

# 面向对象第一天:

---

## 精华笔记:

---

### 1. 什么是类? 什么是对象?

- 现实生活是由很多很多对象组成的, 基于对象抽出了类
- 对象: 软件中真实存在的单个的个体/东西  
类: 类型/类别, 代表一类个体
- 类是对象的模子/模板, 对象是类的具体的实例, 可以将类理解为类别/模子/图纸
- 类中可以包含:
  - 对象的属性/特征/数据-----成员变量
  - 对象的行为/动作/功能-----方法
- 一个类可以创建多个对象

### 2. 如何创建类? 如何创建对象? 如何访问成员?

### 3. this: 指代当前对象, 哪个对象调用方法它指的就是哪个对象

- 只能用在方法中, 方法中访问成员变量之前默认有个this.
- this的用法:
  - this.成员变量名-----访问成员变量(必须掌握)  

当成员变量与局部变量同名时, 若想访问成员变量, 则this不能省略
  - this.方法名()-----调用方法(一般不用)
  - this()-----调用构造方法(一般不用)

### 4. 构造方法: 构造函数、构造器、构建器-----好处: 复用给成员变量赋初始值的代码

- 作用: 给成员变量赋初始值
- 语法: 与类同名, 没有返回值类型(连void都没有)
- 调用: 在创建(new)对象时被自动调用
- 若自己不写构造方法, 则编译器默认提供一个无参构造方法, 若自己写了构造方法, 则不再默认提供
- 构造方法可以重载

## 笔记:

---

### 1. 什么是类? 什么是对象?

- 现实生活是由很多很多对象组成的, 基于对象抽出了类
- 对象: 软件中真实存在的单个的个体/东西  
类: 类型/类别, 代表一类个体
- 类是对象的模子/模板, 对象是类的具体的实例, 可以将类理解为类别/模子/图纸
- 类中可以包含:

- 对象的属性/特征/数据-----成员变量
- 对象的行为/动作/功能-----方法
- 一个类可以创建多个对象

## 2. 如何创建类？如何创建对象？如何访问成员？

```
public class Student {
    //成员变量
    String name;
    int age;
    String className;
    String stuId;

    //方法
    void study(){
        System.out.println(name+"在学习...");
    }
    void sayHi(){
        System.out.println("大家好，我叫"+this.name+"，今年"+this.age+"岁了，所在
班级为"+this.className+"，学号为:"+this.stuId);
    }
    void playWith(String anotherName){
        System.out.println(this.name+"正在和"+anotherName+"一起玩...");
    }
}

public class StudentTest {
    public static void main(String[] args) {
        //创建一个学生对象
        Student zs = new Student();
        //访问成员变量
        zs.name = "张三";
        zs.age = 24;
        zs.className = "jsd2302";
        zs.stuId = "001";
        //调用方法
        zs.study();
        zs.sayHi();
        zs.playWith("李四");

        Student ls = new Student();
        ls.name = "李四";
        ls.age = 25;
        ls.className = "jsd2302";
        ls.stuId = "002";
        ls.study();
        ls.sayHi();
        ls.playWith("张三");

        //1)创建了一个学生对象
        //2)给所有成员变量赋默认值
        Student ww = new Student();
        ww.study();
        ww.sayHi();
        ww.playWith("张三");
    }
}
```

```
}  
}
```

3. this: 指代当前对象, 哪个对象调用方法它指的就是哪个对象

- 只能用在方法中, 方法中访问成员变量之前默认有个this.
- this的用法:
  - this.成员变量名-----访问成员变量(必须掌握)

当成员变量与局部变量同名时, 若想访问成员变量, 则this不能省略

- this.方法名()-----调用方法(一般不用)
- this()-----调用构造方法(一般不用)

