159; .**

#### 5. Выражения и операции в среде Visual C++ 2010

В языке C++ следующим уровнем представления данных после переменных и констант являются выражения.

**Выражение** – это некоторая допустимая комбинация переменных, констант, функций и знаков операций для вычислений в программах. Выражения в языке C++ записываются в строчку.

Например, формула

$$d = \frac{a + b(c + e)}{c(a + b) + t}$$

в языке C++ запишется следующим образом: **d = (a + b\*(c + e))/(c\*(a + b) + t) .**

В приведенном выше выражении знак “=” обозначает операцию **присваивания**, которая выполняется следующим образом. Вычисляется значение выражения в правой части и присваивается переменной **d**.

В данной лабораторной работе будут использоваться привычные для студентов арифметические операции; в полном объеме операции среды Visual C++ 2010 приведены в лекционном материале по курсу и будут исследованы в следующих работах. Приоритет (очередность выполнения) арифметических операций такой же, как и в математике (см. таб. 1).

Математические действия с переменными, константами и функциями в языке C++ записываются только в строчку, при этом для соблюдения необходимой по условию задачи очередности операций используются круглые скобки.

**Таблица 1. Арифметические операции в языке C++**

Операция	Знак в языке C++
Сложение	+
Вычитание	-
Умножение	*
Деление	/

#### 6. Стандартные математические функции в среде Visual C++ 2010

В языке C++ в выражения можно вставлять стандартные математические функции, которые вызываются из библиотеки **<math.h>**. Перечень математических функций, которые чаще всего встречаются в вычислениях приведены в таблице 2.

**Таблица 2. Основные стандартные математические функции (библиотека math.h)**

Название функции	Что вычисляет	Тип данных функции и аргумента
<b>abs(x)</b>	Абсолютное значение (модуль) аргумента $ x $	<b>int abs(int x)</b>
<b>exp(x)</b>	Экспонента $e^x$	<b>double exp(double x)</b>
<b>log(x)</b>	Натуральный логарифм $\ln x$	<b>double log(double x)</b>
<b>log10(x)</b>	Десятичный логарифм $\lg x$	<b>double log10(double x)</b>
<b>pow(x,y)</b>	Возведение в степень $x^y$	<b>double pow(double x, double y)</b>
<b>sqrt(x)</b>	Квадратный корень $\sqrt{x}$	<b>double sqrt(double x)</b>
<b>fmod(x,y)</b>	Остаток от деления $x/y$	<b>double fmod(double x, double y)</b>
<b>sin(x)</b>	Синус (угол задается в радианах)	<b>double sin(double x)</b>
<b>asin(x)</b>	Арксинус (угол задается в радианах от $-1$ до $+1$ )	<b>double asin(double x)</b>
<b>cos(x)</b>	Косинус (угол задается в радианах)	<b>double cos(double x)</b>
<b>acos(x)</b>	Арккосинус (угол задается в радианах от $-1$ до $+1$ )	<b>double acos(double x)</b>
<b>tan(x)</b>	Тангенс (угол задается в радианах)	<b>double tan(double x)</b>
<b>atan(x)</b>	Арктангенс (угол задается в радианах)	<b>double atan(double x)</b>

При обращении к этим функциям необходимо придерживаться следующих правил:

- 1) **x** и **y** должны быть типа **double**;
- 2) углы (аргументы) в тригонометрических функциях задаются в радианах.
- 3) вычисляемые функциями данные имеют тип **double**.

Например, выражение  $y = \sin^3(x^4) \frac{e^x + z^5 - 4.5 \cdot 10^2 \sqrt{x}}{\operatorname{tg}(a)(z^x + b)}$  на языке C++ будет иметь вид:

$$y = \operatorname{pow}(\sin(\operatorname{pow}(x,4)),3) * (\exp(x) + \operatorname{pow}(z,5) - 4.5 * 10 * 10 * \operatorname{sqrt}(x)) / (\tan(a) * (\operatorname{pow}(z,x) + b)) .$$

### 7. Алгоритмизация задач

Алгоритмизация задачи – это представление процесса решения данной задачи в виде последовательности простых операций, необходимых для получения такого решения. Как правило, алгоритм решения задачи представлен в виде блок-схемы, в которой показаны все необходимые для решения задачи операции в строгой последовательности. Состоят блок-схемы алгоритмов из стандартных обозначений.

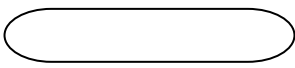

В схемах алгоритмов операционные блоки соединяются между собой линиями потока в виде стрелок. Допускается линии потока, которые идут сверху вниз и слева направо изображать без стрелок. При необходимости блок-схему могут сопровождать комментарии.

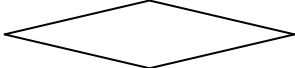
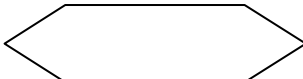






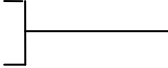
Алгоритмизацию решаемых задач рассмотрим на примере программ с линейной структурой, которые являются самыми простыми и используются, как правило, для реализации вычислений по формулам. В программах с линейной структурой операторы выполняются последовательно один за другим. Алгоритмизация более сложных задач будет исследована при выполнении следующих лабораторных работ.

В таблице 3 представлены условные обозначения основных операционных блоков схем алгоритмов.

Исследуем процесс разработки алгоритма программы с линейной структурой на следующем примере.

