

# Nested Loops



# Контрольные вопросы

- Что такое цикл?
- Что такое итерация?
- Виды циклов в Java
- Что такое управляющая переменная?
- Что такое цикл с предусловием?
- Что такое цикл с постусловием?
- Что такое бесконечный цикл?

# Синтаксис цикла for

```
for (инициализация УП; условие; изменение УП) {  
    тело цикла;  
}
```

## Принцип выполнения цикла:

1. Инициализация управляющей переменной.
2. Проверка условия.
3. Выполнение действия, если условие истинно.
4. Если условие ложно, выполнение следующего за циклом оператора.
5. Если условие было истинно - изменение управляющей переменной.
6. Проверка условия. Далее снова пункт 3 или 4.

# Схема работы цикла for



# Простой пример кода

```
for (int i = 0; i <= 5; i++) {  
    System.out.print(i + " ");  
}
```

Внутри цикла объявляется переменная *i*, равная 0. Это и будет управляющая переменная. Затем, осуществляется проверка значения этой переменной с помощью условия *i*≤5; Если условие истинно (а так будет, пока *i* не достигнет значения 6), выполняется показ значения *i* на экран, и изменение управляющей переменной *i* на 1 (*i*++). Затем, снова проверяется условие. Если условие ложно (то есть значение *i* стало равно 6), то программа переходит на следующую строчку за закрывающейся фигурной скобкой цикла.

**Практика:**

Показать целые числа от 10 до 0.

# Особенности цикла for

Создание УП до цикла:

```
int i;  
for (i = 0; i <= 5; i++) {  
    System.out.println(i);  
}
```

# Особенности цикла for

Создание и инициализация управляющей переменной до цикла:

```
int i = 0;  
for (; i <= 5; i++) {  
    System.out.println(i);  
}
```

# Особенности цикла for

Изменение УП в теле цикла:

```
for (int i = 0; i <= 5; ) {  
    System.out.println(i);  
    i++;  
}
```



# Особенности цикла for

Условие можно пропустить (создаётся бесконечный цикл):

```
for (int i = 0; ; i++) {  
    System.out.println(i);  
}
```

Остановить такое под силу только **break** 😊

# Особенности цикла for

Ни одна из частей цикла не является обязательной:

```
int i = 0;  
for (;;) {  
    System.out.println(i++);  
    if (i > 10) break;  
}
```

# Особенности цикла for

А как вам такое?

```
for (float c = 0, f = 32; c <= 100; c++, f+=1.8) {  
    System.out.println(c + " - " + f);  
}
```

Или даже так:

```
for (float c = 0, f = 32; c <= 100 && f <= 200;  
    System.out.println(c + " - " + f), c++, f+=1.8);
```

# Оператор break

Нередко при работе с циклами возникает необходимость искусственно прервать выполнение цикла. Для этого используется уже знакомый вам (по изучению **switch**) оператор **break**. Этот оператор должен находиться в теле цикла, в том месте, где необходимо сделать остановку. Например, именно с помощью этого оператора, можно решить проблему бесконечного цикла, в ситуации, когда условие в цикле **for** не указывается. Рассмотрим пример:

```
for (int x = 0; x <= 10; x++) {  
    if (x == 5) break;  
    System.out.println(x + ", ");  
}
```

// break может быть использован либо в циклах, либо в операторе switch.  
любое другое размещение приводит к ошибке на этапе компиляции!



# Оператор continue

Оператор **continue** используется для прерывания текущей итерации цикла и осуществления перехода на следующий шаг. В ряде случаев, такие действия являются необходимыми. Если выполняется оператор **continue**, то в зависимости от вида цикла происходит следующее:

- Циклы **while** и **do while** останавливают выполнение шага и переходят к проверке условия.
- Цикл **for** также останавливает выполнение шага. Но, сначала переходит к изменению управляющей переменной, а потом уже к проверке условия.

```
for (int x = 5; x < 20; x++) {  
    if (x % 2 == 0) continue;  
    System.out.println(x);  
}
```



BREAK



CONTINUE

# Рисуем линию из звёзд

```
int count = 10;
for (int stars = 0; stars < count; stars++)
{
    System.out.print("*");
}
System.out.println();
```

# Рисуем прямоугольник из звёзд

```
int height = 5, count = 10;
for (int lines = 0; lines < height; lines++) {
    for (int stars = 0; stars < count; stars++) {
        System.out.print("*");
    }
    System.out.println();
}
System.out.println();
```

# Нужно больше семантики

```
int height = 5;  
int width = 10;  
for (int y = 0; y < height; y++) {  
    for (int x = 0; x < width; x++) {  
        System.out.print("*");  
    }  
    System.out.println();  
}
```





# Домашнее задание

с\л. Ёлочка!

```
  e
  eee
 eeeee
 eeeeeee
 eeeeeeeee
  eee
  eeee
 eeeeeee
 eeeeeeeee
 eeeeeeeeeeee
  eeee
  eeeeeee
 eeeeeeeee
 eeeeeeeeeeee
 eeeeeeeeeeee
 eeeeeeeeeeee
  eeeee
  eeeeeee
 eeeeeeeeeeee
 eeeeeeeeeeee
  eeeee
  eeeeeee
 eeeeeeeeeeee
 eeeeeeeeeeee
 eeeeeeeeeeee
 eeeeeeeeeeee
  eeee
  eeee
  eeee
  eeee
```

Для продолжения нажмите любую клавишу . . .