

Java面向对象编程1

- What?Why?How?



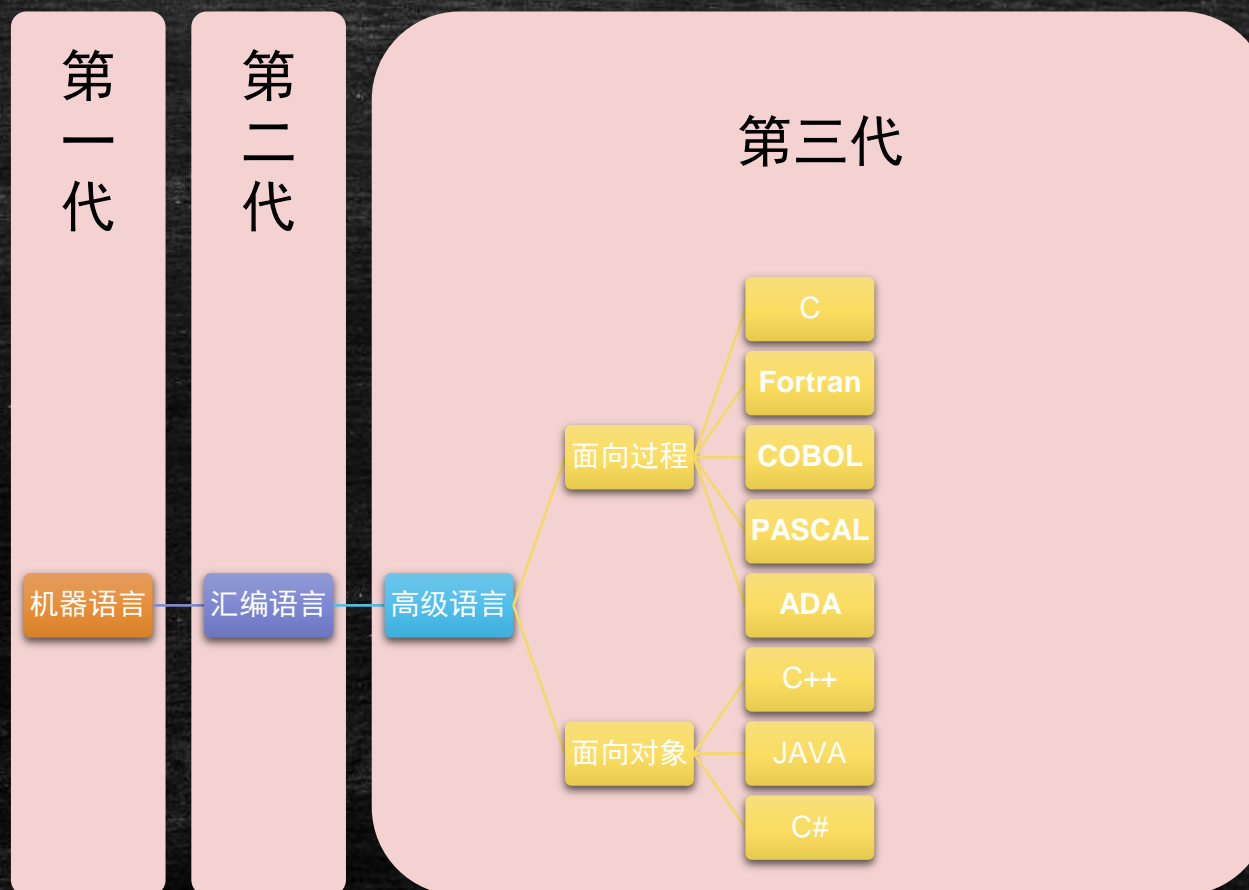
本章概述

- 面向过程编程
- 面向对象编程
- 类和对象
- 局部变量和成员变量



面向对象编程初步

(OOP:Object Oriented Programming)



面向对象编程初步

(OOP:Object Oriented Programming)

	面向过程	面向对象
区别	事物比较简单，可以用线性的思维去解决	事物比较复杂，使用简单的线性思维无法解决
共同点	面向过程和面向对象都是解决实际问题的一种思维方式 二者相辅相成，并不是对立的。 解决复杂问题，通过面向对象方式便于我们从宏观上把握事物之间复杂的关系、方便我们分析整个系统；具体到微观操作，仍然使用面向过程方式来处理	



面向对象编程初步 (OOP:Object Oriented Programming)

- 如何开汽车(事物比较简单, 可以用线性的思维去解决)

- 面向过程:

1. 采离合
2. 挂档
3. 踩油门, 放离合
4. 开了

- 面向对象:

- ✓ 驾驶员
- ✓ 汽车
- ✓ 驾驶员开汽车!
`car.start();`



面向对象编程初步 (OOP:Object Oriented Programming)

如何造汽车(事物比较复杂, 使用
简单的线性思维无法解决)

面向过程:

1. 造车轮?
2. 造发动机?
3. 造车皮?
4. 挡风玻璃?