**Таблица 3. Основные операционные блоки схем алгоритмов**

№ п/п	Условное обозначение	Наименование операции	Описание операции
1		Начало, завершение	Начало и завершение алгоритма
2		Процесс	Вычислительная операция или их совокупность

3		Решение	Проверка условия и выбор дальнейшего направления процесса решения
4		Модификация	Заголовок цикла, проверка условий цикла
5		Данные	Ввод исходных данных, вывод данных и результатов
6		Типовой процесс	Использование ранее созданных алгоритмов, подпрограмм, функций
7		Печать документа	Вывод данных на печать
8		Соединитель внутрестраничный	Разрыв линий потока в пределах одной страницы
9		Соединитель межстраничный	Перенос линий потока на другую страницу
10		Узел	Слияние линий потока
11		Комментарии	Описание операционного блока

**Задача.** Необходимо вычислить силу тока  $I$  в новогодней гирлянде, состоящей из  $n=50$  электрических лампочек сопротивлением  $r=20$  Ом. Используя закон Ома и формулу для расчета суммарного сопротивления последовательной цепи, составим блок-схему алгоритма (рис. 1).

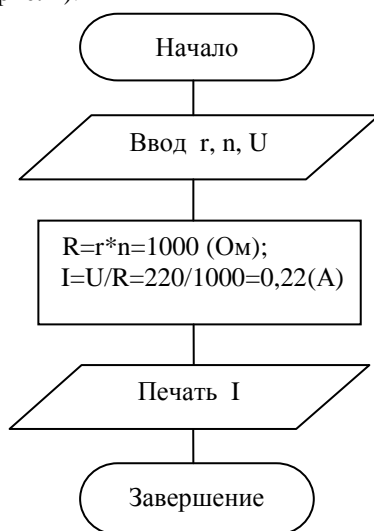


Рис. 1. Блок-схема алгоритма определения силы тока в новогодней электрогирлянде

### 8. Ввод и вывод данных в среде Visual C++ 2010

Лабораторные работы будут выполняться в Консольном приложении Win32 среды программирования Visual C++ 2010 консольный (с использованием клавиатуры и экрана дисплея) ввод данных производится при помощи оператора **cin**. В C++ этот оператор называется также потоком ввода. Например, для ввода значений трех переменных надо записать:

**cin>>a >>b>>c;**

где >> - символ операции извлечения данных из потока; **a**, **b** и **c** – переменные, значения которых будут вводиться. Вводимые значения должны разделяться пробелами, а ввод завершается нажатием клавиши <Enter>. Поточковый ввод и его операции автоматически распознают переменные и данные любого типа.

Консольный (на экран дисплея) вывод данных производится при помощи оператора **cout**. В C++ этот оператор называется также потоком вывода. Например, для вывода значений трех переменных надо записать:

```
cout<<a<<b<<c;
```

где << - символ операции вставки данных в поток; **a**, **b** и **c** – переменные, значения которых будут выводиться на экран. Поточковый вывод и его операции автоматически распознают переменные и данные любого типа. Вывод в Win32 производится в командную строку окна DOS. Помимо данных можно выводить и текстовую строку, заключив ее в кавычки:

```
cout<<"Summa a+b+c = "<<d;
```

Стандартные функции ввода и вывода находятся в библиотечном файле **iostream**. Чтобы связать программу разработчика со стандартной библиотекой ввода-вывода, необходимо в начале программы указать оператор подключения:

```
#include <iostream> .
```

Точка с запятой после операторов **include** не ставится.

Оператор **cout** часто используется с различными опциями (управляющими последовательностями) для расширения его функций.

## ПРАКТИЧЕСКАЯ ЧАСТЬ

На диске d:\ создайте папку с названием и номером группы, в которой создайте папку с Вашей фамилией, а в ней – папку с номером лабораторной работы. Например, **d:\2АЕз\Иванов\Кр1**. Выполнять эти и последующие действия с файлами и папками необходимо в файловом менеджере **Unreal Commander**.

**Задание 1.** Создайте проект **Кр1** для разработки программы, вычисляющей сумму трех целых чисел с использованием операторов консольных ввода и вывода данных. Проанализируйте код (текст) программы с комментариями, приведенный ниже.

Для этого:

1. Самостоятельно разработайте блок-схему алгоритма решения данного задания и аккуратно начертите в отчете (за основу возьмите схему на рис. 1).

2. Запустите Visual C++ 2010 (**Пуск/Программы/Microsoft Visual Studio 2010** или ярлык **Microsoft Visual Studio 2010** на рабочем столе).

3. При создании нового проекта в Visual C++ 2010 необходимо выполнить команды основного меню **Файл→Создать→Проект**. В левой части появившегося диалогового окна выбрать установленный шаблон **Visual C++→Win32** и далее тип проекта – **Консольное приложение Win32**.

4. Ввести имя проекта **Кр1** в текстовое поле **Имя**, удалив перед этим надпись **<Введите имя>**.

5. Нажмите кнопку **Обзор**, выберите папку **d:\2АЕз\Иванов\Кр1** для размещения создаваемого проекта и нажмите **Ок**.

6. В открывшемся диалоговом окне **Мастер приложений Win32** выберите пункт **Параметры приложения** (слева) и тип приложения **Консольное приложение**. После этого нажмите на кнопку **Готово**.