4. 构造方法: 构造函数、构造器、构建器-----好处: 复用给成员变量赋初始值的代码

- 作用: 给成员变量赋初始值
- 语法: 与类同名, 没有返回值类型(连void都没有)
- 调用: 在创建(new)对象时被自动调用
- 若自己不写构造方法, 则编译器默认提供一个无参构造方法, 若自己写了构造方法, 则不再默认提供
- 构造方法可以重载

```
public class Student {  
    //成员变量  
    String name;  
    int age;  
    String className;  
    String stuId;  
  
    //构造方法  
    Student(){  
    }  
    Student(String name,int age,String className,String stuId){  
        this.name = name;  
        this.age = age;  
        this.className = className;  
        this.stuId = stuId;  
    }  
  
    //方法  
    void study(){  
        System.out.println(name+"在学习...");  
    }  
    void sayHi(){  
        System.out.println("大家好, 我叫"+this.name+", 今年"+this.age+"岁了, 所在班级为"+this.className+", 学号为:"+this.stuId);  
    }  
    void playWith(String anotherName){  
        System.out.println(this.name+"正在和"+anotherName+"一起玩...");  
    }  
}  
public class ConsDemo {  
    public static void main(String[] args) {
```

```

        Student zs = new Student(); //调用无参构造方法
        Student ls = new Student("李四",25); //调用2个参构造
        Student ww = new Student("王五",26,"jsd2302","003"); //调用4个参构造

        zs.sayHi();
        ls.sayHi();
        ww.sayHi();
    }
}

```

## 5. 综合练习:

```

public class Car {
    String brand; //品牌
    String color; //颜色
    double price; //价格

    Car(){
    }
    Car(String brand,String color,double price){
        this.brand = brand;
        this.color = color;
        this.price = price;
    }

    void start(){
        System.out.println(brand+"牌子的"+color+"颜色的"+price+"块钱的车启动了...");
    }
    void run(){
        System.out.println(brand+"牌子的"+color+"颜色的"+price+"块钱的车开始跑了...");
    }
    void stop(){
        System.out.println(brand+"牌子的"+color+"颜色的"+price+"块钱的车停止了...");
    }
}

public class CarTest {
    public static void main(String[] args) {
        Car car1 = new Car();
        car1.brand = "奔驰";
        car1.color = "黑";
        car1.price = 80;
        car1.start();
        car1.run();
        car1.stop();

        Car car2 = new Car("奥迪","银",40);
        car2.start();
        car2.run();
        car2.stop();

        Car car3 = new Car("特斯拉","白",70);
    }
}

```

```

        car3.start();
        car3.run();
        car3.stop();
    }
}

```

## 补充:

### 1. 面向过程和面向对象:

- 面向过程: 以方法为单位来解决问题, 比较适合简单的业务(大象装冰箱、去银行取钱)
- 面向对象: 以对象为单位为解决问题, 比较适合复杂的业务(造个汽车、造个航母)

### 2. 课程: 面向对象5天是面向对象的入门, 基本概念、语法、设计都讲了----讲造车零部件

### 3. OO: 面向对象

OOA: 面向对象分析

OOD: 面向对象设计

OOAD: 面向对象分析与设计

OOP: 面向对象编程-----你们所参与的部分

### 4. 高质量的代码: -----你们以后的目标

- 复用性好、扩展性好、维护性好、移植性好、
- 可读性好、健壮性好、效率好.....

### 5. 类是一种引用数据类型

### 6. 创建对象:

	引用		
数据类型	引用类型变量	指向	对象
Student	zs	= new	Student(); //声明Student类型引用zs指向一个学生对象

### 7. 默认值规则:

```

byte, short, int, long, char-----0
float, double-----0.0
boolean-----false
引用类型-----null

```

### 8. 明日单词:

- 1) Person: 人
- 2) eat: 吃
- 3) sleep: 睡
- 4) salary: 工资
- 5) title: 职称
- 6) teach: 讲、教
- 7) cut: 剪、切
- 8) Animal: 动物
- 9) drink: 喝

10)chick:鸡

11)dog:狗

12)fish:鱼

13)layEgg:下蛋

14)look:看

15)home:家