



Язык программирования Java. Начальный уровень

Мультимедийный курс

автор: Васильев А.Н.

www.vasilev.kiev.ua

Киев 2017



Лекция 5. Статические методы



- Статические и нестатические методы
- Описание статического метода
- Перегрузка методов
- Массив как аргумент и результат метода
- Механизм передачи аргументов методу
- Рекурсия
- Метод с произвольным количеством аргументов

- Это Жазель. Француженка. Я признал её. По ноге.

- Не-е, это не Жазель. Жазель была брюнетка, а эта вся белая. из к/ф "Формула любви"

Методы



- Метод - именованный блок программного кода. Этот блок может быть вызван по имени.
- При вызове метода выполняются команды, описанные в этом методе.
- При вызове методу могут передаваться аргументы.
- Методы описываются в классах.
- Методы делятся на статические и нестатические.
- Нестатический метод вызывается из объекта.
- Статическому методу объект не нужен.



Описание статического метода

Синтаксис

Признак
статического
метода

Тип результата,
возвращаемого
методом

Имя метода

Аргументы метода:
описываются в
формате
тип аргумент

```
static      тип      имя (аргументы) {  
    // команды  
}
```

Команды, выполняемые при
вызове метода

Если метод не возвращает
результат, то в качестве
идентификатора типа
используется ключевое
слово `void`

Описание статического метода



Примеры

```
static void show() {  
    System.out.println("Изучаем Java!");  
}
```

**Метод без аргументов,
не возвращает результат**

```
static int next(int n) {  
    return n+1;  
}
```

**Метод с целочисленным аргументом,
возвращает целочисленный результат**

```
static double sqr(double x) {  
    return x*x;  
}
```

**Метод с числовым аргументом,
возвращает числовой результат**

```
static void display(double x, int n) {  
    System.out.println("Первый аргумент: " + x);  
    System.out.println("Второй аргумент: " + n);  
}
```

**Метод с двумя
аргументами, не
возвращает
результат**

Упрощенная схема использования статического метода



```
// Главный класс (класс с главным методом) :  
class Класс{  
    // Описание статического метода в главном классе:  
    static тип метод(аргументы) {  
        // Код метода  
    }  
    // Главный метод:  
    public static void main(String[] args) {  
        // Команды  
        метод(аргументы) ; // Вызов метода  
        // Команды  
    }  
}
```



```
class StatMethDemo{
    // Метод для отображения текста:
    static void show(String txt){
        System.out.println(txt);
    }
    // Метод для вычисления факториала числа:
    static int factorial(int n){
        int s=1;
        for(int k=1;k<=n;k++){
            s*=k;
        }
        return s;
    }
    // Метод для возведения числа в степень:
    static double power(double x,int n){
        double s=1;
        for(int k=1;k<=n;k++){
            s*=x;
        }
        return s;
    }
    // Главный метод:
    public static void main(String[] args){
        // Вызываем методы:
        show("Начинаем вычисления:");
        int m=5;
        double z=3,p;
        show(m+"!="+factorial(m));
        p=power(z,m);
        show(z+" в степени "+m+": "+p);
    }
}
```

Результат:

Начинаем вычисления:

5!=120

3.0 в степени 5: 243.0

Перегрузка методов



- Перегрузка метода - создание нескольких версий метода.
- Все версии имеют одинаковое имя, но отличаются списком аргументов.
- Версия метода при вызове определяется по аргументам, которые переданы методу.