7. Создать файл программного кода **Кр1.cpp**. Для этого в открывшемся окне **Кр1 - Microsoft Visual Studio** в центральной части с заголовком **Кр1.cpp** (окно редактора программного кода) ввести в уже созданную заготовку код (текст) программы (не нужно дублировать уже имеющиеся строки!).

```
#include "stdafx.h"
#include <conio.h> //Обеспечивает задержку окна DOS на экране
#include <iostream>
using namespace std;

int _tmain(int argc, _TCHAR* argv[]) //Объявление главной функции
{ //Начало главной функции
    int a, b; //Объявление переменных целого типа
    int c=2, d=3; //Объявление переменных целого типа и их инициализация
    cout<<"vvedite b"<<endl; //Вывод на экран надписи-приглашения vvedite b . Здесь
    //endl – признак перемещения курсора на новую строку
    cin>>b; // Ввод значения переменной b
    a=b+c+d; //Вычисление суммы переменных b, c и d
    cout<<"b+c+d = "<<a; //Вывод на экран значения a - суммы переменных b, c и d
    getch(); //Функция задержки окна DOS на экране
    return 0;
} // Конец главной функции
```

**Внимание!!** Студентам необходимо разобраться с действием и назначением каждой строки программного кода!!

8. После ввода программного кода в окно редактора тщательно проверьте введенный код на отсутствие ошибок, а затем откройте пункт меню **Построение** и щелкните левой клавишей мыши (ЛК) на команде **Построить решение**. **Visual C++ 2010** проведет анализ Вашей программы с выводом результатов этого анализа в окно **Вывод**. В последней строке выведенных результатов анализа будет указано, есть ли в программе ошибки.

9. Просмотрите текст в окне **Выводы**, где будут указаны характер ошибки и номер строки ее местонахождения. После устранения ошибок в программном коде снова выполните п. 7.

10. Если ошибки отсутствуют, откройте пункт меню **Отладка** и выполните команду **Начать отладку**.

11. В открывшемся окне DOS прочитайте инструкции, введите необходимые данные (подтверждая ввод данных нажатием клавиши <Enter>) и изучите полученные результаты. Результат выполнения **Задания 1** (проект **Кр1**) имеет следующий вид:

```
vvedite b
10
b+c+d = 15
```

12. Создайте в текстовом процессоре **Word** файл **Результат Фамилия Кр1**. Поля документа сделайте по 0,5 см.

13. Поместите окно DOS с результатами решения **Задания 1** в центральной части окна **Microsoft Visual Studio** ниже программного кода **Кр1.cpp** (см. рис. 2) и нажмите клавишу <Prt Scr>, после чего вставьте полученную копию экрана в файл **Результат Фамилия Кр1**. Над вставленным рисунком проставьте номер задания – **1**. Файл результатов не закрывайте до получения оценки за выполненную практическую часть работы в тетрадь с отчетом.

14. Закройте окно DOS.

15. Откройте пункт меню **Файл** и выполните команду **Закрывать решение**.

**Внимание!** Следующие задания данной и других лабораторных работ выполняйте строго в соответствии с приведенным выше порядком действий!

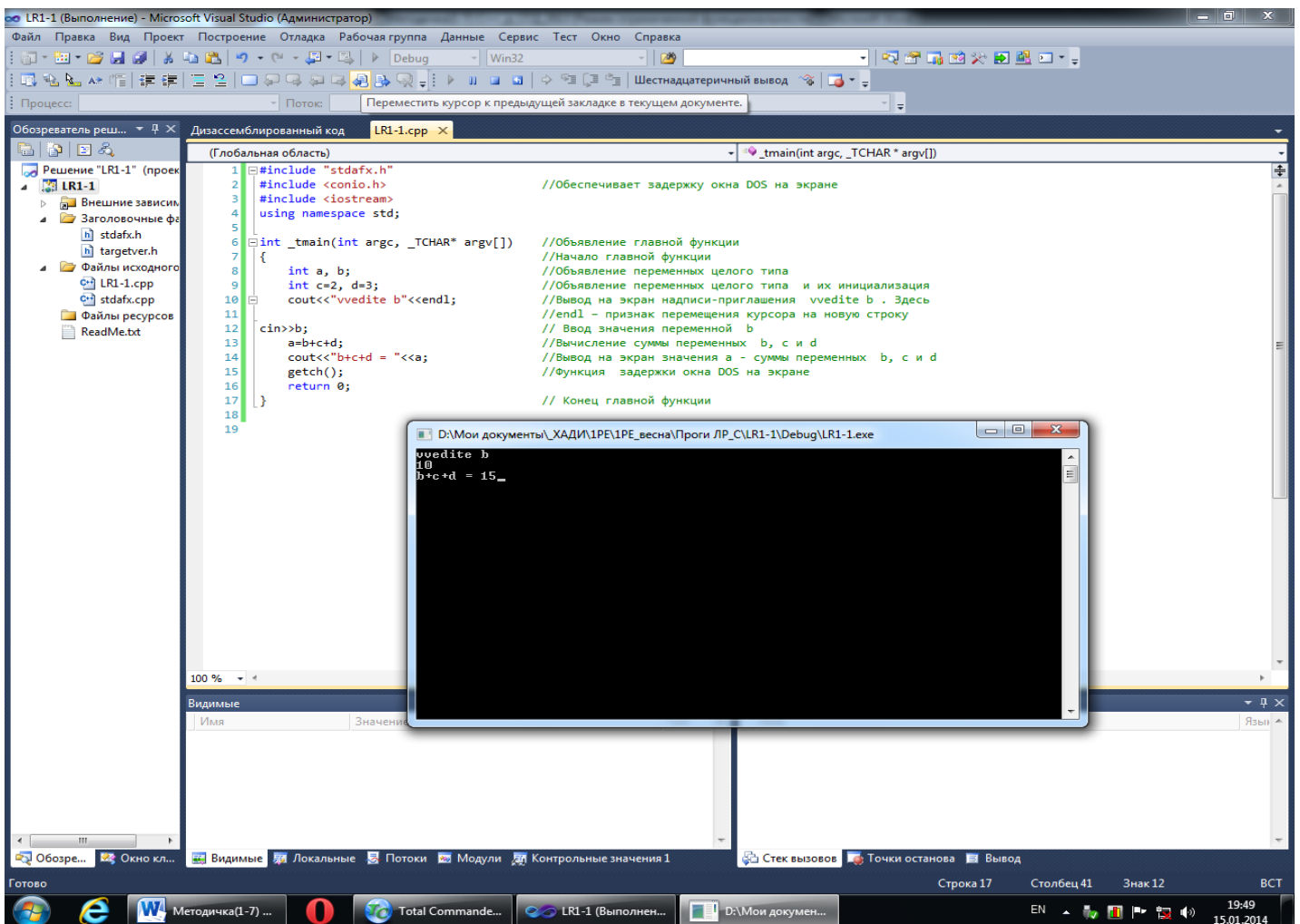


Рис. 2. Результаты выполнения **Задания 1**. Копия экрана с результатами выполнения проекта **Кр1-1**.

