

Основы программирования

Лабораторная работа 2

Тема: Логические операторы и ветвление

Цель лабораторной работы

Целями данной работы являются изучение условных операторов и обучение написанию программ с разветвленной последовательностью действий.

Краткие теоретические сведения

Операции сравнения и логические операции

Операции сравнения и логические операции в качестве операндов и результатов операций используют значения **true** и **false**. В языке C++ значения **true** и **false** представлены типом **bool**.

Операции сравнения и логические операции представлены следующими операторами:

Операторы сравнения	
Оператор	Операция
>	Левый операнд <i>больше</i> чем правый
>=	Левый операнд <i>больше или равен</i> правому
<	Левый операнд <i>меньше</i> чем правый
<=	Левый операнд <i>меньше или равен</i> правому
==	Левый операнд <i>равен</i> правому
!=	Левый операнд <i>не равен</i> правому
Логические операции	
Оператор	Операция
&&	И, конъюнкция
	ИЛИ, дизъюнкция
!	НЕ, отрицание, отрицательная инверсия

Таблица истинности для логических операций:

P	Q	P && Q	P Q	!P
false	false	false	false	true
false	true	false	true	true
true	false	false	true	false
true	true	true	true	false

Как операции сравнения, так и логические операции имеют более низкий приоритет по сравнению с арифметическими операциями (кроме постфиксных форм *инкремента* и *декремента*).

Приоритет логических операций и операций сравнения следующий:

Высший приоритет
Отрицание (!)
Операторы сравнения > >= < <=
Операторы равенства == !=
Конъюнкция (&&)
Дизъюнкция ()
Низший приоритет

Как и в арифметических выражениях, для изменения порядка выполнения операций сравнения и логических операций можно использовать круглые скобки.

Условные операторы

Условные операторы позволяют организовать алгоритмические структуры «*Ветвление*».

Оператор if

Данный оператор служит для того, чтобы на основе проверки истинности некоторого условия (*условной конструкции*) выполнить или не выполнить некоторые действия. Так реализуется вариант конструкции *Ветвления* – **Обход**. Общая форма оператора:

```
if (условное выражение) оператор;
```

Пример:

```
if (x < 0) x = -x;
```

Также можно с помощью фигурных скобок определять блок действий:

```
if (x < 0)
{
    x = -x;
    cout << x << endl;
}
```

Оператор if-else

Данный оператор служит для того, чтобы на основе проверки истинности некоторого условия (*условной конструкции*) выбрать, какой из двух вариантов действий выполнить: один – если условие *истинно*, второй – если условие *ложно*. Так реализуется вариант конструкции *Ветвления* – **Альтернатива**. Общая форма оператора:

```
if (условное выражение) оператор1;
else оператор2;
```

Пример:

```
if (x > 0) cout << "Число больше 0";
else cout << "Число НЕ больше 0";
```

Также можно с помощью фигурных скобок определять блок действий:

```
if (x > y)
{
    r = x - y;
    cout << "Разница между X и Y = " << r;
}
else
{
    r = y - x;
    cout << "Разница между Y и X = " << r;
}
```

Конструкция if-else-if

В программах часто используется конструкция, которую называют *лестницей if-else-if*. Так можно реализовать вариант конструкции *Ветвления* – **Выбор**. Общая форма:

```

if (условное выражение) оператор;
else if (условное выражение) оператор;
...
else оператор;

```

Пример:

```

int x = 1;
if (x == 0) cout << "Нулевой вариант";
else if (x == 1) cout << "Первый вариант";
else if (x == 2) cout << "Второй вариант";
else cout << "Подходящий вариант не выбран";

```

Тернарный оператор ?:

Данный оператор можно использовать вместо оператора *if-else*, записанного в форме:

```

if (условие) переменная = выражение;
else переменная = выражение;

```

Оператор *?:* является *тернарным*, потому что он имеет три операнда. Общая форма:

```

Выражение1 ? Выражение2 : Выражение3;

```

Результат работы оператора определяется следующим образом.

Сначала вычисляется *Выражение1*. Если оно имеет значение *true*, вычисляется *Выражение2* и его значение становится **результатом**. Если *Выражение1* имеет значение *false*, то вычисляется *Выражение3*, и его значение становится **результатом**.