Например:

```
static void show(){  
    System.out.println("Нет аргументов");  
}
```

Метод без аргументов, не возвращает результат

```
static void show(int n){  
    System.out.println("int-аргумент: "+n);  
}
```

Метод с int-аргументом, не возвращает результат

Метод с char-аргументом

```
static void show(char s){  
    System.out.println("char-аргумент: "+s);  
}
```



```
class OverloadMethDemo{
    static void show(String txt){    // Метод для отображения текста
        System.out.println("Текст: "+txt);
    }
    static void show(int num){        // Метод для отображения целого числа
        System.out.println("Целое число: "+num);
    }
    // Метод для отображения действительного числа:
    static void show(double num){
        System.out.println("Действительное число: "+num);
    }
    static void show(char s){        // Метод для отображения символа
        System.out.println("Символ: "+s);
    }
    static void show(int num,char s){ // Метод для отображения числа и символа
        System.out.println("Аргументы "+num+" и "+s);
    }
    // Главный метод:
    public static void main(String[] args){
        // Переменные:
        int num=5;
        char symb='R';
        double z=3;
        // Вызываем метод:
        show(symb);
        show("Язык программирования Java");
        show(num);
        show(z);
        show(num, 'W');
    }
}
```

Результат:

Символ: R
 Текст: Язык программирования Java
 Целое число: 5
 Действительное число: 3.0
 Аргументы 5 и W

```
import java.util.Random;
class ArrayAsArgDemo{
    // Метод для заполнения массива:
    static void fillArray(int[] nums){
        Random rnd=new Random();
        for(int k=0;k<nums.length;k++){
            nums[k]=rnd.nextInt(100)+1;
        }
    }
    // Метод для отображения массива:
    static void show(int[] nums){
        for(int k=0;k<nums.length;k++){
            System.out.print("| "+nums[k]+" ");
        }
        System.out.println("|");
    }
    // Главный метод:
    public static void main(String[] args){
        // Массивы:
        int[] A={1,3,5,7,9,11,13,15};
        int[] B=new int[5];
        fillArray(B); // Массив B заполняется случайными числами
        System.out.println("Массив A:");
        show(A);      // Отображается массив A
        System.out.println("Массив B:");
        show(B);      // Отображается массив B
    }
}
```

Массив A:

| 1 | 3 | 5 | 7 | 9 | 11 | 13 | 15 |

Массив B:

| 80 | 95 | 51 | 29 | 48 |

```
import java.util.*;
class ArrayAsResDemo{
    // Метод для создания массива с числами Фибоначчи:
    static int[] fibs(int n){
        int[] nums=new int[n];
        nums[0]=1;
        if(nums.length==1) return nums;
        nums[1]=1;
        for(int k=2;k<nums.length;k++){
            nums[k]=nums[k-1]+nums[k-2];
        }
        return nums;
    }
    // Метод для создания массива со случайными символами:
    static char[] randS(int n){
        Random rnd=new Random();
        char[] syms=new char[n];
        for(int k=0;k<syms.length;k++){
            syms[k]=(char) ('A'+rnd.nextInt(26));
        }
        return syms;
    }
    // Главный метод:
    public static void main(String[] args){
        int[] A;        // Переменная для целочисленного массива
        char[] B;        // Переменная для символьного массива
        A=fibs(10);      // Создается массив с числами Фибоначчи
        System.out.println("Числа Фибоначчи:\n"+Arrays.toString(A));
        B=randS(8);      // Создается массив со случайными символами
        System.out.println("Случайные символы:\n"+Arrays.toString(B));
    }
}
```

Числа Фибоначчи:

[1, 1, 2, 3, 5, 8, 13, 21, 34, 55]

Случайные символы:

[B, T, O, V, C, S, N, H]

```
import java.util.Arrays;
class ArgsPassDemo{
    // Первый метод:
    static void alpha(int n){
        System.out.println("В методе alpha(). На входе: "+n);
        n++;
        System.out.println("В методе alpha(). На выходе: "+n);
    }
    // Второй метод:
    static void bravo(int[] n){
        String txt=Arrays.toString(n);
        System.out.println("В методе bravo(). На входе: "+txt);
        for(int k=0;k<n.length;k++){
            n[k]++;
        }
        txt=Arrays.toString(n);
        System.out.println("В методе bravo(). На выходе: "+txt);
    }
    // Третий метод:
    static void charlie(int[] n){
        String txt=Arrays.toString(n);
        System.out.println("В методе charlie(). На входе: "+txt);
        int[] m=new int[n.length];
        for(int k=0;k<n.length;k++){
            m[k]=n[k]+1;
        }
        n=m;
        txt=Arrays.toString(n);
        System.out.println("В методе charlie(). На выходе: "+txt);
    }
    // Продолжение кода - на следующем слайде!!!
}
```

