

语言基础第六天：

回顾：

1. for结构：应用率高、与次数相关
2. 三种结构如何选择：
3. break：跳出循环
continue：跳过循环体中剩余语句而进入下一次循环
4. 嵌套循环：
循环中套循环，外层走一次，内层走所有次，层数越少越好，break只能跳出当前一层循环
5. 数组：
 - 是一种数据类型(引用类型)
 - 相同数据类型元素的集合

```
int[] arr = new int[3]; //0,0,0
int[] arr = {1,4,7};
int[] arr = new int[]{1,4,7};
System.out.println(arr.length);
arr[0] = 100;
System.out.println(arr[arr.length-1]);
for(int i=0;i<arr.length;i++){
    arr[i] = (int)(Math.random()*100);
    System.out.println(arr[i]);
}
```

精华笔记：

1. 数组：
 - 复制：
 - System.arraycopy(a,1,b,0,4);
 - int[] b = Arrays.copyOf(a,6);
a = Arrays.copyOf(a,a.length+1); //扩容
 - 排序：

```
Arrays.sort(arr); //升序排列
```

2. 方法：函数、过程
 - 封装一段特定的业务逻辑功能
 - 方法要尽可能独立，一个方法只干一件事
 - 方法可以被反复多次调用
 - 方法可以减少代码重复，有利于代码维护
 - 何时用：只要是一个独立的业务，就建议抽到方法中
3. 方法的定义：五要素

修饰词 返回值类型 方法名(参数列表) {

方法体

}

4. 方法的调用：

- 无返回值：方法名(有参传参);
- 有返回值：数据类型 变量 = 方法名(有参传参);

5. return：

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

笔记：

1. 数组：

- 复制：

- System.arraycopy(a,1,b,0,4);

```
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); //灵活性好
for(int i=0;i<b.length;i++){
    System.out.println(b[i]);
}
```

- int[] b = Arrays.copyOf(a,6);
a = Arrays.copyOf(a,a.length+1); //数组的扩容

```
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>源数组长度，则在末尾补默认值
//----若目标数组的长度<源数组长度，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]);
}

//数组的扩容
int[] a = {10,20,30,40,50};
//数组的扩容(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]);
}
```

```

package day06;
import java.util.Arrays;
//求数组元素的最大值，并将最大值放在数组最后一个元素的下一个位置
public class MaxOfArray {
    public static void main(String[] args) {
        int[] arr = new int[10];
        for(int i=0;i<arr.length;i++){ //遍历arr数组
            arr[i] = (int)(Math.random()*100); //给每个元素赋值为0到99
            //之间的随机数
            System.out.println(arr[i]); //输出每个元素的值
        }

        int max = arr[0]; //假设第1个元素为最大值
        for(int i=1;i<arr.length;i++){ //遍历剩余元素
            if(arr[i]>max){ //若剩余元素大于max
                max = arr[i]; //修改max为较大的
            }
        }
        System.out.println("最大值为:"+max);

        arr = Arrays.copyOf(arr,arr.length+1); //扩容
        arr[arr.length-1] = max; //将max赋值给arr的最后一个元素
        for(int i=0;i<arr.length;i++){
            System.out.println(arr[i]);
        }
    }
}

```

○ 排序:

- Arrays.sort(arr);

```

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("排序后数据:");
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，但数组中数据没有改变
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

3. 方法的定义：五要素

```
修饰词 返回值类型 方法名(参数列表) {  
    方法体  
}
```

```
//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");  
}  
  
//有参有返回值  
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(){  
    /*  
        java规定：  
        1)有返回值的方法中，必须得通过return来返回数据  
        2)并且return后数据的类型，必须与返回值类型匹配  
    */  
    //return "abc"; //编译错误，返回值类型必须匹配  
    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  
}  
  
//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);

```

public class MethodDemo {
    public static void main(String[] args) {
        say(); //调用say()方法

        //sayHi(); //编译错误, 有参则必须传参
        //sayHi(250); //编译错误, 参数类型必须匹配
        sayHi("zhangsan"); //String name="zhangsan" //实参
        sayHi("lisi"); //String name="lisi" //实参

        sayHello("zhangsan",25); //实参 String name="zhangsan",int age=25
        sayHello("lisi",24); //实参
    }
}

```

- 有返回值: 数据类型 变量 = 方法名(有参传参);

```

//方法的演示
public class MethodDemo {
    public static void main(String[] args) {
        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("长度为:"+d.length); //10---模拟对返回值的后续操作
        for(int i=0;i<d.length;i++){ //---模拟对返回值的后续操作
            System.out.println(d[i]);
        }
    }
}

```

5. return:

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

补充:

1. 形参: 定义方法时的参数叫形参
实参: 调用方法时的参数叫实参
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

```
public static void main(String[] args){
    say();
    sayHi("zhangsan");
    sayHello("zhangsan",35);
    double a = getNum(); 输出a(模拟对返回值的后续操作)
    int b = plus(2,4); 输出b(模拟对返回值的后续操作)
    int m=2,n=4; int c = plus(m,n); 输出c(模拟对返回值的后续操作)
    int[] d = testArray(); 输出d的长度(模拟对返回值的后续操作) 遍历输出元素(模拟对返回值
    后续操作)
}
public static void say(){ ... }
public static void sayHi(String name){ ... }
public static void sayHello(String name,int age){ ... }
public static double getNum(){ return 8.88; }
public static int plus(int num1,int num2){ int num=num1+num2; return num; }
public static int[] testArray(){
    int[] arr = ...; for(...){...}
    return arr;
}
```

