**18,09.23 Программирование разветвлений (продолжение)**

// **Пример 6** (Д.З. на 11.09.23)**.** Ввести три числа **x, y, z.** Найти среди них число

// с максимальным значением.

#include <iostream>

using namespace std;

int **main()**{

int x, y, z, max;

cout<<"Enter x, y, z: "; cin>>x>>y>>z;

cout<<**"Variant 1 "**;

if(x > y) max = x; else max = y;

if(z > max) max = z;

cout<<"max = "<<max<<endl;

cout<<**"Variant 2 "**;

if(x > y)

if(x > z) max = x; else max = z;

else

if(y > z) max = y; else max = z;

cout<<"max = "<<max<<endl;

cout<<**"Variant 3 "**;

max = x;

if(y > max) max = y;

if(z > max) max = z;

cout<<"max = "<<max<<endl;

cout<<**"Variant 4 "**;

if(x > y && x > z) max = x;

else if(y > z) max = y;

else max = z;

cout<<"max = "<<max<<endl;

}

**Результат:**

Enter x, y, z: **4 5 2**

Variant 1 max = 5

Variant 2 max = 5

Variant 3 max = 5

Variant 4 max = 5

// **Пример 7**. (Д.З. на 11.09.23) Ввести три числа **x, y, z**.

// Найти среди них максимальное число и два других.

#include <iostream>

using namespace std;

int **main()** {

int x, y, z, max, m1, m2;

cout << "Enter x, y, z: "; cin >> x >> y >> z;

max = x;

if (y > max) max = y;

if (z > max) max = z;

if (max == x) { m1 = y; m2 = z; }

else

if (max == y) { m1 = x; m2 = z; }

else { m1 = x; m2 = y; }

cout << "max = " << max << " m1 = " << m1 << " m2 = " << m2 << endl;

}

**Результат:**

Enter x, y, z: 3 4 5

max = 5 m1 = 3 m2 = 4

**Конструкция** **if..else..if** (*конструкция*-*лестница*)**.** Применяется, когда встречаются задачи, в которыхвыбор следует сделать между более чем двумя возможными вариантами, каждому из которых соответствует свое *условие*. Её можно представить в виде:

**if** (*условие\_*1) оператор\_1;

**else** **if** (*условие*2) оператор\_2;

**else** **if** (*условие*3) оператор\_3; **…**

**else** **if** (*условие*N) оператор\_N;

**else** // необязательная часть

Если окажется, что ни одно из условий не является истинным, будет выполнен последний элемент ***else***.

// **Пример 3.** Написать программу-калькулятор, которая может выполнять

// только **4** действия:  **+, -, \*, /** над значениями переменных **a** и **b**.

#include <iostream>

using namespace std;

int **main()**{

double a, b, rez;

char oper;

cout<<"Enter a and b: "; cin>>a>>b; // ввод, например, **5 6**

cout<<"Input the operation: ";

cin>>oper; // ввод, например, **\***

if(oper == '+'){rez = a + b; cout<<'\n'<<a<<" + "<<b<<" = "<<rez<<endl;}

else

if(oper == '-'){rez = a - b; cout<<'\n'<<a<<" - " <<b<<" = "<<rez<<endl;}

else

if(oper == '\*'){rez = a \* b; cout<<'\n'<<a<<" \* "<<b<<" = "<<rez<<endl;}

else

if(oper == '/')

if(b != 0){rez = a / b; cout<<"\n\t"<<a<<" / "<<b<<" = "<<rez<<endl;}

else cout<<"Division by zero!\n";

if(oper != '+' && oper != '-' && oper != '\*' && oper != '/')

cout<<"Incorrect operation!\n";

}

**Результат:**

Enter a and b: **5 6**

Input the operation: \*

5 \* 6 = 30

1. **Условная (тернарная) операция (?)**

Позволяет сократить определение простейших условных операторов **if** и имеет следующую форму:

*условие* ( операнд\_1) **?** операнд\_2 **:** операнд\_3

Операция использует сразу *три операнда*. Если *условие* **true** (т. е. **истинно**), то возвращается **операнд\_2**, если условие **false** (т. е. **ложно**), то возвращается **операнд\_3**.

**// Пример 4.** Использование условной операции **?**.

#include <iostream>

using namespace std;

int **main()**{

**setlocale(LC\_ALL, "");** // или **setlocale**(LC\_ALL, "russian");

int x = 7, y = 5;

int max = (x > y) ? x : y;

cout<<"max = "<<max<<endl;

char sign;

cout << "Введите операцию + или -: "; cin >> sign;

int result = (sign == '+') **?** (x + y) : (x - y);

cout << " result = " << result << '\n';

}

**Результат:**

max = 7

Введите операцию + или -: -

result = 2

1. **Оператор выбора** **switch**

Условные операторы позволяют легко осуществить выбор между двумя вариантами. Однако иногда возникает необходимость выбрать один вариант из нескольких. Это можно сделать, используя *вложенные условные* операторы **if**, но при этом программа становится более сложной, то есть менее доступной для понимания и анализа.

Удобным средством в этом случае является *оператор* **switch**, формат записи которого имеет следующий вид:

**switch** (*выражение*){

**case** константа\_1: оператор\_1; **break**;

**case** константа\_2: оператор\_2; **break**;

…

**case** константа\_N: оператор\_N; **break**;

**[default**: оператор; **break;]**

}

Оператор **switch** может выполнять операции проверки только *строгого равенства*.

**// Пример 5.** Проанализировать значение переменной **mark**, **которая является**

**// значением выставленной оценки.**

#include <iostream>

using namespace std;

int **main()**{

setlocale(LC\_ALL, ""); // **setlocale**(LC\_ALL, "russian");

int mark;

cout<<"Введи оценку: "; cin>> mark;

**switch**(mark){

case 5: cout<<"оценка отлично \n"; break;

case 4: cout<<" оценка хорошо\n"; break;

case 3: cout<<" оценка три\n"; break;

default: cout<<"Ошибка\n";

}

}

**Результат:**

Введите оценку: **4**

оценка хорошо

В одной метке **case** нельзя указать *диапазон значений*, то есть, если для двух меток необходимо выполнить одни и те же действия, придётся записать отдельные метки ***case***, но одну сделать "**пустой**":

**switch**(m){

case 1:

case 2: cout<<"\*\*\*\*\*"; break;

}

**// Пример6.** Выполнить проверку, попадает ли введённое значение

// переменной **х** в заданный диапазон от **1** до **4**. Вывести сообщение.

// Использовать вложенные операторы **if (**лесенку**)**.

// **Пример 7.** Ввести **a и b.** Решить уравнение **ax = b**. Предусмотреть все случаи.

// Нарисовать схему и написать программу.

**// Пример 8.** Ввести числа **a, b, с, d.**Расположить их значения в порядке

// возрастания.

// **Пример 9.** Составить программу-калькулятор для выполнения**4** действий:

// **+, -, \*, /** над значениями переменных **a** и **b**.. Использовать оператор **switch**.