

Л. Н. ТРОФИМОВА, А. В. ЕРОШЕНКО

ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ C++

ЧАСТЬ 1

ОМСК 2017

Министерство транспорта Российской Федерации
Федеральное агентство железнодорожного транспорта
Омский государственный университет путей сообщения

Л. Н. Трофимова, А. В. Ерошенко

ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ C++

Часть 1

Утверждено методическим советом университета
в качестве учебно-методического пособия к выполнению лабораторных работ и
самостоятельной работы по дисциплине «Информатика»

Омск 2017

УДК 004.432(075.8)
ББК 22.183.492я73
Т76

Основы программирования на языке С++: Учебно-методическое пособие к выполнению лабораторных работ и самостоятельной работы по дисциплине «Информатика». Часть 1/ Л. Н. Трофимова, А. В. Ерошенко; Омский гос. ун-т путей сообщения. Омск, 2017. 40 с.

В учебно-методическом пособии представлены основы программирования на языке С++. Рассматриваются этапы ввода, вывода данных, операции языка, порядок разработки линейных и разветвляющихся алгоритмов. Даны практические рекомендации для решения задач.

Предназначено для студентов первого курса Института автоматизации, телекоммуникаций и информационных технологий всех специальностей очной и заочной форм обучения.

Библиогр.: 2 назв. Табл. 12. Рис. 7.

Рецензенты: канд. физ.-мат. наук, доцент А. А. Романова;
канд. техн. наук, доцент А. Г. Малютин.

ОГЛАВЛЕНИЕ

Введение	5
Лабораторная работа 1. Принципы создания программ на языке С++	6
1.1. Организация работы пользователя	6
1.2. Структура программы	6
1.3. Исполнение программы	8
1.4. Задания	9
Лабораторная работа 2. Запись арифметических выражений на языке С++. Ввод-вывод данных. Операции языка	9
2.1. Алфавит	9
2.2. Переменные. Константы. Служебные слова	10
2.3. Ввод-вывод данных	11
2.4. Операции языка С++	12
2.4.1. Операция присваивания	13
2.4.2. Операции сравнения	13
2.4.3. Арифметические операции	14
2.4.4. Логические операции	16
2.5. Комментарии. Специальные символы	17
2.6. Математические функции	17
2.7. Задания	19
Лабораторная работа 3. Алгоритмы линейных структур	22
3.1. Понятие алгоритма	22
3.2. Алгоритмы линейных структур	23
3.3. Задания	24
Лабораторная работа 4. Алгоритмы разветвляющихся структур	28
4.1. Условный оператор перехода	28
4.2. Операция-условие	30
4.3. Оператор-переключатель	31
4.4. Задания	33
Библиографический список	39

ВВЕДЕНИЕ

Концепция объектно-ориентированного программирования и визуального подхода к построению приложений нашла наиболее полное отражение в языках для разработки Windows приложений *Visual Basic*, *MS Visual C++*. Общим для них является простота и наглядность процесса создания программ, основанных на использовании технологий визуального программирования [1].

Visual C++ представляет собой объектно-ориентированный язык программирования. Объектная ориентированность C++ означает, что он поддерживает стиль программирования, упрощающий кодирование крупномасштабных программ. Возможность программирования появилась в C++ на основе использования логических конструкций из слов и предложений. Эти программные инструкции переводятся на машинный язык *интерпретаторами* и *компиляторами*. *Интерпретатор* превращает инструкции программы в команды машинного кода последовательно, по мере чтения текста программы. *Компилятор* преобразует исходный код программы в некоторую промежуточную форму – объектный файл. Этот этап называется *компиляцией*. Затем компилятор вызывает программу компоновки, которая создает исполняемый файл программы. Большинство современных программ, работающих на персональных компьютерах, написаны на C++. Современные тенденции показывают, что *Visual Basic*, *MS Visual C++* используются для создания приложений, разработки интернет-обозревателей и операционных систем [2].

Данное учебно-методическое пособие предназначено для студентов 1-го курса, изучающих программирование, и содержит материал к выполнению лабораторных работ по дисциплине «Информатика», а также для самостоятельного изучения языка программирования *MS Visual C++*.

Цель пособия – познакомить студентов со структурой, принципами и основными концепциями языка *MS Visual C++*.

В каждой лабораторной работе приведены необходимые теоретические сведения, примеры, иллюстрирующие описанные приемы работы, задания для самостоятельного выполнения. Владения основами информатики достаточно для изучения представленного материала.

Лабораторная работа 1

ПРИНЦИПЫ СОЗДАНИЯ ПРОГРАММ НА ЯЗЫКЕ C++

Цель работы: изучение основных принципов создания программы на языке C++.

1.1. Организация работы пользователя

Запуск среды программирования *Visual C ++* может быть осуществлен разными способами. Рассмотрим два из них.

1. Двойным щелчком левой кнопки мыши (ЛКМ) по значку (ярлыку) на рабочем столе.

2. С помощью команды *Пуск – Все программы – Visual Studio 2013 – Visual Studio 2013*.

После запуска среды в диалоговом окне по команде меню *Файл* во вкладке *Создать* выбрать *Проект*. В раскрывшемся окне *Создать проект* (рис. 1.1) выбрать шаблон *Visual C++* и *Консольное приложение Win 32* – это наиболее простое приложение, в котором видна вся структура программы.

Затем указать имя создаваемого проекта в поле *Имя* и размещение проекта на диске в поле *Расположение*. После определения имени и пути нажать кнопку *ОК*. Созданный проект располагается в месте, указанном при его создании. Он содержит папку *Debug*, в которую записывается информация об изменениях и отладке программного кода, а также основной файл проекта *Имя_проекта.cpp* и файлы, генерируемые системой.

1.2. Структура программы

Программы на языке C++ состоят из объектов, функций, переменных и других элементов. В числе прочих элементов находятся папки, которые содержат файлы с фрагментами программного кода – *заголовочные* файлы. Эти файлы объединяются в библиотеки шаблонов.

Включение в программный код содержимого заголовочного файла выполняется после обработки компилятором директивы *#include*. Первым стоит символ *#*, который является инструкцией для *препроцессора*. *Препроцессор* – это текстовый процессор, обрабатывающий исходный файл до его поступления на вход компилятора.

При каждом запуске компилятора запускается и препроцессор. Он читает исходный текст программы, находит строки, которые начинаются с символа `#`, и обрабатывает их перед началом компиляции программы. Команда *include* (подключить) – это инструкция препроцессора, которая указывает, какой файл следует подключить. Подключенный в результате этого действия файл выполняется так, как если бы он был написан непосредственно в программном коде. Например, строка `#include<iostream>`. В этой строке содержится директива препроцессору о подключении библиотеки ввода-вывода *iostream*. Кроме заголовочных файлов программный код на языке C++ состоит из одной или более функций. *Функция* – коллективное (общее) имя для группы описаний, заключенных в фигурные скобки `{}`. Фигурные (операторные) скобки ограничивают последовательность действий. Любая программа на языке C++ состоит из одной и более функций, одна из которых (и только одна) должна иметь имя *tmain*, и именно ей передается управление из операционной системы.