**Задание 2.** Создайте проект **Кр2** для вычисления переменной **a** по формуле  $a = b \frac{c + 2d - cd}{d(5c + 4b)}$  с использованием операторов консольного ввода и вывода данных. Блок-схема алгоритма решения данного задания аналогична блок-схеме на рис. 1.1. Проанализируйте код (текст) программы с комментариями, приведенный ниже.

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

int _tmain(int argc, _TCHAR* argv[]) //Объявление главной функции
```

```

{
double a; //Начало главной функции
double b, c=2, d=3; //Объявление переменной a
cout<<"vvedite b"<<endl; //Объявление переменных b, c, d и их инициализация
//Вывод на экран надписи-приглашения vvedite b. Здесь
//endl – признак перемещения курсора на новую строку
cin>>b; //Ввод значения переменной b
a=b*(c+2*d-c*d)/(d*(5*c+4*b)); //Вычисление переменной a по заданной формуле
cout<<"a = "<<a; //Вывод на экран значения a
getch(); //Функция задержки окна DOS на экране
return 0;
}

```

Результат выполнения **Задания 2** (проект **Кр1**) имеет следующий вид:

```

vvedite b
10
a = 0.133333

```

Сохраните результаты выполнения **Задания 2** в виде копии экрана таким же образом, как и в **Задании 1**.

**Задание 3.** Самостоятельно создайте проект **Lr1-3** для вычисления площади поверхности и объема цилиндра по формулам

$$S = 2\pi r(h+r),$$

где **S** – площадь поверхности цилиндра; **r** – радиус основания цилиндра; **h** – высота цилиндра;

$$V = \pi r^2 h,$$

где **V** - объем цилиндра.

Радиус основания цилиндра равен номеру Вашего компьютера, а высота – удвоенному номеру компьютера.

Перед составлением программного кода **аккуратно (!)** начертите в отчете блок-схему алгоритма решения задачи, которая аналогична блок-схеме на рис. 1.1.

**Примечание.** Поскольку в C++ нет оператора возведения в степень, то для вычисления **r<sup>2</sup>** следует воспользоваться оператором умножения, менее трудоемким, чем функция **pow**.

Число **π** задайте как константу, а значения **r** и **h** введите с клавиатуры. Результат вычислений должен содержать необходимые пояснения. Например, для 10-го компьютера **V cilindra = 6 280**.

Здесь и далее сохраните результаты выполнения задания таким же образом, как и в **Задании 1**.

**Задание 1.4.** Создайте проект **Кр4** для вычисления переменной **Y** по формуле  $Y = x^5 \frac{\sin^4 x^3 + 2e^{3a} - \operatorname{tg}\left(\frac{1-x}{1+x}\right)}{\cos 4(x+a)}$  с

использованием операторов консольных ввода и вывода данных. Код (текст) программы с комментариями приведен ниже. Переменные **x** и **a** зададим непосредственно в программном коде. Формулу для определения переменной **Y** запишем в виде, принятом в языке C++:

$$Y = \operatorname{pow}(x,5) * (\operatorname{pow}(\sin(\operatorname{pow}(x,3)),4) + 2 * \exp(3*a) - \tan((1-x)/(1+x))) / \cos(4*(x+a)) .$$

Блок-схема алгоритма решения данного задания аналогична блок-схеме на рис. 1.1. Код (текст) программы с комментариями приведен ниже.

// **Кр4.cpp**: определяет точку входа для консольного приложения.

