

语言基础第六天：

笔记：

1. 数组：

◦ 复制：

- System.arraycopy(源数组,源数组的起始下标,目标数组,目标数组的起始下标,要复制的元素个数);

优点：灵活性好

```
int[] a = {10, 20, 30, 40, 50};
int[] b = new int[6]; //0,0,0,0,0,0
//a:源数组
//1:源数组的起始下标
//b:目标数组
//0:目标数组的起始下标
//4:要复制的元素个数
System.arraycopy(a, 1, b, 0, 4);
for (int i = 0; i < b.length; i++) {
    System.out.println(b[i]);
}
```

- int[] b = Arrays.copyOf(a, 6);

特点：整个复制

```
package day06;

import java.util.Arrays;

public class test123 {
    public static void main(String[] args) {
        int[] a = {10, 20, 30, 40, 50};
        //a:源数组
        //b:目标数组
        //6:目标数组的长度
        //若目标数组的长度>源数组的长度，则在末尾补默认值
        //若目标数组的长度<源数组的长度，则将末尾的截掉
        int[] b = Arrays.copyOf(a, 6);
        for (int i = 0; i < b.length; i++) {
            System.out.println(b[i]);
        }
    }
}
```

◦ 扩容：

- a = Arrays.copyOf(a,a.length+1);

特点：Arrays.copyOf()的一种用法

```
package day06;
```

```
import java.util.Arrays;

public class test123 {
    public static void main(String[] args) {
        int[] a = {10, 20, 30, 40, 50};
        //创建了一个更大的新的数组，并将数据复制进去了
        a = Arrays.copyOf(a, a.length + 1);
        for (int i = 0; i < a.length; i++) {
            System.out.println(a[i]);
        }
    }
}
```

○ 排序:

- Arrays.sort(arr);

```
package day06;

import java.util.Arrays;

public class test123 {
    public static void main(String[] args) {
        int[] arr = new int[10];
        for (int i = 0; i < arr.length; i++) {
            arr[i] = (int) (Math.random() * 100);
            System.out.println(arr[i]);
        }
        Arrays.sort(arr); //对arr进行升序排列
        System.out.println("排序后:");
        for (int i = 0; i < arr.length; i++) {
            System.out.println(arr[i]);
        }
        System.out.println("倒序输出:"); //只是倒着显示，数组中数据的顺序没变
        for (int i = arr.length - 1; i >= 0; i--) {
            System.out.println(arr[i]);
        }
    }
}
```

2. 方法（函数、过程）：

- 封装一段特定的业务逻辑功能
- 尽可能独立，一个方法只干一件事
- 方法可以被反复调用多次
- 可以减少代码重复，有利于代码维护
- 何时用：假设有一个功能，在很多地方都要用，就将此功能封装到一个方法中

3. 方法的定义：

```
//无参，无返回值
public static void say() {
    System.out.println("大家好，我叫WKJ，今年38岁了");
}

//有参，无返回值
```

```

public static void sayHi(String name) { //形参
    System.out.println("大家好，我叫" + name + "，今年38岁了");
}

//有参，无返回值
public static void sayHello(String name, int age) { //形参
    System.out.println("大家好，我叫" + name + "，今年" + age + "岁了");
}

//无参，有返回值
public static double getNum() {
    //有返回值的方法中，必须有return
    //并且return后数据的类型还必须与返回值类型匹配
    //return "abc"; //编译错误，返回值类型不匹配
    return 8.88; //1)结束方法的执行 2)返回结果给调用方
}

//有参，有返回值
public static int plus(int num1, int num2) {
    int num = num1 + num2;
    return num; //返回的是num里面的那个数
    //return num1+num2; //返回的是num1与num2的和
    //建议这样写: return num1+num2;
}

//无参，有返回值
public static int[] testArray() {
    int[] arr = new int[10];
    for (int i = 0; i < arr.length; i++) {
        arr[i] = (int) (Math.random() * 100);
    }
    return arr;
}