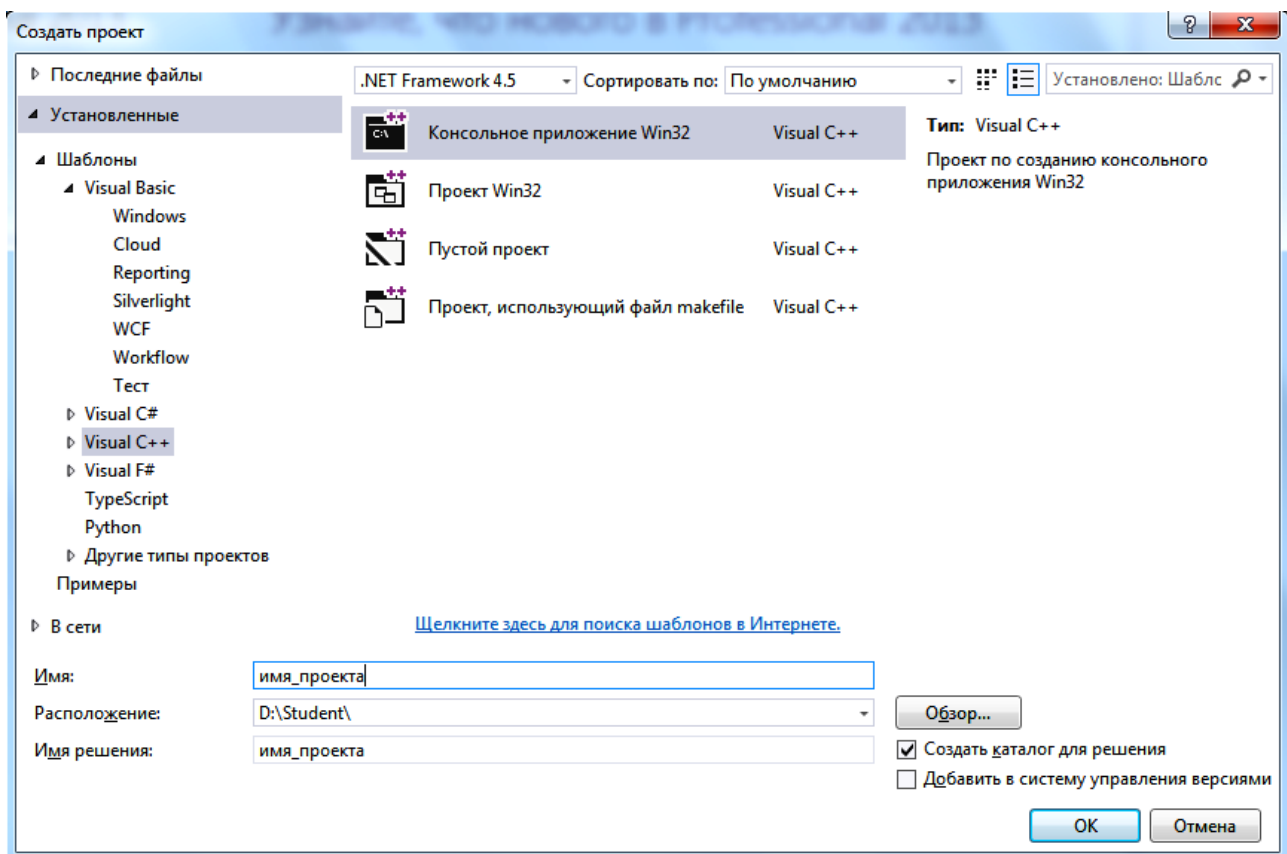


Рис. 1.1. Диалоговое окно *Создать проект*

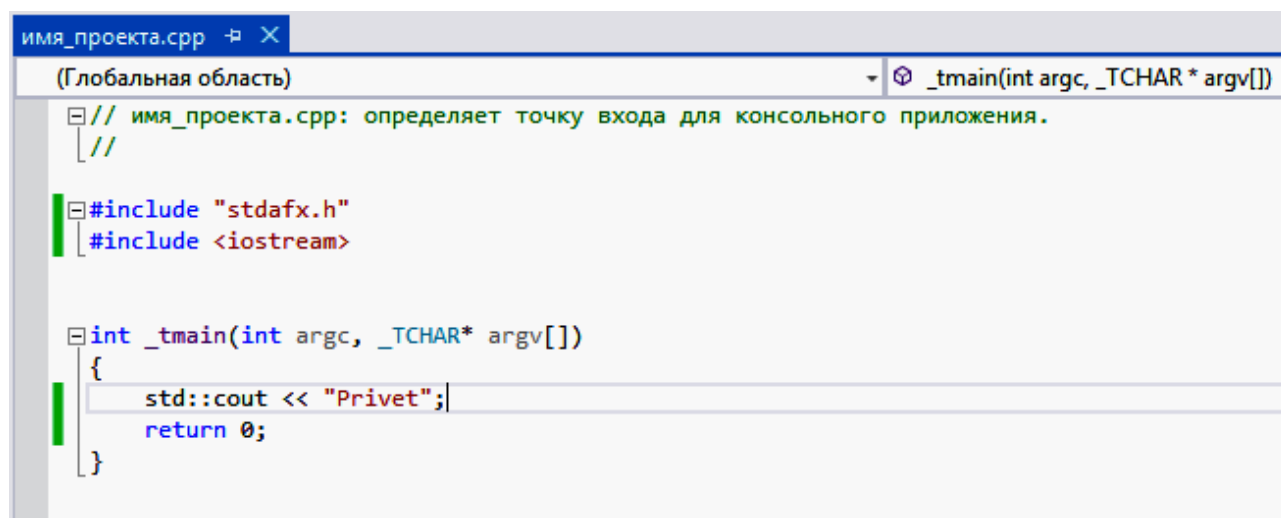
Функция `tmain(int argc, _TCHAR* argv[])`, для которой операторные скобки `{}` определяют область всего программного кода, содержит такие аргументы:

`int argc` – счетчик количества аргументов, передаваемых компилятором для запуска программы;

`_TCHAR * argv[]` – массив указателей, каждый из которых ссылается на один из аргументов.

В операторных скобках последней строкой указывается оператор *return*. Оператор *return* указывает на результат, возвращаемый программой операционной системе.

Код простейшей программы на языке C++ представлен на рис. 1.2.



```
имя_проекта.cpp  X
(Глобальная область)  _tmain(int argc, _TCHAR* argv[])
// имя_проекта.cpp: определяет точку входа для консольного приложения.
//
#include "stdafx.h"
#include <iostream>

int _tmain(int argc, _TCHAR* argv[])
{
    std::cout << "Privet";
    return 0;
}
```

Рис 1.2. Пример программы

1.3. Исполнение программы

После окончания записи программного кода выполняется компиляция программы без исполнения. Для этого выбирается меню *Сборка – Только проект – Собрать только имя проекта*. При этом для созданного файла *Имя_проекта.cpp* компилятор проверяет пути расположения необходимых заголовочных файлов, наличие синтаксических ошибок в теле программного кода. В нижней части окна среды *Visual C++* перечисляются результаты работы компилятора: указывается номер строки программного кода, в котором возникла ошибка, и описание ошибки. Щелчок мыши по строке ошибки ставит курсор в место ошибки текста. Компилятор создает объектный файл. Далее объектный код обрабатывается специальной программой – редактором связей, который выполняет связывание объектных модулей и машинного кода стандартных функций, находящихся в библиотеке транслятора. На этом этапе формируется работоспособное приложение – исполнимый файл, или исполнимый код с рас-

ширением *.exe*, для его создания используется режим *Сборка*. Запуск исполняемого файла и получение результата происходит при выборе меню *Отладка – Запуск без отладки*. Для завершения работы программы можно нажать любую клавишу.

1.4. Задания

Задание 1. Создайте новый проект. Назовите проект PROBA_1.

Задание 2. Добавьте заголовочный файл `#include <iostream>`.

Задание 3. Запишите в тело программы следующую строку:

```
std::cout<<"PRIVET";
```

Задание 4. Откомпилируйте программный код.

Задание 5. Запустите программу. Посмотрите результат.

Лабораторная работа 2

ЗАПИСЬ АРИФМЕТИЧЕСКИХ ВЫРАЖЕНИЙ НА ЯЗЫКЕ C++.

ВВОД-ВЫВОД ДАННЫХ. ОПЕРАЦИИ ЯЗЫКА

Цель работы: изучение правильного написания арифметических выражений на языке C++.

2.1. Алфавит

Программы составляются и записываются по определенным правилам с помощью *алфавита* языка – разрешенного к использованию набора символов, с помощью которого могут быть образованы слова и величины данного языка.

Алфавит языка C++ включает в себя: прописные и строчные буквы латинского алфавита; десятичные цифры от 0 до 9; специальные символы: " {}, | [] () + - / \ ; ' : ? < = > _ ! & # ~ ^ . * ; разделители (знаки пунктуации).

Идентификатор – последовательность из букв латинского алфавита, десятичных цифр и символов подчеркивания, начинающихся не с цифры. Идентификаторы используются для обозначения имен переменных, функций, типов данных. Один идентификатор может состоять не более чем из 32 символов.

2.2. Переменные. Константы. Служебные слова

Переменная – это величина, значение которой может изменяться в ходе выполнения программы. Переменная характеризуется типом, именем и значением.

Имя переменной (идентификатор) состоит из комбинации букв, цифр. На первом месте должна быть буква, в имени переменной допускается использовать знак подчеркивания.

Пример:

double a1. Переменной присвоено имя *a1*, тип – вещественный.

Основные типы переменных и их размер приведены в табл. 2.1.

Т а б л и ц а 2.1

Основные типы переменных

Тип	Название	Размер	Тип	Название	Размер
int	Целый	2 байта	long int	Длинный целый	4 байта
float	Вещественный	4 байта	short int	Короткий целый	2 байта
double	Вещественный двойной точности	8 байтов	signed short int	Знаковый короткий целый	2 байта
char	Символьный	1 байт	unsigned long int	Беззнаковый длинный целый	2 байта

Язык C++ различает регистр букв. Прописные и строчные буквы считаются разными. Переменные с именами *age*, *Age* и *AGE* рассматриваются как три различные переменные. Если для имени переменной необходимы два слова (например, *my car*), то в соответствии с наиболее популярными соглашениями возможны два варианта: *my_car* или *myCar*.

При написании программы на языке C++ допускается создавать несколько переменных с помощью одного оператора, указав их тип и перечень имен, разделенных запятыми, например: *int x, y, z;*

Некоторые слова изначально зарезервированы в языке C++, и поэтому их нельзя использовать в качестве имен переменных. Такие слова называются *служебными (ключевыми)*, например: *if*, *for*, *cout*.