```

#include "stdafx.h"
#include <conio.h> // Файл conio.h обеспечивает задержку окна DOS на экране дисплея
#include <iostream> // Директива include подключает файл заголовка iostream
using namespace std; // Подключает все имена из пространства имен std
int _tmain(int argc, _TCHAR* argv[]) // Объявление главной функции
{
double Y; // Начало главной функции
double a=3, x=0.3; // Объявление переменной Y
Y=pow(x,5)*(pow(sin(pow(x,3)),4)+2*exp(3*a)-tan((1-x)/(1+x)))/cos(4*(x+a)); // Объявление переменных a, x и их инициализация
cout<<"Y = "<<Y; // Вычисление Y
getch(); // Вывод на экран значения Y
return 0; // Функция задержки окна DOS на экране
}

```

Результат вычислений: **Y = 48.8651 .**

**Задание 5.** Создайте проект для вычисления функции **Y** по заданной формуле в соответствии со своим вариантом (последняя цифра номера зачетки). Варианты заданий находятся в таблице 4. Ввод исходных данных организуйте непосредственно в программе (см. задание 4).

Перед разработкой программного кода запишите заданную формулу в отчете на языке C++.



Таблица 4 Исходные данные и формулы для расчета Y (Задание 1.5)

№ варианта	Формула для расчета Y	Значения x, a, b
1	$Y = \operatorname{tg} \frac{1-x}{1+x} + \sin^2 5x + e^{5a}$	x=0,5; a=3,5
2	$Y = \frac{\sin \frac{x+1}{4}}{\sin^2 5x + e^{3a}}$	x=0,7; a=1,5
3	$Y = \frac{\sin^3(x+a) - \cos^2(x+a)}{(x+a)^4}$	x=0,82; a=2,55
4	$Y = \frac{\operatorname{tg}^3(x+a) - \arccos^2(x+a)}{(x+a)^4}$	x=0,68; a=5,55
5	$Y = \operatorname{ctg} \frac{1-3x}{1+2x} + \cos^2 5x + e^{7a}$	x=0,35; a=4,8
6	$Y = \frac{\cos^3(x+a) - 7(x+a)}{\operatorname{tg}(x+a)^4}$	x=0,62; a=4,55
7	$Y = \frac{\cos \frac{3a+1}{4}}{\sin^3 3x + e^{4a}}$	x=0,43; a=2,6
8	$Y = \frac{\sin^3(x+a) - \cos^2(x+a)}{(x+a)^4}$	x=0,74; a=1,55
9	$Y = \frac{\operatorname{tg} \frac{4a^2+1}{4}}{\cos^3 2x + e^{2a}}$	x=0,14; a=2,55
10	$Y = \sin \frac{1-x}{1+x} + \operatorname{tg}^4 5x + e^{5a}$	x=0,34; a=4,95
11	$Y = \frac{\sin^3(x+a) - \arccos^2(x+a)}{\cos(x+a)^4}$	x=0,14; a=2,95
12	$Y = \frac{\operatorname{ctg} \frac{x^3+1}{4}}{\cos^2 5x + e^{3a}}$	x=0,75; a=1,9
13	$Y = \frac{\operatorname{ctg}^3(3x+a) - \sin^2(x+7a)}{(5x+a)^3}$	x=0,44; a=2,95
14	$Y = \frac{\arcsin^3 \frac{4x+1}{4}}{\operatorname{ctg}^2 3x + e^{3a}}$	x=0,27; a=1,9

15	$Y = \frac{\arcsin^3 \frac{4x+1}{4}}{\operatorname{ctg}^2 3x + e^{3a}}$	x=0,49; a=3,7
16	$Y = \frac{\sin^3(x+a) - \cos^2(x+a)}{(x+a)^4}$	x=0,83; a=4,7
17	$Y = \frac{\operatorname{arccctg} \frac{2x^3+1}{4}}{\cos^2 5x + e^{3a}}$	x=0,37; a=2,75
18	$Y = \frac{\operatorname{tg}^3(x+a) - 5(\sin x + a)}{\sin^3(x+a)^4}$	x=0,13; a=0,7

**Задание 6.** Создайте проект для вычисления функции Y по заданной формуле в соответствии со своим вариантом (последняя цифра номера зачетки). Варианты заданий находятся в таблице 5. Ввод исходных данных организуйте непосредственно в программе (см. задание 4). Перед разработкой программного кода запишите заданную формулу в отчете на языке C++.

**Таблица 5** Исходные данные и формулы для расчета Y (Задание 6)

№ варианта	Формула для расчета Y	Значения x, a, b
1	$Y = \operatorname{tg} \frac{1-x}{1+x} + \sin^2 5x + e^{5a}$	x=0,5; a=3,5
2	$Y = \frac{1 + \cos \frac{1-x^2}{1+x^2}}{\sqrt{1 + \left(\frac{1-a^3}{1+a^4}\right)^2}} - e^{-\frac{x^2}{2}}$	x=1,573; a=1,775
3	$Y = \ln\left(1 + \sqrt{1+x^2}\right) + \frac{1+x^2}{\sqrt{1 + \frac{a^4}{2}}}$	x=1,573; a=1,775
4	$Y = \frac{\sin^3(a+x^2)}{\sqrt{a + \frac{x^3+1}{2} + \left(\frac{x^2+1}{2}\right)^5}}$	x=1,573; a=1,775
5	$Y = \left(a + \frac{a}{a^4 + x^2}\right) - \sqrt{1 + \frac{a}{a^2 + x^2}}$	x=1,573; a=1,775
6	$Y = 2a^3 + \frac{2x^4}{\sqrt{1+a^2}} - e^{-\frac{a^2+1}{2}}$	x=1,573; a=1,775
7	$Y = \frac{e^{\frac{x^2}{2}} + e^{1+\frac{a^2}{2}}}{1 + \frac{a^2}{2} + \left(\frac{x^4}{2}\right)^2}$	x=1,573; a=1,775
8	$Y = \lg\left(1 + \sqrt{1+a^5}\right) + \frac{e^a + x^4}{\sqrt[4]{1 + \frac{x^3}{2}}}$	x=1,573; a=1,775

9	$Y = \frac{(a+x)^3 \cdot \ln \frac{a+x}{2}}{\sqrt[3]{1 + \frac{(a+x^3)^2}{4}}}$	x=1,573; a=1,775
10	$Y = 2x^3 + \frac{2a^4}{\sqrt[3]{1+x^3}} - e^{-\frac{a^2+1}{2}}$	x=1,573; a=1,775
11	$Y = \sqrt[4]{\frac{1 + \operatorname{tg}^2 \frac{x+1}{4}}{1 + \frac{a^3+1}{4}}} - e^{-\frac{x+1}{4}}$	x=1,573; a=1,775
12	$Y = \frac{e^{\frac{5x^2}{2}} + e^{\frac{1+4a^3}{2}}}{1 + \frac{x^2}{2} + \left(\frac{a^3}{2}\right)^2}$	x=1,573; a=1,775
13	$Y = \frac{\lg(1+x^4)}{\sqrt{1 + \frac{x^2+a^3}{2} + \left(\frac{x^2+1}{2}\right)^2}}$	x=1,573; a=1,775
14	$Y = \frac{1 + \left(\frac{x}{a}\right)^{2,5} + \sqrt{1 + \left(\frac{x}{a}\right)^{2,5}}}{1 + e^{x^3} + \sqrt{1 + \left(\frac{x}{a}\right)^{2,5}}}$	x=1,573; a=1,775
15	$Y = \left(x^3 + \frac{a}{a^2+x^2}\right) - \sqrt{1 + \frac{a}{a^2+x^4}}$	x=1,573; a=1,775
16	$Y = \sqrt[4]{\frac{1 + \operatorname{tg}^2 \frac{a^3+1}{4}}{1 + \frac{x+1}{4}}} - e^{-\frac{a^2+1}{4}}$	x=1,573; a=1,775
17	$Y = \frac{1 + \left(\frac{x^3}{a}\right)^{2,5} + \sqrt{1 + \left(\frac{x}{a}\right)^{2,5}}}{1 + e^x + \sqrt{1 + \left(\frac{x}{a^3}\right)^{2,5}}}$	x=1,573; a=1,775
18	$Y = \ln\left(1 + \sqrt{1+x^2}\right) + \frac{1+x^2}{\sqrt[3]{1 + \frac{a^3}{2}}}$	x=1,573; a=1,775

### 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. 3 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**.

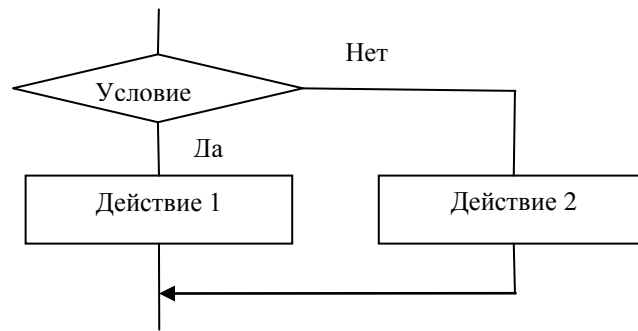


Рис. 3. Структура ветвления

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:

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

**Задание 7.** Исследовать и выполнить программу для определения переменной **a** по следующему условию:

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

Блок-схема алгоритма решения задания 1 представлена на рис. 4. Начертите ее в контрольной работе.

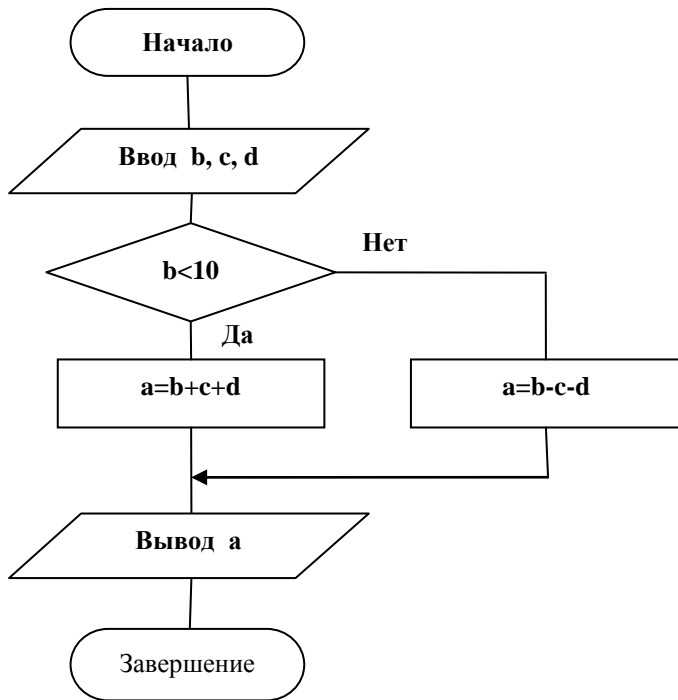


Рис. 4. Блок-схема алгоритма определения переменной **a** (простые операторы)

Введите и выполните программный код, реализующий алгоритм:

```

#include "stdafx.h"
#include <conio.h>
#include <iostream>
using namespace std;

int _tmain(int argc, _TCHAR* argv[])
{
    int a, b;
    int c=2, d=3;
    cout<<"Vvedite b"<<endl;
    cin>>b;
    if (b<10)
        a=b+c+d;
    else
        a=b-c-d;
    cout<<"a = "<<a;
    getch();
    return 0;
}
  
```

//Файл **conio.h** обеспечивает задержку окна DOS на экране дисплея  
 //Директива **include** подключает файл ввода–вывода **iostream**  
 //Подключает все имена из пространства имен **std**  
 //Объявление главной функции **\_tmain**  
 //Начало главной функции  
 //Объявление переменных целого типа  
 //Объявление переменных целого типа и их инициализация  
 //Вывод на экран надписи-приглашения **Vvedite b**  
 //Ввод значения переменной **b**  
 //Условный оператор **if ...else** (1-я часть)  
 //Вычисление переменной **a** при выполнении условия  
 //Условный оператор **if ...else** (2-я часть)  
 //Вычисление переменной **a** при невыполнении условия  
 //Вывод на экран значения переменной **a**  
 //Функция задержки окна DOS на экране

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

Создайте в текстовом процессоре **Word** файл **Результат Фамилия Кр7**. Поля документа сделайте по 0,5 см.

**Задание 8.** 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. 5. Обратите внимание на отличия данной блок-схемы от схемы, приведенной на рис. 5. Объясните их. Начертите блок-схему в отчете по лабораторной работе.

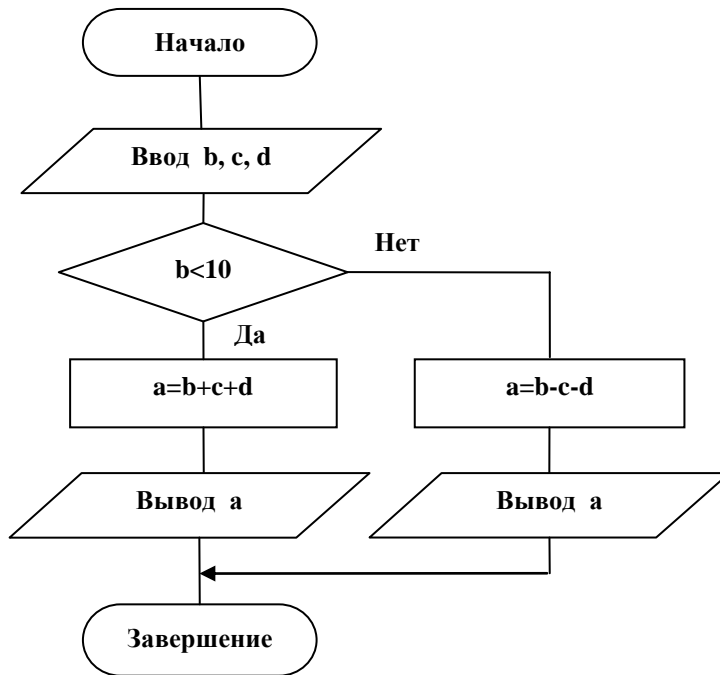


Рис. 5. Блок-схема алгоритма определения переменной **a** (сложные операторы)

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>
#include <iostream>
using namespace std;

int _tmain(int argc, _TCHAR* argv[])
{
    int a, b;
    int c=2, d=3;
    cout<<"Vvedite b"<<endl;
    cin>>b;
    if (b<10)
    {
        a=b+c+d;
        cout<<"a=b+c+d= "<<a;
    }
    else
    {
        a=b-c-d;
        cout<<"a=b-c-d= "<<a;
    }
    getch();
    return 0;
}
  