```

4. 方法的调用：

- 无返回值：方法名(有参传参);

```

say(); //调用say()方法

//sayHi(); //编译错误，有参必须传参
//sayHi(250); //编译错误，参数类型必须匹配
sayHi("张三"); //String name="张三" //实参
sayHi("李四"); //String name="李四" //实参
sayHi("王五"); //String name="王五" //实参

sayHello("张三", 25); //实参 String name = "张三"; int age = 25;
sayHello("李四", 24); //实参

```

- 有返回值：数据类型 变量 = 方法名(有参传参);

```
double a = getNum(); //getNum()的值就是return后的那个数
System.out.println(a); //8.88, 模拟对返回值的后续操作

int b = plus(5, 6); //plus(5,6)的值就是return后的那个数
System.out.println(b); //11, 模拟对返回值的后续操作

int m = 5, n = 6;
int c = plus(m, n); //传递的是m和n里面的数
System.out.println(c); //11, 模拟对返回值的后续操作

int[] d = testArray();
System.out.println(d.length); //10, 模拟对返回值的后续操作
```

5. return:

- return 值; (用在有返回值的方法中)
 - 1)结束方法的执行
 - 2)返回结果给调用方
- return; (用在无返回值的方法中)
 - 1)结束方法的执行

补充:

1. 形参: 形式参数, 定义方法时的参数叫做形参
实参: 实际参数, 调用方法时的参数叫做实参
2. 明日单词:

```
1)class: 类
2)new: 新的
3)student: 学生
4)name: 姓名
5)age: 年龄
6)address: 家庭地址
7)study: 学习
8)sayHi: 问好
9>null: 空
10)overload: 重载
11)show: 显示
12)battleship: 战舰
13)bomb: 炸弹
14)observe: 侦察
16)mine: 水雷
17)submarine: 潜艇
18)world: 世界
19)width: 宽
20)height: 高
21)x: x坐标
22)y: y坐标
23)speed: 速度
24)life: 命
25)move: 移动
26)test: 测试
```

```

package day06;

public class Test {
    public static void main(String[] args) {
        say();

        sayHi("java");

        sayHello("java", 99);

        double a = getNum();
        System.out.println(a);

        int b = plus(1, 2);
        System.out.println(b);

        int m = 3, n = 4;
        int c = plus(m, n);
        System.out.println(c);

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

    //无返回值, 无参
    public static void say() {
        System.out.println("大家好");
    }

    //无返回值, 有一个参
    public static void sayHi(String name) {
        System.out.println("大家好, 我的名字是" + name);
    }

    //无返回值, 有多个参
    public static void sayHello(String name, int age) {
        System.out.println("大家好, 我的名字是" + name + ", 年龄是" + age);
    }

    //有返回值, 无参
    public static double getNum() {
        return 3.14159;
    }

    //有返回值, 有参
    public static int plus(int num1, int num2) {
        return num1 + num2;
    }

    //有返回值, 无参
    public static int[] testArray() {
        int[] arr = new int[10];
        for (int i = 0; i < arr.length; i++) {
            arr[i] = (int) (Math.random() * 100);
        }
        return arr;
    }
}

```

```
}  
}
```

扩展练习:

1. 定义一个方法generate()

- 根据第一个参数生成指定长度的int数组;
- 根据第二个范围参数随机生成0到指定范围的随机数, 填充到数组元素中, 并将该数组返回;
- 测试数组内容。

```
public class extra_1 {  
    public static void main(String[] args) {  
        int[] a = generate(10, 5);  
        for (int i = 0; i < a.length; i++) {  
            System.out.println(a[i]);  
        }  
    }  
  
    public static int[] generate(int len, int range) {  
        int[] arr = new int[len];  
        for (int i = 0; i < len; i++) {  
            arr[i] = (int) (Math.random() * (range + 1));  
        }  
        return arr;  
    }  
}
```