Константа – это величина, значение которой не изменяется в ходе выполнения всей программы, например: 123; 45.6; 2E-5.

2.3. Ввод-вывод данных

Для ввода данных с клавиатуры используют входной поток *cin*, после которого следует знак экстрактора (оператор извлечения) *>>*, имя переменной, значение которой необходимо ввести с клавиатуры, в конце ставится точка с запятой.

Пример:

std::cin>>x; – данная запись означает ввод с клавиатуры значения переменной *x*.

Для ввода с клавиатуры значений нескольких переменных используют следующую конструкцию: *std::cin>>x>>y;*

Для вывода данных применяют выходной поток *cout*, после которого следует оператор вывода *<<*, имя переменной, значение которой необходимо вывести с клавиатуры, в конце ставится точка с запятой.

Пример:

std::cout<<x; – данная запись означает вывод значения переменной *x* на экран монитора.

Аналогично вводу записывается вывод нескольких переменных.

Для того чтобы перед каждым применением операторов *cin*, *cout* не писать идентификатор *std::*, существуют два способа: в начале текста кода сообщают компилятору об использовании объектов *cin*, *cout* только из стандартной библиотеки, для чего используют ключевое слово *using* (рис. 2.1); указывают компилятору на то, что будет использовано стандартное пространство имен *using namespace std* (рис. 2.2)

Компилятор C++ распознает некоторые специальные символы, предназначенные для формирования текста. Чтобы вставить эти символы в программу, необходимо ввести одиночную кавычку, обратную косую черту, указывающую на то, что следующий за ней символ является управляющим, и снова одиночную кавычку (например, *'\t'* – символ табуляции; *'\n'* – символ перевода на новую строку).

Пример:

```
int x = 4, y = 5;
```

```
std::cout<<x<<' '<<y;
```

На экране увидим: 4 5

```
//  
  
#include "stdafx.h"  
#include<iostream>  
  
int _tmain(int argc, _TCHAR* argv[])  
{  
    using std::cin;  
    using std::cout;  
    int x,y;  
    cout<<"Vvedite chisla: ";  
    cin>>x;  
    cin>>y;  
    cout<<x+y;  
    return 0;  
}
```

Рис. 2.1. Фрагмент кода программы
с использованием ключевого слова *using*

```
//  
  
#include "stdafx.h"  
#include<iostream>  
  
int _tmain(int argc, _TCHAR* argv[])  
{  
    using namespace std;  
    int x,y;  
    cout<<"Vvedite chisla\n";  
    cin>>x;  
    cin>>y;  
    cout<<x+y;  
    return 0;  
}
```

Рис. 2.2. Фрагмент кода программы
с использованием ключевого слова *namespace*

2.4. Операции языка C++

Переменные в языке C++ необходимы не только для хранения данных, над переменными можно выполнять различные операции: арифметические, логические, сравнения, сдвига. Знаки операций обеспечивают формирование

и последующее вычисление выражений. Один и тот же знак операции может употребляться в нескольких выражениях и по-разному интерпретироваться в зависимости от контекста.

2.4.1. Операция присваивания

Оператор присваивания ($=$) служит для вычисления значения выражения и присваивания этого значения переменной. Так, выражение $sum = a + b$; присваивает переменной sum значение результата сложения a и b . Оператор присваивания используется также для присваивания переменной значения (инициализации переменной).

Пр и м е р:

```
int x;
```

```
x = 5;
```

Эти две строки можно объединить в одну и инициализировать переменную x в процессе ее определения:

```
int x = 5;
```

В C++ предусмотрена сокращенная запись операции присваивания. Если слева и справа от оператора присваивания стоит одна и та же переменная, то запись операции присваивания можно сократить, например, вместо $s = s + i$ записывают $s += i$; вместо $a = a / b$ используют запись $a /= b$.

2.4.2. Операции сравнения

Операторы отношения используются для определения равенства или неравенства двух значений. Результатом выполнения операции сравнения является 0 (ложь) или 1 (истина). Операторы отношения представлены в табл. 2.2.

Таблица 2.2

Операторы отношения

Имя	Оператор	Пример	Значение
1	2	3	4
Равно	$==$	$100 == 50$ $50 == 50$	Ложь Истина

1	2	3	4
Не равно	<code>!=</code>	100 <code>!=</code> 50 50 <code>!=</code> 50	Истина Ложь
Больше	<code>></code>	100 <code>></code> 50 50 <code>></code> 50	Истина Ложь
Больше или равно	<code>>=</code>	100 <code>>=</code> 50 50 <code>>=</code> 50	Истина Истина
Меньше	<code><</code>	100 <code><</code> 50 50 <code><</code> 50	Ложь Ложь
Меньше или равно	<code><=</code>	100 <code><=</code> 50 50 <code><=</code> 50	Ложь Истина

2.4.3. Арифметические операции

Над переменными можно выполнять различные арифметические операции. Стандартные арифметические операции – умножение, деление, сложение, вычитание – выполняются аналогично этим операциям в математике, однако результат выполнения операций будет зависеть от типа объявленных переменных. Так, если делимое и делитель объявлены как целые величины, то результат также будет целой величиной.

Пример:

```
int v = 7, x = 2;
```

```
int p = v / x;
```

```
std::cout << p;
```

В результате на экране появится: 3.

Для получения вещественного результата необходимо, чтобы хотя бы одна из величин (делимое или делитель) и обязательно частное были вещественной величиной.

Пример:

```
double v = 7;
```

```
int x = 2;
```

```
double p = v / x;
```

```
std::cout << p;
```

В результате на экране появится: 3.5.

В языке C++ существуют специальные операции. Приоритет выполнения и описание операторов приведены в табл. 2.3.

Таблица 2.3

Приоритет специальных операций

Приоритет	Оператор	Описание	Пример	Значение
1	$-A$	Возвращает противоположное по знаку, равное по модулю значение	<pre>int v = 6; int x = -v; cout << x;</pre>	- 6
2	$++$	Оператор инкремента. Увеличивает значение аргумента на единицу	<pre>int v = 6; v ++; cout << v;</pre>	7
2	$--$	Оператор декремента. Уменьшает значение аргумента на единицу	<pre>int v = 6; v --; cout << v;</pre>	5
3	$\%$	Возвращает остаток от целочисленного деления	<pre>int v = 16; int x = 7; int p = v%x; cout << p;</pre>	2

Общий приоритет операций определяется скобками.

Возведение в степень в C++ выполняется функцией *pow*, она записывается следующим образом:

pow(*основание степени x*, *показатель степени y*) – на языке математики это означает x^y .

Операции инкремента и декремента имеют префиксную ($++n$; $--n$) или постфиксную ($n++$; $n--$) запись. В первом случае значение операнда *n* сначала изменяется, а затем применяется для дальнейших вычислений.

Пример:

```
int x, y;
```

```
x = 5;
```

```
y = ++x;
```

```
std::cout << x << y;
```


В результате на экране появится: 6 6.

При использовании постфиксной записи переменная *n* сначала применяется, а затем изменяется.

Пр и м е р:

```
int x, y;
```

```
x = 5;
```

```
y = x ++;
```

```
std::cout << x << y;
```

В результате на экране появится: 6 5.

2.4.4. Логические операции

При написании программ часто возникает необходимость проверить несколько условных выражений, например, даны два числа, если первое окажется положительным, а второе четным, то найти произведение этих чисел. В противном случае обнулить оба числа.

Прежде чем выполнить соответствующее действие, программа должна установить, что в данном случае оба условия истинны. Для принятия решения используют логические операторы. Основные логические операторы и правила их применения приведены в табл. 2.4.

Таблица 2.4

Логические операторы

Имя	Оператор	Пример	Значение
И	&&	Выражение 1 && выражение 2	Истинно, если оба выражения истинны; ложно – в противном случае
ИЛИ		Выражение 1 Выражение 2	Истинно, если хотя бы одно из выражений истинно; ложно – в противном случае
НЕ	!	! Выражение 1	Результат противоположен значению выражения 1

2.5. Комментарии. Специальные символы

Комментарии представляют собой текст, который игнорируется компилятором, но позволяет описать в тексте программы назначение отдельной строки или целого блока.

В языке C++ используются два типа комментариев: с двойной наклонной чертой (//) и сочетанием наклонной черты и звездочки (/*). Комментарии, которые начинаются с двойной наклонной черты, заканчиваются при переходе на новую строку.

Пример:

```
int x, y; // объявление переменных целого типа  
x = 5; // присвоение переменной x значения 5
```

При использовании комментариев с наклонной чертой и звездочкой компилятором игнорируется все, что следует за символами /*, до того момента, пока не встретится символ завершения комментария – звездочка и наклонная черта. Каждой открывающейся паре символов (/*) должна соответствовать закрывающаяся пара символов (*).