很难决定上面这些步骤之间的
关系! 先造发动机还是先造车
轮?

面向对象:

车轮
 买橡胶
 到马来西亚
 找到橡胶厂
 掏钱买
 用船将橡胶运到国内
 造磨具
 将橡胶放入磨具
 出车轮
发动机
....
车壳
....
座椅
...
挡风玻璃
....
将上面的造出的东东, 组装, 汽车造



面向对象编程初步 (OOP:Object Oriented Programming)

解决问题：如何统一中国？（很复杂，不能用面向过程解决！）

蒋介石

面向过程的
杰出代表



1. 抢占战败日本的物资，收编伪军，拉美国的金援。
2. 和共产党谈判，拖延时间，为战争准备争取时间
3. 抢占城市等战略要点
4. 开打！
5. 赢！哦耶！

毛泽东

面向对象的
杰出代表



共产党（加强党的建设、增强凝聚力战斗力）

农民

贫农 中农 富农

工人

知识分子

国民党

进步派 中间派 反动派

各民主派别

用统一战线
战略来处理
这些对象的
关系



类和对象

- 现实世界是由什么组成的？



世界由对象组成



类和对象



类可以生成对象，类是对象的抽象



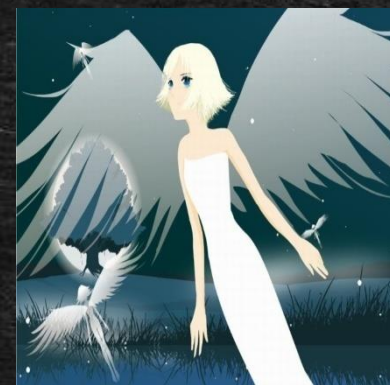
对象和类的概念

- 对象：是具体的事物
- 类：是对对象的抽象（抽象 抽出象的部分）
- 先有具体的对象，然后抽象各个对象之间象的部分，归纳出类通过类再认识其他对象。
- 天使类如何归纳？



通过具体对象抽象出类 A

- 四个具体的天使：



- 抽象他们**共同**的部分：
 - 带翅膀（带翅膀不一定是天使，还可能是鸟人）
 - 女孩
 - 善良



通过类认识其他的对象 A

- 我是天使吗?

```
public class Angel {  
    String wing;  
    String girl;  
    String goodness;  
}
```