Например:

```

int x = 10;
int y = x > 9 ? 100 : 200; // в y запишется 100, так как 10 > 9

```

Оператор switch

Этот оператор предназначен для выбора ветви вычислительного процесса исходя из значения *управляющего выражения*. Общая форма оператора:

```

switch (управляющее выражение)
{
    case значение1:
        последовательность операторов;
        break;
    case значение2:
        последовательность операторов;
        break;
    ...
    default:
        последовательность операторов;
}

```

Пример:

```

char ch;
cin >> ch; /* чтение символа */
switch(ch){
    case '1': cout << "Введена 1!"; break;
    case '2': cout << "Введена 2!"; break;
    case '3': cout << "Введена 3!"; break;
    default : cout << "Не введена цифра от 1 до 3";
}

```

Задание

Разработать программу для каждой задачи из **общего** и **индивидуального** задания.

Типы переменных устанавливаются в соответствии с условием задачи. Используйте **поточковый ввод-вывод** (*cin, cout*).

В первом индивидуальном задании используйте **условные операторы** или **тернарный оператор** в решении.

Во втором индивидуальном задании написать программу с использованием оператора множественного выбора **switch**.

Методические указания

Выполните **общие** и **индивидуальные** задания.

При выполнении **общих** заданий не требуется отчет, но все еще нужна защита с объяснением кода программы.

При выполнении **индивидуальных** заданий необходимо выполнить следующие этапы для каждой задачи отдельно:

- 1) словесная постановка задачи;
- 2) анализ задачи и формальная постановка задачи;
- 3) проектирование (разработка алгоритма) – лучше всего в графической форме (блок-схема), но можно и в словесной форме или псевдокод;
- 4) реализация (кодирование, отладка);
- 5) тестирование.

Результаты выполнения **индивидуальных** заданий оформить в виде отчета.

Общее задание

При выполнении использовать **логические** и **арифметические** операторы (**без условных операторов**). В решении необходимо использовать особенность преобразования типа **bool** в целое число в арифметических выражениях (*true* = 1, *false* = 0).

Пользователь вводит с клавиатуры стоимость товара (вещественное число) в рублях.

Если стоимость больше 100 рублей, то сделать скидку ровно в 5 рублей.

Если же скидка больше 1000 рублей, то сделать скидку в 50 рублей.

Вывести, сколько в итоге нужно заплатить за товар.

Примеры работы программы:

Стоимость товара (руб.): 150

Сумма к оплате: 145 руб.

Стоимость товара (руб.): 1100

Сумма к оплате: 1050 руб.

Индивидуальные задания

Вариант 1

1. Даны три вещественных числа x , y , z . Вывести наибольшее и наименьшее среди них.
2. Задано начальное направление стрелки: «север», «восток», «запад» или «юг» (определяется просто числом от 1 до 4-х). Вводится символ 'w', 'a' или 'd', обозначающий поворот направления: 'w' – не поворачиваться, 'a' – поворот влево, 'd' – поворот вправо. Вывести новое направление после поворота. Если символ некорректен – вывести сообщение об этом.

Вариант 2

1. Даны два вещественных числа x , y и логическая переменная b . Если b равна истине, то нужно преобразовать числа в *положительные*. Если b равна лжи, то нужно преобразовать числа в *отрицательные*.
2. Даны два вещественных числа и символ, обозначающий арифметическую операцию ('+', '-', '*' или '\'). Выполнить арифметическое действие с этими числами, соответствующее заданному символу. Если задан некорректный символ – вывести сообщение об этом.

Вариант 3

1. Даны три вещественных числа x , y , z . Посчитать сколько среди них *положительных* и сколько *отрицательных* чисел.
2. В месяце 31 день. Первый день – понедельник. Дан номер дня. Вывести каким днем недели является заданный день. Если дан некорректный номер – вывести сообщение об этом. В решении данной задачи вам поможет **оператор %** (нет необходимости расписывать *switch* для всех 31 дней).

Вариант 4

1. Даны три вещественных числа x , y , z . Составить программу, выводящую отдельно те из них, которые не принадлежат отрезку $[-5, 5]$, и те, что не принадлежат.
2. Колода карт состоит из 36 карт. Она упорядочена по номиналам карты от «6» до «Туза». Также масти карт следуют друг за другом: «пики», «черви», «буби» и «крести». Задан номер карты. Вывести отдельно *номинал* и *масть* карты. Если задан некорректный номер – вывести сообщение об этом. В решении данной задачи вам помогут **операторы % и /**.