Пример:

```
y = x++; /* переменной y присвоено значение переменной x, затем значение переменной x увеличивается на единицу */
```

2.6. Математические функции

Если при написании программы возникает необходимость в использовании математических функций, то для этого в начале текста помещают директиву `#include<math.h>`, которая означает, что необходимо подключить заголовочный файл средств связи с библиотекой математических функций. Для правильного применения математических функций необходимо знать их обозначение на языке C++, приоритеты арифметических операций. Список основных математических функций приведен в табл. 2.5.

Список основных математических функций

Функция	Обозначение функции	Тип		Файл описания
		функции	аргумента	
1	2	3	4	5
Абсолютное значение	abs(x)	int	int	<stdlib.h>
	cabs(x)	double	struct	<math.h>
	fabs(x)	double	double	<math.h>
	labs(x)	long	long	<stdlib.h>
Арккосинус	acos(x)	double	double	<math.h>
Арсинус	asin(x)	double	double	<math.h>
Арктангенс	atan(x)	double	double	<math.h>
Косинус	cos(x)	double	double	<math.h>
Синус	sin(x)	double	double	<math.h>
Тангенс	tan(x)	double	double	<math.h>
Синус гиперболический	sinh(x)	double	double	<math.h>
Округление до большего целого	ceil(x)	double	double	<math.h>
Округление до меньшего целого	floor(x)	double	double	<math.h>
Экспоненциальная функция e^x	exp(x)	double	double	<math.h>
Экспоненциальная функция x^{2n}	ldexp(x,n)	double	double x int n	<math.h>
Степенная функция x^y	pow(x,y)	double	double	<math.h>
Степенная функция 10^n	pow10(n)	double	int	<math.h>
Логарифм натуральный	log(x)	double	double	<math.h>
Логарифм десятичный	log10(x)	double	double	<math.h>
Корень квадратный	sqrt(x)	double	double	<math.h>
Остаток деления x на y	fmod(x,y)	double	double	<math.h>

1	2	3	4	5
Генератор случайных чисел в диапазоне от 0 до 32767	rand()	int		<stdlib.h>
Инициализатор случайных чисел, начиная с числа n	strand(n)		int	<stdlib.h>

Примечание. Аргументы тригонометрических функций задаются в радианах.

2.7. Задания

Задание 1. Выполните запись арифметических выражений на языке C++ в соответствии со своим вариантом (табл. 2.6).

Таблица 2.6

Арифметические выражения

Вариант	Функция	Вариант	Функция
1	2	3	4
1	1) $Y = x^2 - \lg(\cos^3(x+4))$. 2) $Y = \arctg^3(\sqrt{x} - 3x^3)$. 3) $Y = \lg^3(2+x^3) - \sqrt[4]{x}$.	7	1) $Y = \sqrt{ \ln(x+6) + \cos x }$. 2) $Y = \sqrt{\ln(x^3) - 4x}$. 3) $Y = 8,05x^{-2} - \ln(\sqrt{x})$.
2	1) $Y = \sqrt{2,5x^3 - 3\cos x}$. 2) $Y = \sqrt{\cos x + 6,31x}$. 3) $Y = x^3 + 2,079\sqrt{\sin x}$.	8	1) $Y = \sin(7x^2) + \ln 2x$. 2) $Y = \frac{\cos(x+5)}{\lg(x^2)}$. 3) $Y = \lg^3(2+x^3) - \sqrt[4]{x}$.
3	1) $Y = 2\sin(\ln x^2 - 10,06)$. 2) $Y = \cos^3 2x - \ln x^2$. 3) $Y = \frac{x^2}{2,5x - \sin x}$.	9	1) $Y = \lg\left(\frac{x+2}{x^3 + \sin 3x}\right)$. 2) $Y = 11,28 \ln x^3 - \sqrt{x} $. 3) $Y = 2\sin(x^2) - 3\cos(x)$.
4	1) $Y = \sqrt{\frac{\cos 3x + 100}{x^3}}$. 2) $Y = \sqrt{53x^3 - 2 \lg x}$. 3) $Y = 5,02 \lg x^2 - \sqrt{x}$.	10	1) $Y = \lg(x^4) - \ln(x^3) - 1$. 2) $Y = \sqrt{\cos x + 6,31x}$. 3) $Y = \frac{\cos(x+5)}{\lg(x^2)}$.

1	2	3	4
5	1) $Y = \cos 7x + \sqrt{2x}$. 2) $Y = \sin(8x) + 5 x^7 $. 3) $Y = \operatorname{arctg}^3(\sqrt{x} - 3x^3)$.	11	1) $Y = 16x^7 + \ln(\sqrt{x})$. 2) $Y = \cos^9(x^5 + 1) - \operatorname{tg} x$. 3) $Z = \log_5(x + 5) \cdot \sqrt[7]{x}$.
6	1) $Y = 3,5x^{-4} + 9 \lg x^3 $. 2) $Y = \sqrt{\sin 2x - 9,99x^2}$. 3) $Y = x^{-3} - \ln 20,39 - \sqrt{x}$.	12	1) $Y = \cos^6(x \operatorname{tg} x + 34)$. 2) $Y = \cos 7x + \sqrt[5]{x + 8}$. 3) $Y = \operatorname{tg}^4(\sqrt{x + 7} - 9x^5)$.

Задание 2. Выполните запись арифметических выражений на общепринятом математическом языке в соответствии со своим вариантом (табл. 2.7).

Таблица 2.7

Математические выражения, записанные на языке C++

Вариант	Выражение
1	2
1	1) $z = \sin(x) + (x + \operatorname{pow}(a, 2)) / x + 3E - 8$. 2) $z = \operatorname{pow}(\tan(2E7 + a, 1./3) + 3 \cdot \log 10(x))$. 3) $y = \cos(\operatorname{fabs}(\operatorname{sqrt}(x + 6)/5)) + \operatorname{pow}(2, x)$.
2	1) $z = \sin(x) + (y + \operatorname{pow}(a, 2)) / 3E - 4$. 2) $y = \operatorname{pow}(\tan(2E7 * a), 3) / b + 3 * x$. 3) $t = \exp(4) + \cos(x + 6) / \operatorname{pow}(x, -3) + 7$.
3	1) $y = 1.2E - 5 + \operatorname{pow}(\cos(x), -6) / x + 8$. 2) $z = \operatorname{sqrt}(x + 3) - \log 10(\operatorname{pow}(x, 3)) + 78$. 3) $s = \exp(x + 5) / \operatorname{fabs}(x * \log(2 * x)) - 12$.
4	1) $d = \operatorname{pow}(\tan(x + 5), -6) / 2,3E - 2 + \operatorname{sqrt}(x + 67)$. 2) $y = \sin(\operatorname{sgrt}(x + 6)) + a \tan(x)$. 3) $z = \log 10(\operatorname{fabs}(x + 6)) + a \sin(x * 23)$.

1	2
5	1) $y = \sqrt{\text{pow}(2E - 3, x)} + 9.1 / \tan(\text{fabs}(x + 7)) + 45.$ 2) $s = \text{pow}(\sin(\text{pow}(x, 4)), -2) + 78.6.$ 3) $z = 1 / \tan(x + 3) * \log(\text{fabs}(x - 4)) + 6.$
6	1) $t = a \sin(\text{pow}(x, -7)) - \log_{10}(\text{fabs}(x / 2.3 + 6)).$ 2) $z = 12E - 3 / \sin(\sqrt{\text{fabs}(x + 3)}) + 7.5.$ 3) $y = \sqrt{\sin(\text{pow}(x, 2)) + 3.5 * \text{fabs}(x + 6)}.$
7	1) $z = 23E2 + \exp(x / 2.3) + \cos(3).$ 2) $b = a \cos(\text{pow}(x - 7, -4) + 45) / 12.4 + \log(x - 45).$ 3) $y = \log(x) / \log(5) * \text{fabs}(x + 2.5) - 12E - 6.$
8	1) $h = \text{pow}(\cos(x - 7), -2) / (12.3 - \tan(\text{fabs}(x))).$ 2) $c = \log_{10}(\text{fabs}(x + \sqrt{56.3})) + 6E - 7.$ 3) $z = 2.3 / \tan(45E2 + x) * \log(x - 8) + 45.$
9	1) $v = \sin(\text{pow}(x + 2, -4)) / 1 / \tan(x) + 67.$ 2) $r = \log(y / 4) / \log(8) - \text{fabs}(y + 1 / \tan(y)).$ 3) $z = \sqrt{\text{pow}(x, -3) + \sin(x)} - \log_{10}(34).$
10	1) $r = 1 / \text{pow}(\tan(x + 7), -8) + \sqrt{\text{fabs}(\sin(x))} + 6.$ 2) $v = \exp(x - 2) + a \cos(2.3 * x + 13).$ 3) $s = \text{pow}_{10}(x + 23) / \cos(\sqrt{x}) + 3.4.$
11	1) $y = \log(x) / \log(6) + \sqrt{\text{fabs}(\sin(x))} + 5.6.$ 2) $s = \sqrt{\cos(x + 2) - \tan(1 / 2 * x + 6.7)}.$ 3) $t = \exp(y + 7) / 1 / \tan(y * 3.4 + 6).$
12	1) $v = \sin(\text{pow}(x + 5, -3)) * \log(\text{fabs}(x)) / \log(9) + 7.8.$ 2) $t = 1 / \tan(z + 9) * \cos(z) + 7E - 4.$ 3) $z = \exp(\sqrt{x + 5}) / \text{fabs}(\sin(x * 2E - 3)).$

АЛГОРИТМЫ ЛИНЕЙНЫХ СТРУКТУР

Цель работы: изучение особенностей написания алгоритмов линейных структур, освоение правил ввода и вывода данных, выполнение вычислений в C++.

