

Циклы for

Циклы for стали логичным продолжением циклов while.

Разделим программный код реализации цикла while на структурные блоки следующим образом:

```
{блок_инициализации}  
while (условие) {  
    тело_цикла  
    {блок_пост_операций}  
}
```

Чтобы было понятнее, разберём конкретную задачу. Пусть на вход подаётся число n, после чего ещё n чисел. Требуется посчитать сумму этих чисел.

```
import java.util.Scanner;  
public class Main {  
  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        int n = sc.nextInt();  
        int sum = 0;  
        int i=0;  
        while (i<n){  
            int a = sc.nextInt();  
            sum += a;  
            i++;  
        }  
        System.out.println(sum);  
    }  
}
```

Для решения этой задачи, мы ввели, так называемый счётчик i. Его используют для подсчёта количества выполнений тела цикла. Для этого перед циклом мы создали и инициализировали счётчик значением 0. После чего в цикле считываем новые значения, прибавляем их к переменной, в которой будет храниться сумма. В качестве пост. Операции мы просто увеличиваем значение счётчика на 1 каждый раз, когда выполняется тело цикла.

Перепишем эту программу через цикл for:

```
import java.util.Scanner;  
public class Main {  
  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        int n = sc.nextInt();  
        int sum = 0;  
        for (int i=0; i<n; i++){  
            int a = sc.nextInt();  
            sum += a;  
        }  
        System.out.println(sum);  
    }  
}
```

Как можно заметить такое использование цикла гораздо более компактное и ёмкое. Очень важно понимать, что цикл for довольно мощный инструмент. Общий вид цикла for выглядит следующим образом:

```
for (инициализация; условие; пост_операции) {  
    тело_цикла  
}
```

В качестве инициализации можно создать не просто одну переменную. Через запятую можно добавлять сколько угодно переменных:

```
for (int i=0, j=5; i<n; i++){  
}
```

Блок условий мы подробно рассматривать не будем, т.к. условия были уже рассмотрены в предыдущих уроках.

Наибольший интерес представляет блок пост-операций. Здесь через запятую мы можем писать сколь угодно сложные инструкции. Для наглядности рассмотрим решение следующей задачи:

Пусть на вход подаётся число n , после чего ещё n натуральных чисел. Требуется посчитать кол-во чётных.

```
import java.util.Scanner;  
  
public class Main {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        int n = sc.nextInt();  
        int sum = 0;  
        int cnt = 0;  
        for (int i=0, j=5; i<n; i++){  
            int a = sc.nextInt();  
            if (a%2==0){  
                cnt++;  
            }  
        }  
        System.out.println(cnt);  
    }  
}
```

Заметим, что при помощи тернарной операции можно записать этот код гораздо красивее:

```
import java.util.Scanner;  
  
public class Main {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        int n = sc.nextInt();  
        int sum = 0;  
        int cnt = 0;  
        for (int i=0, j=5; i<n; i++){  
            cnt += sc.nextInt()%2==0?1:0;  
        }  
        System.out.println(cnt);  
    }  
}
```

Но мы говорим о блоке пост. операций. Попробуем тело цикла перенести полностью в блок пост-операций.

```
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int sum = 0;
        int cnt = 0;
        for (int i=0, j=5; i<n; i++, cnt += sc.nextInt()%2==0?1:0) {
        }
        System.out.println(cnt);
    }
}
```

Такой код писать не стоит в силу плохой читаемости, но он хорошо показывает возможности цикла for.

В случае, если тело цикла пустое, вместо фигурных скобок, можно поставить точку с запятой.

```
for (int i=0, j=5; i<n; i++, cnt += sc.nextInt()%2==0?1:0);
```

Обратите внимание, что “;” после объявления цикла for означает, что цикл не имеет тела. Поэтому, если Вы пишете цикл for с непустым телом, нив коем случае не ставьте “;” после его объявления.