Вариант 5

1. Даны три вещественных числа x , y , z . Вывести сообщение о том, каких чисел среди них больше, *четных* или *нечетных*.

2. Задан номер месяца. Вывести название заданного месяца. Отдельно выведите время года, которому принадлежит этот месяц. Если задан некорректный номер – вывести сообщение об этом. Старайтесь, чтобы строки вывода сообщений не повторялись. В решении данной задачи вам поможет **оператор /**.

Вариант 6

1. Даны три вещественных числа x , y , z . Составить программу, вычисляющую, какие два числа дают минимальное произведение.

2. Даны два числа и символ '<', '>' или '='. В зависимости от символа, вывести наименьшее ('<') число, наибольшее число ('>') или вывести сообщение о том, равны ли числа ('='). Если заданный символ некорректный – вывести сообщение об этом.

Вариант 7

1. Даны два действительных числа x , y . Если $x > y$, значение x заменить на среднее арифметическое этих двух чисел, иначе y – на их произведение. Вывести x , y .

2. Вывести название пальца и конечности, где он находится, по заданному номеру пальца. Пальцы номеруются от «большого» до «мизинца». С 1 по 5 пальцы – левая рука, с 6 по 10 – правая рука, с 11 по 15 пальцы – левая нога, с 16 по 20 – правая нога. Название конечности выводите отдельно. Старайтесь, чтобы строки вывода сообщений не повторялись. Если номер некорректный – вывести сообщение об этом.

Вариант 8

1. Даны два действительных числа x , y . Вывести их полусумму, если $x+5 > y-5$, и их разность, если это не так.

2. Даны стоимость товара и категория скидки. Вычислить стоимость с учетом скидки: категория 1 – 30%; категория 2 – 20 %; категория 3 – 10 %; категория 4 – 5 %; категория 5 – 3 %; категория 6 – 1%. Заданная сумма меньше или равна 0, или категория скидки некорректна – вывести сообщение об этом.

Вариант 9

1. Даны три действительных числа x , y , z . Составить программу, вычисляющую максимальное из $x+y$, $y+z$ и $x+z$.

2. Задается номер диапазона: 1 – [0,1], 2 – [0,5], 3 – [0,10], 4 – [0,50] , 5 – [0,100]. Если задано некорректное число – вывести сообщение об этом. Генерируется случайное число в данном диапазоне (см. функ. *rand()*). После этого пользователь вводит еще одно число. Если

оно совпало со сгенерированным, то вы вывести сообщение «Победа». Если нет, вывести сгенерированное число как ответ.

Вариант 10

1. Даны четыре действительных числа x_1, x_2, x_3, x_4 . Найти наибольшее число.
2. Соответствие веса номеру категории: 1 – 10 кг, 2 – 50 кг, 3 – 100 кг, 4 – 250 кг, 5 – 500 кг. Даны номера категорий весов двух грузов. Вычислить общий вес этих грузов. Если номер хоть одной из них некорректный, вместо веса вывести сообщение об этом.

Вариант 11

1. Даны три действительных числа x, y, z . Составить программу, вычисляющую наибольшее из $|x-y|, |y-z|$ и $|z-x|$.
2. Задано целое число. Диапазон от 10 до 20. Выведите его представление в двоичном коде. К примеру, 12 – «1100». Если задано некорректное число – вывести сообщение об этом.

Вариант 12

1. Даны три действительных числа x, y, z . Составить программу, вычисляющую минимальное из $x, y+z$ и xz .
2. Задан номер. Выведите ФИО студента вашей подгруппы под этим номером. Если задано некорректное число – вывести сообщение об этом.

Вариант 13

1. Даны три действительных числа x, y, z . Составить программу, вычисляющую:
$$r = (x+y+z)/(xyz), \text{ если } x, y, z \neq 0, \text{ и } r = x+y+z \text{ в противном случае.}$$
2. Задано число – час. Диапазон от 1-24. Выведите соответствующее сообщение этому времени. К примеру, 3 – «3 часа ночи», 12 – «Полдень», 20 – «8 часов вечера». Если задано некорректное число – вывести сообщение об этом.

Вариант 14

1. Даны три действительных числа x, y, z . Если $x \neq 0$, то вывести y/x и z/x . В противном случае если $y \neq 0$, вывести z/y , иначе вывести z .
2. Задано целое число – это номер. Вывести символ английского алфавита под заданным номером. Если задано некорректное число – вывести сообщение об этом.