3.1. Понятие алгоритма

Алгоритм – строго определенная конечная последовательность арифметических и логических действий для решения задачи, обязательно приводящая к некоторому результату. Исполнителем алгоритма может быть человек или автоматическое устройство, способное воспринять предписания и выполнить предусмотренные в них действия.

Алгоритм имеет следующие свойства.

Дискретность. Описываемый с помощью алгоритма процесс должен быть разбит на последовательность отдельных шагов, т. е. алгоритм должен состоять из отдельных законченных действий.

Массовость. Обычно алгоритмы предназначены для решения не одной конкретной задачи, а некоторого класса задач определенного типа. В простейшем случае массовость подразумевает возможность использования различных исходных данных.

Определенность. Это свойство означает, что неоднозначность толкования записи алгоритма недопустима, многократное применение алгоритма к одним и тем же исходным данным должно приводить к одним результатам.

Результативность. Любой алгоритм должен завершаться за конечное число шагов.

Формальность. Любой исполнитель, способный воспринимать и выполнять инструкции алгоритма, действует формально, т. е. отвлекается от содержания поставленной задачи, не вникает в ее смысл, а лишь строго выполняет инструкции.

Способы описания алгоритма:

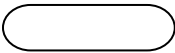
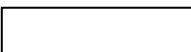
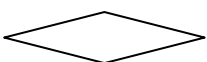
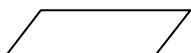


словесное описание, которое представляет структуру алгоритма на естественном языке (например, инструкция по эксплуатации бытового прибора);

псевдокод – описание структуры алгоритма на естественном, частично форматизированном языке, позволяющее выявить основные этапы решения задачи, перед точной его записью на языке программирования;

графическая схема алгоритма (ГСА) – описание структуры алгоритма с помощью геометрических фигур с линиями-связями, показывающими порядок выполнения отдельных инструкций. В ГСА каждой формальной конструкции соответствует определенная геометрическая фигура или связанная линиями совокупность фигур. Основные конструкции, используемые для построения ГСА, приведены в табл. 3.1.

Т а б л и ц а 3.1

Основные элементы ГСА

Символ (блок)	Обозначение	Пояснение
Пуск, останов		Начало, конец программы
Процесс		Вычислительное действие или их последовательность
Решение		Проверка условия
Ввод, вывод данных		Ввод, вывод данных
Предопределенный процесс		Вычисления по подпрограмме
Линия потока		Указатель последовательности связей между блоками

3.2. Алгоритмы линейных структур

Линейной называют алгоритмическую конструкцию, реализованную в виде последовательности действий (шагов), в которой каждое действие (шаг) алгоритма выполняется только один раз. –

Пример. Составьте алгоритм и программу вычисления $y = \sqrt[3]{tg^2(a+x)}$ при $a = -0,12$; $x = 0,75$.

Решение. ГСА вычисления приведена на рис. 3.1. Блоки 2 и 3 вводят и выводят исходные значения a и x . Блок 4 вычисляет значение y в соответствии с расчетной формулой. Блок 5 выводит результат.

Программа:

```
#include «stdafx.h»
#include<iostream>
#include<math.h>
using namespace std;
int _tmain(int argc, _TCHAR* argv[])
{   double y, a = -0.12, x = 0.75;
    cout<<a<<" "<<x<<endl;
    y = pow(tan(a+x),2/3.);
    std::cout<<y<<endl;
    return 0;
}
```

При запуске программы увидим ответ: 0.810085.

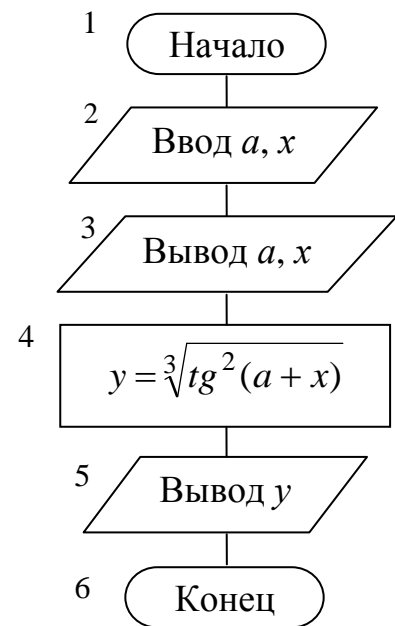


Рис. 3.1. Графическая схема линейного алгоритма

3.3. Задания

Задание 1. Составьте ГСА и программу вычисления значения функции при заданных значениях аргументов в соответствии с вариантами табл. 3.2, сравните результат с контрольным. В случае несовпадения результата вычислений в C++ с ответом в таблице выполните проверку в MS Excel.

Таблица 3.2

Вычисление значения функции по заданному значению аргумента

№ п/п	Функция	Исходные данные	Ответ
1	2	3	4
1	$t = \frac{2 \cos(x - \pi/6)}{1/3 - \sin^2 y}$	$x = 1,42;$ $y = 1,22$	$t = -2,277$
2	$y = \frac{ x^4 - \sqrt[3]{a/x} }{x - a^2}$	$x = 1,82;$ $a = 18,25$	$y = -0,027$

Окончание табл. 3.2

1	2	3	4
3	$s = \frac{e^3 + \sqrt[3]{a}}{\sin^2 a}$	$a = 1,378$	$s = 22,006$
4	$y = \frac{\sin^2 b - \cos b}{e^2 + b}$	$b = 2,2 \cdot 10^{-3}$	$y = -0,135$
5	$z = \cos^2(x^3) - \frac{x}{\sqrt{a^2 + x^2}}$	$x = 0,44;$ $a = 3,1 \cdot 10^{-2}$	$z = -0,0047$
6	$c = \sqrt{x^2 + b} - \sqrt[5]{x} b^2$	$x = 0,165;$ $b = 1,1 \cdot 10^{-4}$	$c = 0,147$
7	$z = \frac{3 \cdot 10^5 + \sqrt{a + \sqrt{b}}}{a + b}$	$a = 2,371;$ $b = 1171$	$z = 255,68$
8	$t = 17,5 \operatorname{tg}^2(x) - \frac{a}{\sin^2(x/a)}$	$x = -0,765;$ $a = 3,2$	$t = -41,01$
9	$a = \frac{c^{2x} + b^{-x}}{\sqrt{c x}}$	$x = 0,515;$ $c = 3,75;$ $b = 0,0002$	$a = 60,605$
10	$y = \frac{\sin^2 x}{\sqrt{1 + m^3 \sin(x^2)}}$	$x = 1,71;$ $m = -2,2 \cdot 10^{-3}$	$y = 0,98$
11	$a = \frac{\ln(b + x^2) + \operatorname{tg}^2 x}{\sqrt{ x - b }}$	$b = 4,78;$ $x = 10,378$	$a = 2,834$
12	$c = \frac{p^2 + \sin^2 p}{e^p + \sqrt[3]{p}}$	$p = 0,137$	$c = 0,0225$

Задание 2. Составьте ГСА и программу вычисления значения функции при $x = 5$ и $a = 3,3$ в соответствии со своим вариантом (табл. 3.3), сравните результат с контрольным.

