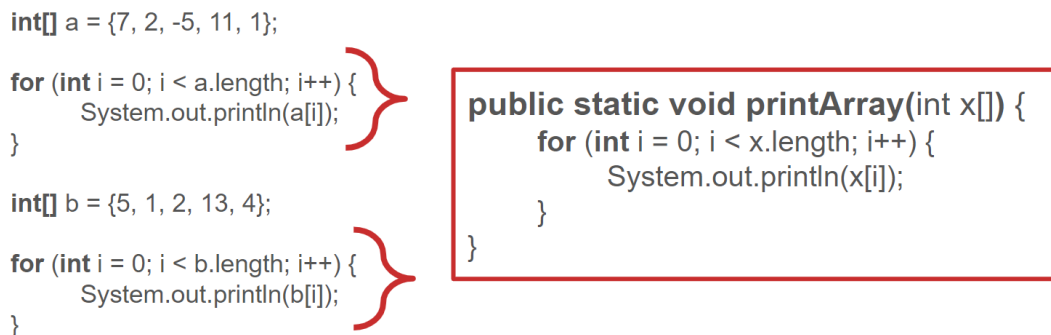


# Методы. Часть 1.

**Методы** в Java представляют собой подпрограммы, которые позволяют структурировать код, облегчая его понимание и поддержку. Они особенно полезны для выполнения повторяющихся задач в программе.

```
int[] a = {7, 2, -5, 11, 1};  
  
for (int i = 0; i < a.length; i++) {  
    System.out.println(a[i]);  
}  
  
int[] b = {5, 1, 2, 13, 4};  
  
for (int i = 0; i < b.length; i++) {  
    System.out.println(b[i]);  
}
```



```
public static void printArray(int x[]) {  
    for (int i = 0; i < x.length; i++) {  
        System.out.println(x[i]);  
    }  
}
```

**Сигнатура метода** в Java — это уникальная комбинация его имени и списка параметров.

- **void** указывает, что метод не возвращает значение (будет подробнее рассмотрено в следующих уроках). После выполнения метода не предоставляется никакой информации вызывающему коду, но имеется **побочный эффект** (например, вывод в консоль)
- **Название метода** — это идентификатор, используемый для его вызова.
- **Формальные параметры** - это переменные, которые определяются в скобках после названия метода. Они действуют как "приемники" значений, переданных в метод. Типы этих переменных должны соответствовать типу передаваемых значений.
- Внутри фигурных скобок находится **тело метода**, содержащее инструкции, которые определяют, что должен делать метод. Это может включать операции с параметрами, вызовы других методов и различные вычисления. В случае `void` метода, тело может выполнять операции и иметь побочный эффект.
- **public static** - ключевые слова, которые будут рассмотрены в следующих уроках.

```

        название  параметр
    public static void printArray(int x[]) {

        for (int i = 0; i < x.length; i++) {
            System.out.println(x[i]);
        }
    }

```

тело

**Вызов метода** — это процесс, при котором программа исполняет код, содержащийся внутри этого метода. Для вызова метода используется его имя, после которого в скобках указываются аргументы (если они требуются). Это имя должно совпадать с именем, заданным при определении метода.

Если метод определен с параметрами, при вызове необходимо передать аргументы, соответствующие по типу и порядку объявленным параметрам. Например, если метод определен как `void process(int x, String y)`, то при вызове необходимо передать целочисленное значение и строку: `process(5, "text")`.

```
int[] a = {7, 2, -5, 11, 1};
```

```
printArray(a);
```

```
int[] b = {5, 1, 2, 13, 4};
```

```
printArray(b);
```