```

//Файл **conio.h** обеспечивает задержку окна DOS на экране дисплея  
 //Директива **include** подключает файл ввода–вывода **iostream**  
 //Подключает все имена из пространства имен **std**  
 //Объявление главной функции **\_tmain**  
 //Начало главной функции  
 //Объявление переменных целого типа  
 //Объявление переменных целого типа и их инициализация  
 //Вывод на экран надписи-приглашения **Vvedite b**  
 //Ввод значения переменной **b**  
 //Условный оператор **if...else** (1-я часть)  
 //Начало сложного оператора  
 //Вычисление переменной **a** при выполнении условия  
 //Вывод на экран значения переменной **a**  
 //Конец сложного оператора  
 //Условный оператор **if...else** (2-я часть)  
 //Начало сложного оператора  
 //Вычисление переменной **a** при невыполнении условия  
 //Вывод на экран значения переменной **a**  
 //Конец сложного оператора  
 //Функция задержки окна DOS на экране

**Задание 9.** Самостоятельно разработайте и выполните программу для определения переменной **Y** по следующему условию:

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

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. 3. Чертить ее необязательно. Результаты сохраните.

**Задание 9.** Самостоятельно разработать алгоритм и программу для вычисления переменной  $Y$  (использовать оператор **if...else** и консольный ввод-вывод переменных). Определите свой номер варианта как последняя цифра номера зачетки.

**Таблица 6** Исходные данные и формулы для расчета  $Y$  (Задание 9)

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

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

## 2. Использование условного оператора if...else для 3-х интервалов значений переменных

Выше исследовались случаи применения условного оператора **if...else** для 2-х интервалов значений переменных, т. е. рассматривались простые условия (логические выражения) типа  $x < a$ . При этом число  $a$  делит числовую ось  $x$  на два интервала (рис. 6).

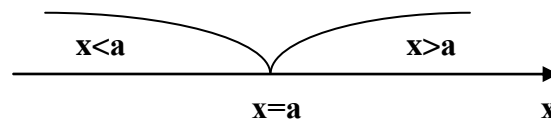


Рис. 6. Графическая интерпретация логического выражения для двух интервалов значений  $x$

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. 7). В этом случае условие запишется следующим образом:

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

где **&&** - операция логического И.



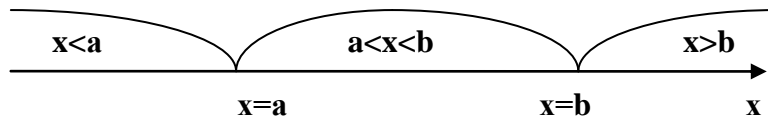


Рис. 7. Графическая интерпретация логического выражения для трех интервалов значений  $x$

### 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. 7 приведен пример структуры вложенного условного оператора `if...else` (в качестве базовой использована структура, изображенная на рис. 8).

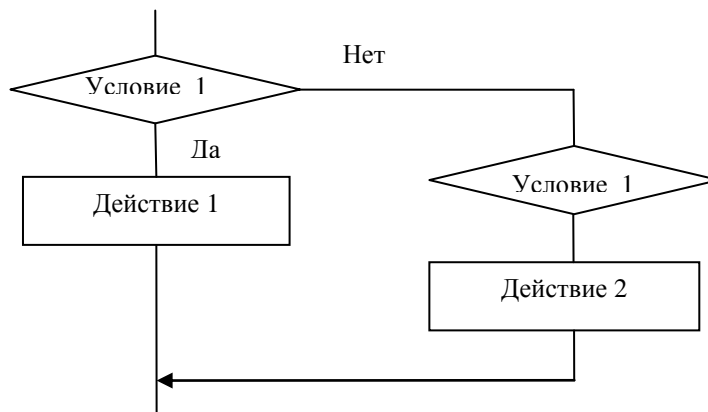


Рис. 8. Структура вложенного условного оператора `if...else`

### 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-схема алгоритма, реализующего работу оператора выбора `switch`, приведена на рис. 9.

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).