Т а б л и ц а 3.3

Вычисление значения функции по заданному значению аргумента

№ п/п	Функция	Ответ
1	$y = \sqrt[3]{\frac{0,5 \cos^2(x^a) + e^{x^3}}{2 \ln x ^2}}$	0,67
2	$y = \frac{e^{\sin x} + \sqrt[3]{\cos x}}{\ln(\sqrt{a^2 + 3,71 - a}) + 4,5}$	0,27
3	$y = \sqrt[4]{\frac{7,61 \cos^3(3,5x) + 0,6}{\left a^3 - \sin^3 \frac{1}{x} \right }}$	0,371
4	$y = 2,31e^{x^{\cos ax}} - \sqrt{ \operatorname{tg} 3x - 12,7 ^3}$	−46,75
5	$y = 7,31a5x^{3a} - \cos^2(\sin(3,7x^3))$	$1,003 \cdot 10^9$
6	$y = \sqrt{\cos^3 85^\circ + \sqrt{1,24 + e^{x^{-2a}}}}$	3,02
7	$y = \frac{12,7 \sin^3(3x^2)}{\ln^2 \sqrt{x + e^a}}$	−0,245
8	$y = \cos^2 \sqrt{0,71x^{a\pi}} - x^3 $	−124,85
9	$y = \operatorname{arctg}\left(\sqrt[3]{9,8e^{0,2x} - \ln^2 x}\right)$	1,237
10	$y = 2,171a^3 \sqrt{e^{0,7x^3} - \cos^5 \frac{1}{x}}$	107,59
11	$y = 2,5 \sin^2(\cos 64^\circ + e^{-3x})$	0,45
12	$y = 4,39 \cos(\lg(3x^2)) - 1,95$	−3,265

З а д а н и е 3. Составьте программу по заданному условию в соответствии со своим вариантом. На экран выведите название программы, приглашение о вводе исходных данных. Исходные данные введите с клавиатуры. Напишите комментарии к получившемуся результату.

1. Рассчитать площадь параллелограмма.
2. Вычислить стоимость некоторого количества (по весу) яблок.
3. Преобразовать введенное с клавиатуры число в денежный формат.
4. Определить время, за которое путник пройдет определенное расстояние.
5. Даны два натуральных числа. Найти среднеарифметическое кубов этих чисел и среднее геометрическое модулей этих чисел.
6. Даны два действительных числа – x и y . Вычислить их сумму, разность, произведение и частное.
7. Дана длина ребра куба. Найти площадь грани, площадь полной поверхности и объем этого куба.
8. Известна длина окружности. Найти площадь круга, ограниченного этой окружностью.
9. Найти площадь треугольника, две стороны которого равны a и b , а угол между этими сторонами равен β .
10. Найти все углы треугольника со сторонами a , b , c . Предусмотреть в программе перевод радианной меры угла в градусы, минуты и секунды.
11. Вычислить расстояние между двумя точками с данными координатами $(x_1; y_1)$ и $(x_2; y_2)$.
12. Вычислить сумму цифр заданного трехзначного числа.
13. Найти произведение цифр заданного четырехзначного числа.
14. Дано некоторое число a . Не используя никаких функций и никаких операций, кроме умножения, получить a^8 за три операции.

Лабораторная работа 4

АЛГОРИТМЫ РАЗВЕТВЛЯЮЩИХСЯ СТРУКТУР

Цель работы: изучение особенностей написания программ разветвляющихся структур.

4.1. Условный оператор перехода

Вычислительные процессы, в которых в зависимости от тех или иных условий должны выполняться различные ветви алгоритма вычисления, называются *разветвляющимися*.

Для построения алгоритмов, реализующих такие вычислительные процессы, необходимы специальные команды (управляющие структуры), позволяющие управлять ходом выполнения алгоритма, а именно осуществить выбор одного из нескольких всевозможных действий.

Условный оператор позволяет выбрать и выполнить один из двух входящих в него операторов в зависимости от значения некоторого выражения. Условный оператор имеет сокращенную и полную форму.

1. Сокращенная форма:

If (выражение – условие)

Оператор 1;

Оператор 2;

Если проверяемое выражение принимает значение «истина», то выполняется оператор 1, затем выполняется оператор 2. Если проверяемое условие не выполняется, т. е. оно «ложь», то управление передается оператору 2.

Графическая схема алгоритма сокращенной формы условного оператора показана на рис. 4.1.

Пример. Введите с клавиатуры некоторое значение переменной x . Если значение x отрицательно, то измените знак x на противоположный. Выведите полученное значение x .

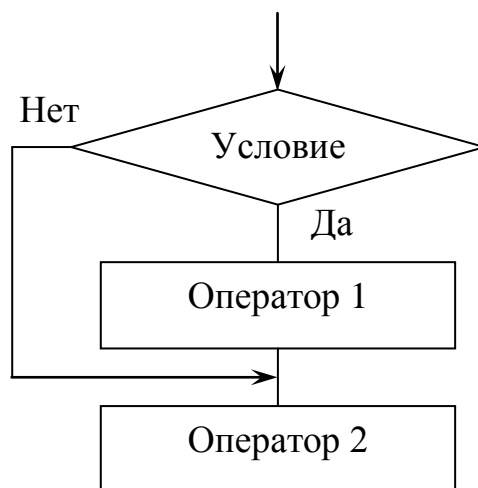


Рис. 4.1. ГСА сокращенной формы условного оператора

```

#include "stdafx.h"
#include<iostream>
using namespace std;
int _tmain(int argc, _TCHAR* argv[])
{
    double x;
    cout<<"введите x\t";
    cin>>x;
    if(x<0)
        x = -x;
    cout<<"x="<<x<<endl;
    return 0;
}

```

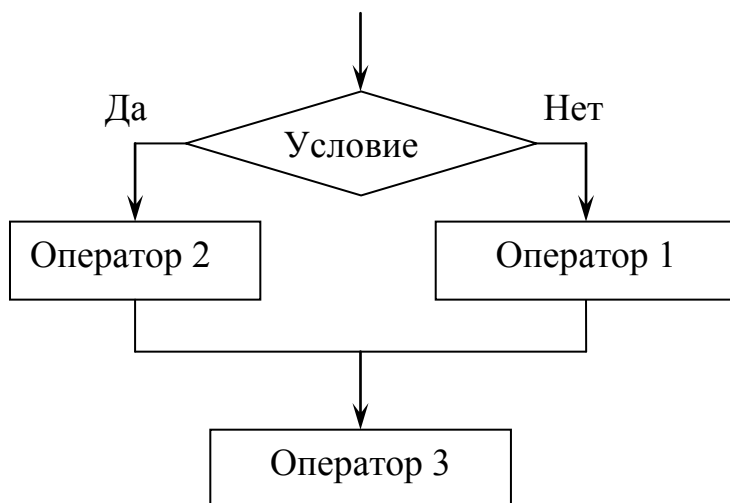


Рис. 4.2. ГСА полной формы
условного оператора

2. Полная форма:

```

if (выражение – условие)
    Оператор 1;
else
    Оператор 2;
Оператор 3;

```

Если условное выражение принимает значение «истина», то выполняется оператор 1; если оно принимает значение «ложь», то выполняется оператор 2. И в любом случае далее выполняется оператор 3.

Графическая схема алгоритма полной формы условного оператора показана на рис. 4.2.

Пример. Вычислите модуль введенного с клавиатуры числа.

Программа:

```

#include "stdafx.h"
#include<iostream>
using namespace std;
int _tmain(int argc, _TCHAR* argv[])
{

```

```

double x,y;
std::cout<<"Vvedite chislo:\t";
std::cin>>x;
if(x>=0)
    y = x;
else
    y = -x;
std::cout<<"/"<<x<<"!="<<y<<endl;
return 0;
}

```

Оператор *if* может иметь вложенную конструкцию. На любом уровне вложенности каждый внутренний *if* должен полностью входить во внешний для него *if*.

Если необходимо выполнить одновременно несколько операторов, то используется составной оператор или блок, представляющий собой группу операторов, заключенных в операторные скобки:

```

if (выражение-условие)
{
    Оператор 1;
    .....
    Оператор m;
}
else
{
    Оператор m+1;
    .....
    Оператор m+n;
}

```

4.2. Операция-условие

Операция-условие в некоторых случаях позволяет более компактно представлять разветвляющийся процесс. Операция-условие имеет следующую конструкцию: *(выражение 1) ? выражение 2 : выражение 3*;

Если выражение 1 истинно, то значением всего условного выражения будет выражение 2. Если выражение 1 ложно, то за значение условного выражения принимается величина, вычисляемая в выражении 3.

Пример. Определите абсолютное значение целого числа.

Программа:

```
#include "stdafx.h"
#include<iostream>
using namespace std;
int _tmain(int argc, _TCHAR* argv[])
{
    int x,y;
    cout<<"Vvedite chislo\t";
    cin>>x;
    y=(x<0) ? -x : x;
    cout<<"/"<<x<<"/="<<y<<endl;
    return 0;
}
```

4.3. Оператор-переключатель

Если в программе необходимо выбрать один из нескольких многочисленных вариантов, то вместо вложенной конструкции *if* более целесообразно применить оператор-переключатель *switch*, имеющий следующий синтаксис.

```
Switch (<выражение>)
{
    case <константа 1>:<операторы 1>;
    case <константа 2>:<операторы 2>;
    .....
    case <константа n>:<операторы n>;
    [ default: <операторы>;]
}
```

Для выполнения выбирается тот вариант (группа операторов), константа которого совпадает со значением выражения. Как выражение, так и метки (константы) должны иметь значения целого или символьного типа.

