面向对象编程

1. 对象和类

1.面向对象程序设计

- 面向过程的结构化程序设计

>.缺乏对数据的封装。

>.数据和操作数据的方法分离。

- 什么是抽象数据类型

>.抽象数据类型，将不同类型的数据和集合组成一个整体，用来描述一种新的事物。

- 什么是类

>.类是java的基本组成单位，且是一种抽象数据类型。

>.类的组成单位为属性(成员变量/数据)，还定义了方法(函数/操作数据的结构)，

2.定义一个类

- 定义类的成员变量

>.成员变量类型 名称：String name;

>.用于描述该类型对象共同的数据结构。

>.对于成员变量，如果在使用之前，没有给他赋初值，则系统会给其默认的初始值。如下：

|  |  |
| --- | --- |
| 成员变量类型 | 默认初始值 |
| 整型(byte\short\int\long) | 0 |
| 浮点型(float\double) | 0.0 |
| boolean | false |
| char | 空字符 |
| 引用类型(String\Scanner\Emp...) | null |

- 定义类的方法

>.修饰词 返回值类型 方法名(参数列表){方法体}

public void print (Emp emp) {}

>.用于描述对象的行为，封装对象的功能。

3.创建并使用对象

- 使用new关键字创建对象

>.类创建完后，可以使用new关键字创建对象。创建对象的过程称为类的实例化。

>.new + 类名();

new Person(); 实例化人对象

new Emp(); 实例化员工对象

new JFrame(); 实例化窗体对象

- 引用类型变量

>. 为了能够对实例化对象的对象进行访问控制，需要使用特殊的变量。

Person p = new Person(); p : 引用

>. 引用通常存放该对象的地址信息，称为指向该类对象的引用，通过其可以对该对象中的属性和方法进行访问。

>. 除了8种基本类型外，其余所有类型都为引用数据类型(包括：数组、接口、String等)

- 访问对象的成员变量、调用方法

>.通过 引用.属性 或 引用.方法 的形式，调用对象的属性和方法。

- 引用类型变量的赋值

>.引用类型变量存储的是对象的地址信息，相同类型的引用类型变量之间可以相互赋值。

Person p = new Person(); Person p2 = p;将p的地址信息赋值给p2

p和p2指向了同一对象，

>.两个相同类型之间的赋值不会创建新的对象，但可能会两个引用指向同一对象。

- null和nullPointerException

>.对于引用类型变量，可以对其赋值为null，且默认值为null。“空”值该引用不会指向任何一个对象。

>.当一个引用为null时，如果通过该引用访问对象成员变量或者调用方法时，是不符合逻辑的，因此会产生NullPointerException

二、方法

1.方法的重载

- 方法的签名

>.方法的签名：方法名+参数列表类型

>.一个类中不可以出现两个方法的签名完全相同的方法。但是可以有方法名相同但参数列表不同的方法。

- 方法的重载及其意义

>.重载：方法的名字相同，但参数列表不同(数据类型，个数，不同类型的顺序)

- 编译时根据签名绑定调用方法

>.编译器在编译时，会根据签名来绑定调用不同的方法。可以理解为几种不一样的方法，只不过恰好方法名相同而已。

2.构造方法(重)

- 构造方法的语法结构

>. 访问修饰符 类名() {}

>. 构造方法实在类中定义的方法，不同于其他方法。

>. 构造方法的规则：

- 方法名和类名相同

- 构造方法没有返回值，且不写void

public Person() {}

- 通过构造方法初始化成员变量

>. 构造方法的主要作用，是用来初始化属性的。让创建对象时，属性拥有初始值。

- this关键字的使用

>. 区分属性和参数同名的问题。

>. 用于指向调用该方法或属性的当前对象。

谁调用了属性和方法，那么this就指谁。

zw- 默认的构造方法

>. 如果该类没有构造方法，则系统会提供一个默认的无参构造方法。

>. 如果显式的写出一个构造方法，则系统不会提供默认的无参构造方法。

- 构造方法的重载

>. 构造方法也符合重载规则。

>. 根据参数调用不同的构造方法。

>. this()表示调用当前重载的构造方法。

>. this()且必须放在构造方法的第一行(笔试)

三、引用类型数组

- 数组是对象

>. 在Java中，数组属于引用类型。

>. 数组对象和对象的储存位置一致，都存储与堆中，数组的引用存储地址信息，

指向数组对象。

>. 数组的元素，可以看成类型全部相同的成员变量。

int[] arr = new int[3]; Person[] p = new Person[3];

- 引用类型数组的声明

>. 类名[] 引用 = new 类名 [数组长度];

Person[] p = new Person[100];

- 引用类型数组的初始化

>. 采用for循环初始化，只不过每一个元素为对象。

for (int i = 0; i < p.length; i++) {

p[i] = new Person();

}

- 数组的元素。是基本类型数组

>. int[][] arr= new int[3][];

arr[0] = new int[2];

arr[1] = new int[2];

arr[2] = new int [2];

arr[1][1] = 100;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | j = 0 | j = 1 | j = 2 | j = 3 |
| i = 0 | arr[0][0] | arr[0][1] | arr[0][2] | arr[0][3] |
| i = 1 | arr[1][0] | arr[1][1] | arr[1][2] | arr[1][3] |
| i = 2 | arr[2][0] | arr[2][1] | arr[2][2] | arr[2][3] |

作业：

1. 新建类Student
2. 创建属性：学生姓名、学生班级、学号、学生分数
3. 创建构造方法：对属性进行赋值
4. 创建打印学生信息的方法：

[studentName:\*\*\*\*,studentClass:\*\*\*\*，studentNumber:\*\*\*\*,studentScore：\*\*\*\*]

1. 创建测试类TestStudent，测试并打印学生信息的方法
2. 创建学生分数数组，并排序打印，且计算平均分打印。
3. 找出最大分数并加在数组最后一位。