

Материал для подготовки к семинару 3

- Переключатели;
- Параметрический цикл (for);
- Методы с передачей параметров по значению

Оператор безусловного перехода (goto)

Помеченные операторы

Метка - уникальный идентификатор, который может быть помещен перед каждым оператором.

Метка отделяется от оператора двоеточием.

Оператор безусловного перехода

goto <метка>;

Удобно использовать для выхода из глубоко вложенных циклов.

Важно:

Метки внутри блока недостижимы для оператора перехода, размещённого вне блока (метки локализованы в блоке).

Переключатель (switch)

```
switch (переключающее_выражение) {  
    case константное_выражение_1:  
        операторы_1; завершение;  
    case константное_выражение_2:  
        операторы_2; завершение;  
    ...  
    case константное_выражение_n:  
        операторы_n; завершение;  
    default:  
        операторы_default; завершение;  
}
```

ОПЦИОНАЛЬНО

константное_выражение = целое число | char | string | bool |
элемент перечисления

завершение = *break* | goto i | goto default | return | throw

Переключатель (пример)

```
int ball=4; // оценка в баллах [1, 10]:
switch (ball)
{
    case 1:
    case 2:
    case 3:
        Console.WriteLine("Неудовлетворительно");
        break;
    case 4:
    case 5:
        Console.WriteLine("Удовлетворительно");
        break;
    case 6:
    case 7:
        Console.WriteLine("Хорошо");
        break;
    case 8:
    case 9:
    case 10:
        Console.WriteLine("Отлично");
        break;
    default: Console.WriteLine("Ошибка в данных");
        break;
} // Конец переключателя
```

Дополнения в switch (C# 7)

1) Расширение переключающего выражения:

`switch (переключающее_выражение) {...}`

`переключающее_выражение` – *не тождественное null выражение*

2) Использование типов в case (type pattern):

`case type varname`

например:

`case Rectangle r:`

`case null: // для null значений`

3) Использование предложения when в case:

`case Square sq when sq.Area > 10:`

4) Важен порядок в case.

Параметрический цикл (for)

for (*инициализатор; условие; заверш._выр.*)
 тело_цикла

//

инициализатор;
while (*условие*) {
 тело_цикла
 заверш._выр.
}

Параметрический цикл (пример)

1) Описываем и инициализируем переменную limit

2) Описываем и инициализируем переменную i

3) Проверяем условие выполнения тела цикла

```
int limit = 15;  
for (int i = 0; i < limit; i++)  
{  
    Console.WriteLine(i + " ");  
}
```

6) Переходим к п.3 (проверяем условие)

4) Выполняем тело цикла

5) Изменяем значение управляющей переменной

Передача параметров по значению

Параметр передаётся по значению

```
public static void Foo(string st) {  
    st = "Hello from Foo";  
    Console.WriteLine(st);  
}
```

Это изменение локально для **Foo()**

```
static void Main(string[] args) {  
    string str = "Test string";  
    Console.WriteLine(str);  
    Foo(str);  
    Console.WriteLine(str);  
}
```

Значение переменной **str** не изменилось

```
// Test string  
// Hello from Foo  
// Test string
```


Передача параметров по значению

```
class Program {  
    static double Summ(double a, double b) {  
        a = 2 * a;  
        b = 2 * b;  
        return a + b;  
    }  
    static void Main() {  
        double a = 1.44,  
        b = 0.25;  
        double sumDb1 = Summ(a, b);  
        System.Console.Write("a = {0}; b = {1}; sumDb1 = {2}",  
a, b, sumDb1);  
    }  
}
```

параметры

аргументы