После выполнения выбранной группы операторов будут выполняться все оставшиеся операторы до тех пор, пока не произойдет новый переход. Если в конце выбранного варианта поместить оператор *break* (разрыв), то управление сразу будет передано в конец оператора-переключателя.

Когда некоторому значению выражения не соответствует никакая метка, управление передается операторам с меткой *default* (прочие).

Вариант *default* не обязательно должен быть последним и вообще может отсутствовать. В последнем случае, если значение выражения не соответствует ни одной из констант, управление передается оператору, следующему за оператором *switch*.

Пример. Программа работы простейшего калькулятора.

```
#include "stdafx.h"
#include <iostream>
using namespace std;
int _tmain(int argc, _TCHAR* argv[])
{
    double a, b, y;
    char znak;
    cout << "vvedite a, znak operacii, b\t";
    cin >> a >> znak >> b;
    switch (znak)
    {
        case '+': y = a + b; break;
        case '-': y = a - b; break;
        case '*': y = a * b; break;
        case '/': y = a / b; break;
        default: cout << "neverno vveli znak";
    }
    cout << a << znak << b << " = " << y << endl; return 0;
return 0;
}
```

4.4. Задания

З а д а н и е 1. Составьте ГСА и программу по заданному условию в соответствии со своим вариантом (табл. 4.1).

Т а б л и ц а 4.1

Функции для вычисления

№ п/п	Функция	Исходные данные	Результат при заданном значении x
1	2	3	4
1	$z = \begin{cases} \sqrt{a + \sin^2 x^2 \cdot b}, & \text{если } x \text{ четное число;} \\ \sqrt[3]{a + b^2 \cdot \ln b} & \text{в остальных случаях} \end{cases}$	$a = 0,003$ $b = 4,65$	$x = 6 \quad z = 2,139$
			$x = 3 \quad z = 33,375$
2	$z = \begin{cases} \arctg x + \sqrt{b \cdot x}, & \text{если } x \text{ целое число;} \\ \lg^2 a^3 + \sqrt[3]{a + b} & \text{в остальных случаях} \end{cases}$	$a = 1,3$ $b = 4 \cdot 10^2$	$x = 3 \quad z = 35,89$
			$x = 4,5 \quad z = 9,287$
3	$z = \begin{cases} a \cdot x^2 + (a + 1) \cdot x + 1, & \text{если } x < 2; \\ \lg^2(a + x) - b & \text{в остальных случаях} \end{cases}$	$a = 38$ $b = 0,21$	$x = 1,5 \quad z = 145$
			$x = 2,5 \quad z = -0,084$
4	$z = \begin{cases} \sqrt[3]{a \cdot x + b}, & \text{если } x \text{ кратно } 3; \\ \sin^2(a \cdot x) + \ln^2 x & \text{в остальных случаях} \end{cases}$	$a = 800$ $b = 0,13$	$x = 9 \quad z = 19,31$
			$x = 7 \quad z = 3,58$
5	$z = \begin{cases} \sqrt{2 \sin^2 x^2}, & \text{если целая часть } x > 4; \\ a^2 + 5 \lg(x + b)^2 & \text{в остальных случаях} \end{cases}$	$a = 0,0006$ $b = 1,2$	$x = 5,2 \quad z = 1,34$
			$x = 3,3 \quad z = 6,53$
6	$z = \begin{cases} \sqrt{ a + x + \operatorname{ctg} x}, & \text{если } x \text{ четное число;} \\ \sqrt[3]{x} + \ln^2(x + 2) & \text{в остальных случаях} \end{cases}$	$a = 0,005$	$x = 4 \quad z = 2,207$
			$x = 5 \quad z = 5,497$
7	$z = \begin{cases} 0,03 \cdot a \cdot \arctg \sqrt{x}, & \text{если } x \text{ кратно } 5; \\ a^3 - e^{x+3} \cdot \ln a & \text{в остальных случаях} \end{cases}$	$a = 0,42$	$x = 25 \quad z = 0,02$
			$x = -13 \quad z = 0,07$
8	$z = \begin{cases} \sqrt{b \cdot x^2}, & \text{если дробная часть } x < 0,5; \\ x \cdot \arcsin \ln x & \text{в остальных случаях} \end{cases}$	$a = 0,5$ $b = 2,07$	$x = 2,3 \quad z = 3,31$
			$x = 1,9 \quad z = 1,32$

Окончание табл. 4.1

1	2	3	4
9	$z = \begin{cases} \sqrt[5]{ \sin x } \cdot \lg x, & \text{если } x > 0; \\ (x + a) - a^2 \cdot x & \text{в остальных случаях} \end{cases}$	$a = 2,25$	$x = 2,5 \quad z = 0,36$
			$x = -1 \quad z = 6,313$
10	$z = \begin{cases} \sqrt[3]{ x } + \log_3^2 b, & \text{если } x \geq 1; \\ \sqrt{a + \sin^2 x^2} & \text{в остальных случаях} \end{cases}$	$a = 0,003$ $b = 4,91$	$x = 1,5 \quad z = 3,24$
			$x = 0,2 \quad z = 0,07$
11	$z = \begin{cases} 0,8 \cdot a^2 \cdot \sin \sqrt{x}, & \text{если } x \text{ кратно } 3; \\ e^{x+3} + \cos x^2 & \text{в остальных случаях} \end{cases}$	$a = 0,83$	$x = 6 \quad z = 0,35$
			$x = 0,3 \quad z = 28,1$
12	$z = \begin{cases} b \cdot x^3 + (a + 1)^x, & \text{если } x < 2; \\ \operatorname{tg}^2(a + x^b) & \text{в остальных случаях} \end{cases}$	$a = 4$ $b = 3,5$	$x = 1 \quad z = 8,5$
			$x = 3 \quad z = 0,298$

З а д а н и е 2. Составьте ГСА и программу по заданному условию в соответствии со своим вариантом (табл. 4.2).

Таблица 4.2

Функции для вычисления

№ п/п	Функция	Исходные данные	Контрольные значения	Результат
1	2	3	4	5
1	$y = \begin{cases} ax^2 + b, & x < 0,2; \\ \pi x, & 0,2 \leq x \leq 2,2; \\ \sqrt{x + e^x}, & x > 2,2 \end{cases}$	$a = 0,12;$ $b = -4,4$	$x_1 = 0,17;$ $x_2 = 1,2;$ $x_3 = 3,9$	$y_1 = -4,39;$ $y_2 = 3,76;$ $y_3 = 7,3$
2	$y = \begin{cases} \sin \pi x, & x \leq 0; \\ \sqrt{ \cos \pi x }, & 0 < x < 1; \\ \operatorname{tg}^2(Lx), & x \geq 1 \end{cases}$	$L = 6,2$	$x_1 = -1,1;$ $x_2 = 0,2;$ $x_3 = 1,6$	$y_1 = 0,31;$ $y_2 = 0,89;$ $y_3 = 0,29$

Продолжение табл. 4.2

1	2	3	4	5
3	$y = \begin{cases} \ln x^2 + 1 , & x < 2; \\ ax^3, & x = 2; \\ \sqrt{ax + 0,2}, & x > 2 \end{cases}$	$a = 2,8$	$x_1 = 0;$ $x_2 = 2;$ $x_3 = 5,3$	$y_1 = 0;$ $y_2 = 22,4;$ $y_3 = 3,88$
4	$y = \begin{cases} a \cdot \sin(\pi x + b), & ax < 0; \\ 0, & ax = 0; \\ \ln(x + a), & ax > 0 \end{cases}$	$a = 0,01;$ $b = -2\pi$	$x_1 = -3,6;$ $x_2 = 0;$ $x_3 = 7,1$	$y_1 = 0,009;$ $y_2 = 0;$ $y_3 = 1,9$
5	$y = \begin{cases} \ln a + bx , & x < 0,3; \\ ax^2 + b, & 0,3 \leq x < 0,7; \\ \sqrt{7x + b}, & x \geq 0,7 \end{cases}$	$a = 98,3;$ $b = 4,5$	$x_1 = 0,1;$ $x_2 = 0,33;$ $x_3 = 1,8$	$y_1 = 4,59;$ $y_2 = 15,2;$ $y_3 = 4,13$
6	$y = \begin{cases} x^2 + a^3, & x < a; \\ \sqrt{ax + b}, & a \leq x \leq b; \\ -0,3 \lg x, & x > b \end{cases}$	$a = 2,81;$ $b = 5,96$	$x_1 = -12,6;$ $x_2 = 3,2;$ $x_3 = 10,3$	$y_1 = 180,9;$ $y_2 = 3,867;$ $y_3 = -0,3$
7	$y = \begin{cases} \sin x + \cos^2 x, & x \leq \pi/2; \\ \sin(ax^2), & \pi/2 < x \leq 3\pi/2; \\ \operatorname{tg}(x + a), & x > 3\pi/2 \end{cases}$	$a = 4,4$	$x_1 = -7,5;$ $x_2 = 3,98;$ $x_3 = 1000$	$y_1 = -0,82;$ $y_2 = 0,55;$ $y_3 = -1,28$
8	$y = \begin{cases} 1 + \sqrt{ax}, & x < a; \\ \sqrt{a} + \lg x^2, & x = a; \\ \sin ax, & x > a \end{cases}$	$a = 2$	$x_1 = 1,3;$ $x_2 = 2;$ $x_3 = 9,9$	$y_1 = 2,612;$ $y_2 = 2,016;$ $y_3 = 0,813$
9	$y = \begin{cases} \lg x , & x < -0,2; \\ a + 0,65e^x, & -0,2 \leq x \leq 0,2; \\ \sin^2 ax + \cos ax^2, & x > 0,2 \end{cases}$	$a = 11,3$	$x_1 = -2,3;$ $x_2 = 0,1;$ $x_3 = 2,3$	$y_1 = 0,362;$ $y_2 = 12,01;$ $y_3 = -0,42$
10	$y = \begin{cases} (x^{-3} + e^x) \sin x^2, & ax < 2; \\ 1 - 0,7 \ln^2 ax, & ax = 2; \\ \lg x^2, & ax > 2 \end{cases}$	$a = 0,1$	$x_1 = -9,2;$ $x_2 = 20;$ $x_3 = 28,3$	$y_1 = -0,002;$ $y_2 = 0,66;$ $y_3 = 2,90$