// Главный метод:

```
public static void main(String[] args){
    int A=100;        // Переменная для передачи аргументом
    System.out.println("До вызова метода alpha(): A="+A);
    alpha(A);
    System.out.println("После вызова метода alpha: A="+A);
    int[] B={1,3,5};    // Массив для передачи аргументом
    System.out.println("До вызова метода bravo(): B="+Arrays.toString(B));
    bravo(B);
    System.out.println("После вызова метода bravo():
B="+Arrays.toString(B));
    int[] C={2,4,6};    // Массив для передачи аргументом
    System.out.println("До вызова метода charlie():
C="+Arrays.toString(C));
    charlie(C);
    System.out.println("После вызова метода charlie():
C="+Arrays.toString(C));
}
}
```

Аргументы в методы передаются по значению: для формального аргумента создается техническая копия, которая собственно и передается в метод

Результат выполнения программы:

До вызова метода alpha(): A=100
 В методе alpha(). На входе: 100
 В методе alpha(). **На выходе: 101**
 После вызова метода alpha: **A=100**

До вызова метода bravo(): **B=[1, 3, 5]**
 В методе bravo(). На входе: [1, 3, 5]
 В методе bravo(). На выходе: [2, 4, 6]
 После вызова метода bravo(): **B=[2, 4, 6]**

До вызова метода charlie(): **C=[2, 4, 6]**
 В методе charlie(). На входе: [2, 4, 6]
 В методе charlie(). На выходе: [3, 5, 7]
 После вызова метода charlie(): **C=[2, 4, 6]**

Рекурсия



При рекурсии описание метода содержит вызов того же метода (но, как правило, с другим значением аргумента)

Факториал числа:

$$n! = n \cdot (n - 1) \cdot \dots \cdot 2 \cdot 1 = n \cdot (n - 1)!$$

```
static int factorial(int n){
    if(n==1) return 1;
    else return n*factorial(n-1);
}
```

Сумма квадратов:

$$1^2 + \dots + n^2 = (1^2 + \dots + (n - 1)^2) + n^2$$

```
static int sqrs(int n){
    if(n==0) return 0;
    else return n*n+sqrs(n-1);
}
```

Числа Фибоначчи:

$$F_n = F_{n-1} + F_{n-2}$$

```
static int fibs(int n){
    if(n==1||n==2) return 1;
    else return fibs(n-1)+fibs(n-2);
}
```

Отображение содержимого массива:

```
static void show(int[] a,int k){
    System.out.print(a[k]+" ");
    if(k==a.length-1){
        System.out.println();
    }
    else{
        show(a,k+1);
    }
}
```

```

class RecursionDemo{
    // Метод для вычисления факториала числа:
    static int factorial(int n){
        if(n==1) return 1;
        else return n*factorial(n-1);
    } // Метод для вычисления чисел Фибоначчи:
    static int fibs(int n){
        if(n==1||n==2) return 1;
        else return fibs(n-1)+fibs(n-2);
    } // Метод для вычисления суммы квадратов чисел:
    static int sqrs(int n){
        if(n==0) return 0;
        else return n*n+sqrs(n-1);
    }
    // Метод для отображения содержимого массива:
    static void show(int[] a,int k){
        System.out.print(a[k]+" ");
        if(k==a.length-1){
            System.out.println();
        }
        else{
            show(a,k+1);
        }
    }
    // Перегрузка метода:
    static void show(int[] a){
        show(a,0);
    }
    // Продолжение - на следующем слайде!!!

