

# Практическая работа №5

## Использование операторов цикла при решении вычислительных задач

Циклы позволяют в зависимости от определенных условий выполнять определенное действие множество раз. В языке Java есть три оператора циклов: for (цикл с параметром), while (цикл с предусловием), do...while (цикл с постусловием).

Рассмотрим стандартный цикл for:

```
for (int i = 1; i < 10; i++){  
    System.out.printf("Квадрат числа %d равен %d \n", i, i * i);  
}
```

Первая часть объявления цикла `int i = 1` создает и инициализирует счетчик `i`. В данном случае это то же самое, что и объявление переменной. Счетчик необязательно должен представлять тип `int` - это может быть и любой другой числовой тип, например, `double`. Перед выполнением цикла значение счетчика будет равно 1. Вторая часть - условие, при котором будет выполняться цикл. В данном случае цикл будет выполняться, пока `i` не достигнет 10. И третья часть - приращение счетчика на единицу. В итоге блок цикла сработает 9 раз, пока значение `i` не станет равным 10.

Необязательно указывать все условия при объявлении цикла – ряд блоков можно опустить. Например, аналогичный цикл мы можем написать так:

```
int i = 0;  
for (; i++ < 9; ) {  
    System.out.printf("Квадрат числа %d равен %d \n", i, i * i);  
}
```

У нас по-прежнему есть счетчик, только создан он вне цикла, и есть приращение счетчика, но уже в блоке условия выполнения цикла.

Цикл for может определять сразу несколько переменных и управлять ими:

```
int n = 10;  
for (int i=0, j = n - 1; i < j; i++, j--){  
    System.out.println(i * j);  
}
```

Цикл while проверяет истинность некоторого условия, и если условие истинно, то код цикла выполняется:

```
int i = 1;  
while (i < 10) {  
    System.out.printf("Квадрат числа %d равен %d \n", i, i * i);  
    i++;  
}
```

Следует помнить, что цикл `while` будет выполнен только в том случае, если на момент начала его выполнения логическое выражение будет иметь значение `true`. В противном случае тело цикла с предусловием не будет выполнено ни разу.

Цикл `do` сначала выполняет код цикла, а потом проверяет условие в инструкции `while`. Цикл повторяется, пока это условие истинно:

```
int i = 1;
do {
    System.out.printf("Квадрат числа %d равен %d \n", i, i * i);
    i++;
}
while (i < 10);
```

Важно отметить, что цикл `do` гарантирует хотя бы однократное выполнение действий, даже если условие в инструкции `while` не будет истинно.

Оператор `break` позволяет выйти из цикла в любой его момент, даже если цикл не закончил свою работу. Когда счетчик станет равным 5, сработает оператор `break`, и цикл завершится:

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

Теперь сделаем так, чтобы если число равно 5, цикл не завершался, а просто переходил к следующей итерации. Для этого используем оператор `continue`:

```
for (int i = 0; i < 10; i++){
    if (i == 5)
        continue;
    System.out.println(i);
}
```

В этом случае, когда выполнение цикла дойдет до числа 5, программа просто пропустит это число и перейдет к следующему.

Также допустимо использование вложенных циклов, когда один цикл выполняется внутри другого. Рассмотрим программу, выводящую на экран таблицу умножения:

```
for (int i = 1; i < 10; i++) {  
    for (int j = 1; j < 10; j++) {  
        System.out.printf("%dx%d=%d\t", i, j, i * j);  
    }  
    System.out.println();  
}
```

При вложенных циклах количество итераций (повторений тела цикла) умножается. В данном примере два вложенных цикла, в каждом из которых по 9 итераций, формируют таблицу из 81 значения.

## Практические задания

1. Начав тренировки, лыжник в первый день пробежал 10 км. Каждый следующий день он увеличивал пробег на 10% от пробега предыдущего дня. Составить программу, позволяющую определить:
  - а) пробег лыжника за второй, третий, ..., десятый день тренировок;
  - б) суммарный путь, который он пробежал за первые десять дней тренировок.
2. Составить программу, позволяющую вычислить:
  - а) среднее арифметическое всех целых чисел от 1 до 1000;
  - б) среднее арифметическое всех целых чисел от 1 до  $n$  (значение  $n$  вводится с клавиатуры).
3. Выяснить, является ли совершенным введённое с клавиатуры натуральное число. Натуральное число называется совершенным, если оно равно сумме своих делителей, включая 1 и исключая это самое число. Например, число 6 — совершенное ( $6 = 1 + 2 + 3$ ).
4. Составить программу, позволяющую найти и вывести на экран 10 первых натуральных чисел, больших 100, оканчивающихся на цифру 7 и кратных числу 9.
5. Среди квадратов натуральных чисел (значений 1, 4, 9, 16, 25, ...) найти первое число, большее введённого с клавиатуры целого числа  $n$ .