

## Оператор за многовариантен избор `switch`

### Синтаксис:

```
switch (<израз>
{
    case <израз_1>:<оператор_1>;<оператор_2>;...<оператор_N>; [break;]
    case <израз_2>:<оператор_1>;<оператор_2>;...<оператор_N>; [break;]
    ...
    case <израз_n>:<оператор_1>;<оператор_2>;...<оператор_N>; [break;]
    [default:<оператор_1>;<оператор_2>;...<оператор_N>;]
}
```

където:

<оператор\_1>, <оператор\_2>, ... <оператор\_N> са произволни оператори, допустими за езика;

<израз> е израз (може и променлива) от дискретен тип (целочислен, символен, булев). Изразът не може да бъде от реален тип. Нарича се още израз-селектор.

<израз\_1>, <израз\_2>, ..., <израз\_n> са константни изрази, т. е. стойностите им не зависят от входните данни. Наричат се етикети. Стойностите на етикетите са различни.

- Семантика:**
1. Изчислява се стойността на израза.
  2. Тази стойност се сравнява последователно със стойностите на етикетите <израз\_1>, <израз\_2>, ..., <израз\_n>. При съвпадение на стойностите се изпълнява редицата от оператори на съответния вариант и редицата от оператори на всички варианти, разположени след него, до срещане на оператор **break**.
  3. Ако стойността на израза не съвпада с нито една от стойностите на етикетите, тогава се изпълнява операторът след етикета **default** до срещане на оператора **break**.
  4. Етикетът **default** не е задължителен и ако липсва, не следват никакви действия от оператор **switch**.

## 2. Оператор `break`

**Синтаксис:** `break;`

**Семантика:** Прекратява изпълнението на най-вътрешния, съдържащ го оператор `switch`. Изпълнението на програмата продължава с оператора, следващ прекъснатия.

Чрез оператор за многовариантен избор `switch` поставената в началото задача 1 може да бъде решена по-кратко и разбираемо.

**Решение:**

```
//Program_1b.cpp;
#include<iostream.h>
int main()
{
    int day;
    cin>>day;
```

```

switch (day)
{
    case 1 : cout<< "Понеделник\n";break;
    case 2 : cout<< "Вторник\n";break;
    case 3 : cout<< "Сряда\n";break;
    case 4 : cout<< "Четвъртък\n";break;
    case 5 : cout<< "Петък\n";break;
    case 6 : cout<< "Събота\n";break;
    case 7 : cout<< "Неделя\n";break;
    default : cout<<"Няма такъв ден\n";
}
return 0;
}

```

Нека решим още една задача, чрез която да илюстрираме използването на оператора за многовариантен избор със символен тип данни.

**Задача 2:** Да се състави програма, която въвежда две числа  $a$  и  $b$  и знак за операция събиране, изваждане, умножение или деление. Според зададената операция програмата да извършва съответното действие с двете числа  $a$  и  $b$ , и да извежда резултат. На входа се задават последователно на един ред разделени с интервал, двете числа  $a$  и  $b$  и операцията. На изхода се извежда резултатът от операцията или подходящо съобщение.

**Решение:**

```

//Program 2.cpp;
#include<iostream.h>
int main()
{
    double a,b;
    cin>>a>>b;
    char c;
    cin>>c;
    switch(c)
    {
        case '+':cout<<a+b<<'\n';break;
        case '-':cout<<a-b<<'\n';break;
        case '*':cout<<a*b<<'\n';break;
        case '/':if(b!=0) cout<<a/b<<'\n';
                else cout<<"Деление на нула!\n";break;
        default :cout<<"Грешна операция!\n";
    }
    return 0;
}

```

**Примерен вход:**

2 5 +

**Примерен изход:**

10

**Примерен вход:**

8 -3 -

**Примерен изход:**

11

**Примерен вход:**

2 3 %

**Примерен изход:**

Грешна операция

*Да се изпробва дадената задача, като се премахне операторът **break**.*