

Практическое занятие 11

Тема: «Оператор switch»

Теоретическая часть

Альтернативой управляющей конструкции if-else может служить оператор ветвления switch. Он имеет следующий синтаксис:

```
switch(выражение)
{
  case константное_выражение :
    группа операторов;
  break;
  case константное_выражение :
    группа операторов;
  break;
  default константное_выражение :
    группа операторов;
}
```

Конструкция switch-case представляет собой своеобразный переключатель, который работает следующим образом.

В начале выполняется анализ **выражения**, которое задано в операторе switch. Далее осуществляется переход к той ветви программы, для которой его значение совпадает с указанным константным_выражением. После этого выполняются операторы этой ветви до тех пор, пока не встретится ключевое слово break, что приведет к выходу из тела оператора switch-case.

Если значение **выражения** не совпадет ни с одним из случаев константных_выражений, то выполнится ветвь программы, определенная с помощью ключевого слова default. Ключевые слова default и break не являются обязательными и зачастую их умышленно опускают. Подобный прием следует применять с осторожностью.

Пример 1. Приведенный ниже фрагмент иллюстрирует работу конструкции switch-case. Объявленная символьная переменная Answer служит для приема из входного потока ответа пользователя на вопрос о продолжении работы с программой.

```
#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
  char Answer;
  cout << "Do you want to continue? ";
  cin >> Answer;
```

```

switch(Answer)
{
    case 'y':
    case 'Y':
    case 'д':
    case 'Д':
        cout << "To be continue...\n";
        break;
    default:
        cout << "Close \n";
        cout << "And exit...\n";
        return 0;
}
cin.get();
return EXIT_SUCCESS;
}

```

Как видно из примера, при выборе любой из букв (у, У, д, Д) будет выполнен оператор `cout<<"To be continue...\n"`; и далее будет осуществлен переход к `cin.get()`. Если будет нажата любая другая клавиша, то переход будет осуществлен к метке **default:**, так как ни с одним из значений (у, У, д, Д) не будет совпадения. Группа операторов при метке **default:** завершается оператором **return 0;** т. е. программа будет завершена.

Если в программе не указывать оператор `break`, то на экран будут выведены сообщения обеих частей и программа завершится.

Оператор **switch** допускает вложения в себя аналогичных операторов, но следует избегать подобных конструкций, так как код становится визуально плохо воспринимаемым.

Пример 2. Наиболее эффективно применение связки операторов `switch-case` в программе с перечислениями. В приведенном ниже фрагменте объявляется перечисление цветов гаммы Rainbow и соответствующая этому типу переменная `Color`. Далее осуществляется сравнение значения переменной `Color` с заданными значениями цветов.

```

#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
    enum Rainbow{Red,Orange, Yellow,Green,Blue,Cyan,Magenta};
    Rainbow Color;
    // Вывод значений констант перечисления
    cout << "Red =" << Red << endl;
    cout << "Orange =" << Orange << endl;
    cout << "Yellow =" << Yellow << endl;
    cout << "Green =" << Green << endl;
    cout << "Blue =" << Blue << endl;
    cout << "Cyan =" << Cyan << endl;
    cout << "Magenta =" << Magenta << endl;
    // Задание цвета
    Color=Green;
    switch(Color)
    {
        case Red:

```

```

        case Orange:
        case Yellow:
            cout << "Выбрана теплая гамма" << endl;
            break;
        case Green:
        case Blue:
        case Cyan:
        case Magenta:
            cout << "Выбрана холодная гамма" << endl;
            break;
        default:
            cout << "Радуга не имеет такого ";
            cout << "цвета!\n";
    }
    cin.get();
    return EXIT_SUCCESS;
}

```

В приведенной программе используется функция перечисления `enum`, при помощи которой задается имя перечисления и значения констант перечисления. Имя перечисления используется в последующем для объявления. Значения констант перечисления являются целыми числами и, как правило, начинаются со значения 0. В программе выводятся значения этих констант.

Если на момент проверки переменная `Color` примет значение, соответствующее красному (Red), оранжевому (Orange) или желтому (Yellow) цвету, будет выведено сообщение "Выбрана теплая гамма". Если проверяемая переменная будет соответствовать зеленому (Green), синему (Blue), голубому (Cyan) или фиолетовому (Magenta) цвету, будет отображено сообщение "Выбрана холодная гамма". Если же проверяемое значение переменной `Color` не совпадет ни с одним из вышеперечисленных, будет выведена строка "Радуга не имеет такого цвета!".

Пример 3. Условие задачи: написать программу, которая складывает, вычитает, умножает, делит два числа введенных с клавиатуры. Операция выбирается пользователем.

```

#include <iostream>

using namespace std;

int main(int argc, char* argv[])
{
    int count; // переменная для выбора в switch
    double a,b; // переменные для хранения операндов
    cout << "Vvedite pervoe chislo: ";
    cin >> a;
    cout << "Vvedite vtoroe chislo: ";
    cin >> b;
    cout << "Vibirite deistvie: 1-clojenie; 2-vichitanie; 3-ymnojenie; 4-
delenie: ";
    cin >> count;
    switch (count) // начало оператора switch
    {
        case 1: // если count = 1
        {
            cout << a << " + " << b << " = " << a + b << endl; // сложение

```

```
break;
}
case 2: // если count = 2
{
cout << a << " - " << b << " = " << a - b << endl; // вычитание
break;
}
case 3: // если count = 3
{
cout << a << " * " << b << " = " << a * b << endl; // умножение
break;
}
case 4: // если count = 4
{
cout << a << " / " << b << " = " << a / b << endl; // деление
break;
}
default: // если count равно любому другому значению
cout << "Error" << endl;
}
return 0;
}
```

Практическая часть

1. Реализуйте приведенные в теоретической части примеры. Сделайте скрин-шоты их работы.
2. Напишите программу, которая по вводу порядкового номера дня недели выводит расписание на этот день. В программе используйте оператора switch(). Поясните работу программы. Приведите скрин-шоты ее работы.
3. Подготовьте отчет.