```



```
// Главный метод:
public static void main(String[] args){
    System.out.println("Факториал числа:");
    for(int k=1;k<=10;k++){
        System.out.println(k+"!="+factorial(k));
    }
    System.out.println("Числа Фибоначчи:");
    for(int k=1;k<=10;k++){
        System.out.print(fibs(k)+" ");
    }
    System.out.print("\nСумма квадратов от 1 до 10: ");
    System.out.println(sqrs(10));
    // Числовой массив:
    int[] A={1,3,5,7,9,11,13,15,17,19,21};
    // Отображение всех
    // элементов массива:
    show(A);
    // Отображение элементов,
    // начиная с третьего:
    show(A,2);
}
```

Факториал числа:

1!=1

2!=2

3!=6

4!=24

5!=120

6!=720

7!=5040

8!=40320

9!=362880

10!=3628800

Числа Фибоначчи:

1 1 2 3 5 8 13 21 34 55

Сумма квадратов от 1 до 10: 385

1 3 5 7 9 11 13 15 17 19 21

5 7 9 11 13 15 17 19 21

Метод с произвольным количеством аргументов



- Троеточие `...` после идентификатора типа в описании аргумента означает, что за этим аргументом "скрывается" список произвольного количества аргументов соответствующего типа
- Такой аргумент обрабатывается как массив и описывается последним
- Количество фактических параметров, "спрятанных" за данным аргументом можно узнать с помощью свойства `length`

Например:

```
тип метод(int... a){  
    // команды  
}
```

Метод с произвольным количеством целочисленных аргументов

```
тип метод(String t,int... a){  
    // команды  
}
```

Первый аргумент текстовый, а далее - произвольное количество целочисленных аргументов

Сначала массив из целых чисел, а затем произвольное количество символов

```
тип метод(int[] a,char...b){  
    // команды  
}
```

```
class ArbArgsDemo{
    // Метод для вычисления суммы чисел:
    static int sum(int... a){
        int res=0;
        for(int k=0;k<a.length;k++){
            res+=a[k];
        }
        return res;
    } // Метод для извлечения символов из текста:
    static String getText(String t,int... a){
        String res="";
        for(int k=0;k<a.length;k++){
            res+=t.charAt(a[k]);
        }
        return res;
    } // Метод отображает значения аргументов:
    static void show(int[] a,char... b){
        System.out.print("Чисел "+a.length+": ");
        for(int k=0;k<a.length-1;k++){
            System.out.print(a[k]+" ");
        }
        System.out.println("и "+a[a.length-1]);
        System.out.print("Символов "+b.length+": ");
        for(int k=0;k<b.length-1;k++){
            System.out.print(b[k]+" ");
        }
        System.out.println("и "+b[b.length-1]);
    }
} // Продолжение - на следующем слайде!!!
```

```
// Главный метод:  
public static void main(String[] args){  
    // Примеры вызова методов.  
    // Вычисление суммы чисел:  
    System.out.println("Сумма чисел: "+sum(1,6,9,2,4));  
    System.out.println("Сумма чисел: "+sum(5,1,2));  
    // Формируется текст:  
    System.out.println(getText("Раз два три",0,10,8,1));  
    System.out.println(getText("Бревно",3,5,1,5,4));  
    // Отображаются аргументы:  
    show(new int[]{1,3,5}, 'A', 'B', 'C', 'D', 'E');  
    show(new int[]{1,3,5,7,9}, 'A', 'B', 'C', 'D');  
}  
}
```

Результат выполнения программы:

Сумма чисел: 22

Сумма чисел: 8

Рита

ворон

Чисел 3: 1 3 и 5

Символов 5: A B C D и E

Чисел 5: 1 3 5 7 и 9

Символов 4: A B C и D

Домашнее задание



- Напишите программу с методом для вычисления двойного факториала числа $n!! = n \cdot (n - 2) \cdot (n - 4) \dots$. Предложите вариант реализации метода с использованием рекурсии и без него.
- Напишите программу с перегруженным методом, который отображает содержимое массивов разных типов. Массив передается аргументом методу.
- Напишите программу с методом, который возвращает результатом массив из нечетных чисел. Аргументом методу передается размер массива.
- Напишите программу с методом, который возвращает наибольшее из значений, переданных аргументами методу. Количество аргументов может быть произвольным.