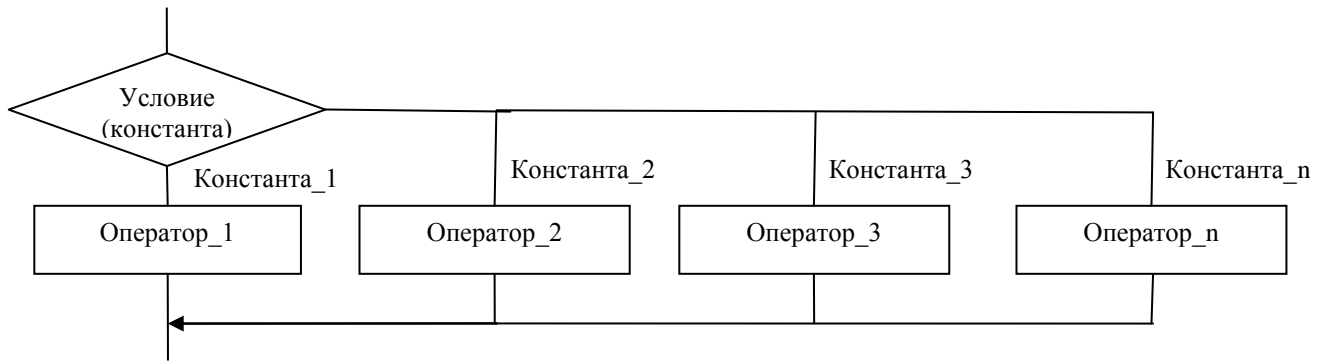


Рис. 9. Блок-схема алгоритма работы оператора **switch**

**Задание 10.** 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, содержащимся 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>
#include <iostream>
using namespace std;

int _tmain(int argc, _TCHAR* argv[])
{
    double a, b, res;
    char operation;
    cout << "Vvedite a : "<<endl;
    cin >> a;
    cout << "Vvedite operation : "<<endl;
    cin >> operation;
    cout << "Vvedite b : "<<endl;
    cin >> b;
    switch (operation)
    {
        case '+': res = a + b; break;
        case '-': res = a - b; break;
        case '*': res = a * b; break;
        case '/': res = a / b; break;
        default : cout <<"Unknown operation"<<endl;
    }
    cout << "Result : " << res;
    getch();
    return 0;
}

```

//Файл **conio.h** обеспечивает задержку окна DOS на экране дисплея  
//Директива **include** подключает файл ввода-вывода **iostream**  
//Подключает все имена из пространства имен **std**  
//Объявление главной функции **\_tmain**  
//Начало главной функции  
//Объявление вещественных переменных  
//Объявление символьной переменной  
//Вывод на экран надписи-приглашения **Vvedite a :**  
//Ввод значения переменной **a**  
//Вывод на экран надписи-приглашения **Vvedite operation :**  
//Ввод знака операции  
//Вывод на экран надписи-приглашения **Vvedite b :**  
//Ввод значения переменной **b**  
//Условие (с чем будет сравнение) в операторе **switch**  
//Начало составного оператора  
//Метка **"+"** и вычисление переменной **res** для этой метки  
//Метка **"-"** и вычисление переменной **res** для этой метки  
//Метка **"\*"** и вычисление переменной **res** для этой метки  
//Метка **"/"** и вычисление переменной **res** для этой метки  
//Вывод сообщения о неизвестной операции  
//Конец сложного оператора  
//Вывод на экран результата вычислений  
//Функция задержки окна DOS на экране  
//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 переход с 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 **A**.

**Задание 11.** 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>
#include <iostream>
using namespace std;

int _tmain(int argc, _TCHAR* argv[])
{
    int a, b;
    int c=2, d=3;
    cout<<"Vvedite b"<<endl;
    cin>>b;
    goto A;
    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 ";
    getch();
    return 0;
}

```

//Файл **conio.h** обеспечивает задержку окна DOS на экране дисплея  
//Директива **include** подключает файл ввода-вывода **iostream**  
//Подключает все имена из пространства имен **std**

//Объявление главной функции **\_tmain**  
//Начало главной функции  
//Объявление переменных целого типа  
//Объявление переменных целого типа и их инициализация  
//Вывод на экран надписи-приглашения **Vvedite b**  
//Ввод значения переменной **b**  
//Оператор **goto**

//Метка **A** и оператор вывода  
//Функция задержки окна DOS на экране

**Задание 12.** Самостоятельно разработать алгоритм и программу решения задачи (использовать оператор **if...else**). Определите свой номер варианта как последняя цифра номера зачетки.

#### 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 пополнить 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 пополнить 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 пополнить 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 для письменного ответа**

1. Что такое проект в программе Visual Studio 2010?
2. Перечислите основные или базовые типы данных в языке C++.
3. Раскройте понятие переменной в языке C++.
4. Раскройте понятие выражения в языке C++.
5. Как записываются математические действия с переменными, константами и функциями в языке C++?
6. Каковы правила обращений к математическим функциям в языке C++?
7. Чем отличаются программы с линейной структурой?
8. Как вывести на экран значение переменной?
9. Как в одном операторе объявить тип переменной и задать ее значение?
10. Перечислите опции оператора **cout** и раскройте их действие.
11. Какие функции выполняет оператор **include**?
12. Какие функции выполняет файл заголовка **io stream**?
13. Какие операторы языка C++ используются для реализации разветвляющихся вычислительных процессов?
14. В каких случаях необходимо применение условных операторов?
15. Запишите условный оператор **if...else** в общем виде и опишите порядок его выполнения.
16. Чем отличается блок операторов от составного оператора?
17. Запишите в общем виде оператор **switch** и опишите порядок его выполнения.
18. Для чего необходим оператор **break**?
19. В каких случаях необходимо применение оператора **goto**?
20. Запишите в общем виде оператор **goto** и опишите порядок его выполнения