

Основы программирования

# Синтаксис языка C#

**Операторы**  
**Операторы цикла**

# Операторы цикла

Используются для вычислений, повторяющихся многократно:

- `for;`
- `while;`
- `do while;`
- `foreach.`

# Операторы цикла. Термины

*Тело цикла* – повторяющийся блок операторов.

*Итерация* – один проход цикла.

*Параметр цикла* – называется счетчик цикла или переменная, влияющая на условие продолжения выполнения цикла.

*Условие продолжения цикла* – выполняется на каждой итерации либо до, либо после тела цикла.

# Цикл с предусловием while

```
while (логическое_выражение)
{
    тело_цикла;
}
```

# Цикл while. Корректная работа

- объявить и инициализировать параметры цикла (до описания тела цикла);
- проверить корректность условия продолжения выполнения цикла;
- модифицировать параметры цикла при выполнении тела цикла.

# Цикл while. Меню

```
Console.WriteLine("1-задача 1, 2-задача 2, 0- выход");
string buf = Console.ReadLine();
//инициализация параметра цикла
while (buf != "0")
{
    switch (buf)
    {
        case "1": блок1; break;
        case "2": блок2; break;
        default: Console.WriteLine("Неверный пункт"); break;
    }
    Console.WriteLine("1-задача 1, 2-задача 2, 0-выход");
    buf = Console.ReadLine(); //изменение параметра цикла
}
```

# Цикл с постусловием do while

```
do
{
    тело_цикла;
}
while (логическое_выражение) ;
```

# Цикл do while. Пример

```
string buf = null; //объявление и инициализация параметра
    цикла
do
{
    Console.WriteLine("1 - задача 1, 2 - задача 2, 0 -
    выход");
    buf = Console.ReadLine();           //ввод пункта меню
    пользователем
    switch (buf)
    {
        case "1": блок1; break;
        case "2": блок2; break;
        case "0": break;
        default: Console.WriteLine("Неверный пункт меню");
        break;
    }
}
while (buf != "0");
```



# Цикл с предусловием for

```
for (инициализации_счетчика;  
     логическое_выражение;  
     модификация_счетчика)  
{  
    тело_цикла;  
}
```

Счетчик необязательно должен представлять тип `int`. Это может быть и другой числовой тип, например, `double`.

# Порядок выполнения цикла for

1. Задаются значения переменным, объявленным и проинициализированным в секции инициализации;
2. Вычисляется результат логического выражения. Если выражение истинно – выполняется тело цикла, если ложно – цикл завершается;
3. После выполнения тела цикла исполняется секция изменения. Затем переход к пункту 2.

# Цикл for. Пример

```
Console.Write("Введите xmin:");
double xmin = Double.Parse(Console.ReadLine());
Console.Write("Введите xmax:");
double xmax = Double.Parse(Console.ReadLine());
Console.Write("Введите шаг табуляции dx:");
double dx = Double.Parse(Console.ReadLine());
Console.WriteLine("|---x---|---y---|");

for (double x = xmin; x <= xmax; x +=dx)
    // табуляция функции
{
    Console.WriteLine("|{0,7}|{1,7}|", x, Math.Sin(x));
}
```

# Цикл for. Вариация 1 примера

В секции инициализации нет объявления переменных цикла:

```
double x;  
for (x = xmin; x < xmax; x +=dx)  
    // табуляция функции  
{  
Console.WriteLine("|{0,7}|{1,7}|", x, Math.Sin(x));  
}
```

# Цикл с for. Вариация 2 примера

Секция инициализации отсутствует:

```
double x = xmin;  
for (; x < xmax; x +=dx)  
    // табуляция функции  
{  
    Console.WriteLine("|{0,7}|{1,7}|", x, Math.Sin(x));  
}
```

# Цикл с for. Вариация 3 примера

Секции инициализации и изменения отсутствует:

```
double x = xmin;  
for (; x < xmax;)           // табуляция функции  
{  
    Console.WriteLine("|{0,7}|{1,7}|", x, Math.Sin(x));  
    x += dx;  
}
```

# Цикл с for. Вариация 4 примера

Тело цикла пустое, табуляция выполняется в секции изменения:

```
for (double x = xmin;  
    x < xmax;  
    Console.WriteLine( "{0,7}|{1,7}|", x,  
                        Math.Sin(x) ), x +=dx)  
{  
}
```

# Цикл с for. Без секций

Синтаксически корректная конструкция:

```
for ( ; ; )  
{  
}
```



# Цикл `foreach`

Будет рассмотрен далее при изучении массивов.