1	2	3	4	5
11	$y = \begin{cases} d \sin \sqrt[3]{ x }, & x \leq -15; \\ e^{d\sqrt{x}}, & -15 < x < 5; \\ \cos \ln x, & x \geq 5 \end{cases}$	$d = 2,5$	$x_1 = -25,1;$ $x_2 = 2,6;$ $x_3 = 13,3$	$y_1 = 0,53;$ $y_2 = 56,3;$ $y_3 = -0,9$
12	$y = \begin{cases} \sqrt[3]{ xc }, & x < -1; \\ \sin(\ln x^2), & -1 \leq x \leq 8,9; \\ \cos^3 \sqrt{x+c}, & x > 8,9 \end{cases}$	$c = 0,04$	$x_1 = -9;$ $x_2 = 3,4;$ $x_3 = 31$	$y_1 = 0,711;$ $y_2 = 0,64;$ $y_3 = 0,434$

З а д а н и е 3. Составьте программу по заданному условию в соответствии со своим вариантом.

1. Даны два числа – K и L. Вывести на печать то из них, для которого получается больший целый остаток от его деления на 4.

2. Даны два числа – A и D. Определить и вывести на печать остаток от деления большего из них на меньшее.

3. Даны два вещественных числа – B и C. Вывести на печать B и C в порядке убывания их целых частей.

4. Даны два числа – R и N. Вывести числа на печать в порядке убывания их модулей.

5. Даны два числа – K и N. Если разность этих чисел является четным числом, то вывести на экран их сумму, в противном случае вывести на экран их произведение.

6. Даны два числа – V и W. Вывести на печать сумму их модулей, если их произведение кратно 3. В противном случае вывести на печать их разность.

7. Даны два числа – C и W. Определить, является ли первое число кратным второму.

8. Даны два вещественных числа – P и S. Вывести на печать их сумму, если целая часть их разности не превышает 10, в противном случае вывести на печать их произведение.

9. Даны два числа – C и K. Вывести на печать их разность, если целая часть их отношения равна нулю, в противном случае вывести на печать их сумму.

10. Даны два вещественных числа – J и M. Определить, является ли целая часть их суммы кратной 5.

11. Из двух заданных вещественных чисел A и N вывести на печать то, у которого дробная часть меньше.

12. Даны два вещественных числа – P и Q. Вывести их на печать в порядке убывания их дробных частей.

З а д а н и е 4. Составьте программу по заданному условию в соответствии со своим вариантом.

1. Даны два вещественных числа – C и D. Вывести на печать их произведение, если второе число имеет целую часть больше, чем первое, а дробная часть их суммы не превышает 0,35. В противном случае вывести на печать их сумму.

2. Даны два числа – A и B. Если первое число окажется четным, а второе при этом кратным 5, то вывести на печать их частное. В противном случае вывести на печать результат целочисленного деления A на B.

3. Даны два числа – K и P. Если второе число окажется нечетным, а первое при этом будет представлять собой целый остаток от деления второго числа на 3, то вывести на печать сумму квадратов этих чисел. В противном случае оба числа обнулить.

4. Даны два числа – N и M. Если первое число окажется кратным 5 или второе число будет нечетным, то вывести на экран сумму модулей заданных чисел. В противном случае вывести разность их квадратов.

5. Даны два вещественных числа – W и R. Если хотя бы одно из них имеет целую часть, кратную 3, то вывести на печать произведение заданных чисел. В противном случае вывести на печать их разность.

6. Даны два вещественных числа – D и B. Если хотя бы одно из них имеет дробную часть, не превышающую 0,43, то вывести на печать среднее арифметическое заданных чисел. В противном случае числа обнулить.

7. Даны два вещественных числа – U и L. Если первое число имеет дробную часть больше 0,75 или второе имеет целую часть, кратную 10, то вывести на печать разность квадратов заданных чисел. В противном случае вывести на печать их произведение.

8. Даны три числа – a, b, c. Определить максимальное из них и вывести его на печать.

9. Даны три числа – k , p , m . Определить минимальное из них и вывести его на печать.

10. Определить, в каком квадранте находится точка A с координатами $(x; y)$. Вывести на печать координаты точки и номер квадранта.

11. Даны три натуральных числа. Определить, можно ли построить треугольник с такими длинами сторон (сумма любых двух сторон должна быть больше третьей стороны).

12. Даны два вещественных числа – a и b . Вычислить, если это возможно, сумму натуральных логарифмов этих чисел.

З а д а н и е 5. Выполнить задание в соответствии со своим вариантом.

1. Написать программу, которая по введенному номеру дня недели выдает в качестве результата название этого дня.

2. Написать программу, которая по номеру дня недели выдает в качестве результата расписание занятий в этот день.

3. Составить программу, которая по заданным году и номеру месяца определяет количество дней в этом месяце.

4. Для целого числа a от 1 до 99 напечатать фразу «Мне a лет», учитывая при этом, что при некоторых значениях a слово «лет» надо заменить на слово «год» или «года», например: 11 лет, 22 года, 51 год.

5. Для каждой введенной цифры (0 – 9) вывести соответствующее ей название на английском языке (0 – zero, 1 – one, 2 – two, ...).

6. Составить программу, позволяющую получить словесное описание школьных отметок (1 – плохо, 2 – неудовлетворительно, 3 – удовлетворительно, 4 – хорошо, 5 – отлично).

7. Пусть элементами круга являются радиус (первый элемент), диаметр (второй элемент) и длина окружности (третий элемент). Составить программу, которая по номеру элемента запрашивала бы его соответствующее значение и вычисляла бы площадь круга.

8. Написать программу, которая по введенному номеру времени года (1 – зима, 2 – весна, 3 – лето, 4 – осень) выдавала бы соответствующие этому времени года месяцы и количество дней в каждом из месяцев.

9. Написать программу, которая по введенному номеру единицы измерения (1 – дециметр, 2 – километр, 3 – метр, 4 – миллиметр, 5 – сантиметр) и длине отрезка L выдавала бы соответствующее значение длины отрезка в метрах.

10. Написать программу, которая по вводимому числу от 1 до 5 (номеру курса) выдает соответствующее сообщение «Привет, k-курсник». Например, если $k = 1$, «Привет, первокурсник»; если $k = 4$: «Привет, четверокурсник».

11. Написать программу, которая по данному натуральному числу от 1 до 12 (номеру месяца) выдает все приходящиеся на этот месяц праздничные дни (например, если введено число 1, то 1 января – Новый год, 7 января – Рождество).

12. Написать программу, которая по вводимому числу от 1 до 4 выдает название времени года.

Библиографический список

1. Либерти Д. Освой самостоятельно C++ за 21 день / Д. Либерти. М.: Вильямс, 2015. 784 с.

2. Подбельский В. В. Язык C++ / В. В. Подбельский М.: Финансы и статистика, 2014. 560 с.

Учебное издание

ТРОФИМОВА Людмила Николаевна,
ЕРОШЕНКО Александра Викторовна

ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ C++

Часть 1

Учебно-методическое пособие

Редактор Н. А. Майорова
Корректор А. А. Булдакова

Подписано в печать 05.10.2017. Формат $60 \times 84 \frac{1}{16}$.
Офсетная печать. Бумага офсетная. Усл. печ. л. 2,5. Уч.-изд. л. 2,8.
Тираж 75 экз. Заказ .

**

Редакционно-издательский отдел ОмГУПСа
Типография ОмГУПСа

*

644046, г. Омск, пр. Маркса, 35