身边的对象

张浩



顾客

姓名—张浩
年龄—20
体重—60kg

操作：
购买商品

李明



收银员

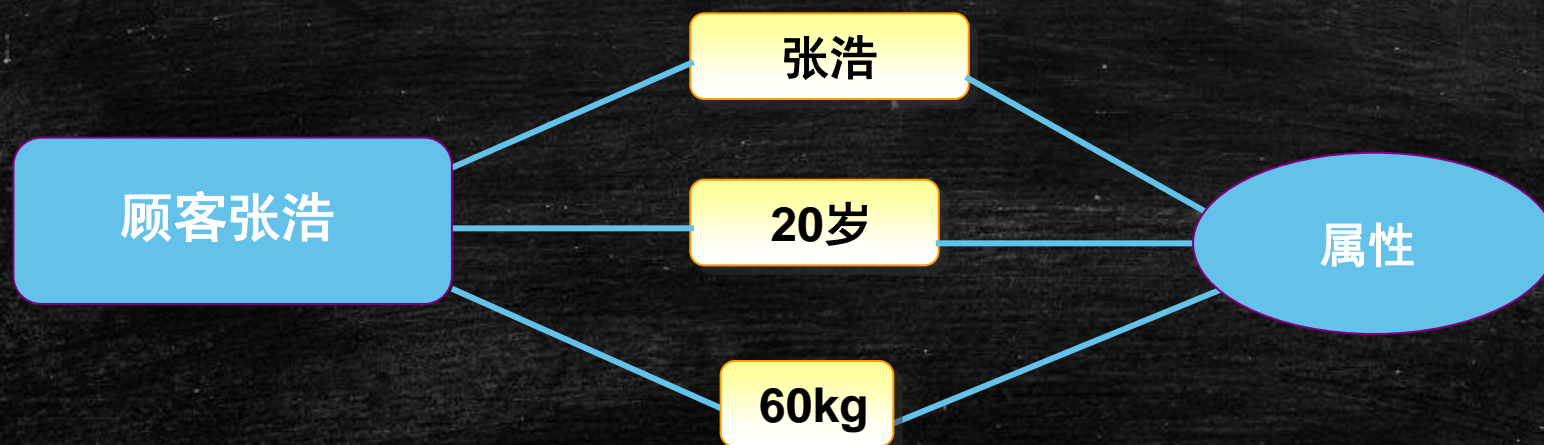
员工号—10001
姓名—李明
部门—财务部

操作：
收款
打印账单



对象的特征——属性

- 属性——对象具有的各种特征
- 每个对象的每个属性都拥有特定值
- 例如：张浩和李明的年龄、姓名不一样



对象的特征——方法

- 方法——对象执行的操作



- 对象：用来描述客观事物的一个实体，由一组属性和方法构成



对象的属性和方法

- 1、法



属性:

品牌: 法拉利
型号: F360 Spider
颜色: 黄色
价格: 380万元

方法:

发动
停止
加速

- 列出



属性:

颜色: 白色

方法:

叫
跑
吃



从对象抽象出“类”

抽取出下列对象的属性和方法的共同特征



轿车

顾客



类

顾客类

轿车类

.....

类

类是模子，确定对象将会拥有的特征（属性）和行为（方法）

具有相同属性和方法的一组对象的集合

类是对象的类型

不同于int类型：具有方法



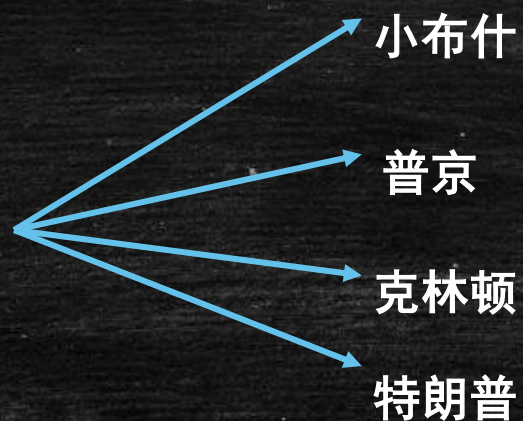
各种口味的球状冰淇淋



类和对象的关系

- 类是抽象的概念，仅仅是模板，比如说：“人”
- 对象是一个你能够看得到、摸得着的具体实体

“人” 类
特征（属性） 年龄 体重
行为（方法） 衣 食 住 行



对象和类的总结 A

- ✓ **对象和类的关系：特殊到一般，具体到抽象。**
- ✓ 类：我们叫做class。
- ✓ 对象：我们叫做Object, instance(实例)。以后我们说某个类的对象，某个类的实例。是一样的意思。
- ✓ 类可以看成一类对象的模板，对象可以看成该类的一个具体实例。
- ✓ 类是用于描述同一类形的对象的一个**抽象**的概念，类中定义了这一类对象所应具有静态和动态属性。
- ✓ 对象是Java程序的核心，在Java程序中“万事万物皆对象”。
- ✓ JDK提供了很多类供编程人员使用，编程人员也可定义自己的类。



类的属性 A

- 属性 field, 或者叫成员变量
- 属性用于定义该类或该类对象包含的数据或者说静态属性。
- **属性作用范围是整个类体**
- 在定义成员变量时可以对其初始化, 如果不对其初始化, Java使用默认的值对其初始化。(数值: 0,0.0 char:\u0000, boolean:false, **所有引用类型:null**)
- 属性定义格式:
 - [修饰符] 属性类型 属性名 = [默认值]

可以省略。
可以是: public,
protected, private
Static, final。
讲到隐藏和封装时再说。

可以是任何类型,
基本类型和引用类
型

合法标识符即可。
首字母小写, 驼
峰原则



类的方法 A

- 面向对象中，整个程序的基本单位是类，方法是从属于对象的
- 方法定义格式：
[修饰符] 方法返回值类型 方法名(形参列表) {
 // n条语句
}
- JAVA中方法参数传递是：值传递！



Java 是面向对象的语言

所有Java程序都以类class为组织单元

关键字class定义自定义的数据类型

```
public class HelloWorld {  
    public static void main(String[] args){  
        System.out.println("Hello World!!!");  
    }  
}
```



Java类模板

- 类将现实世界中的概念模拟到计算机程序中

```
public class 类名 {  
    //定义属性部分  
    属性1的类型 属性1;  
    属性2的类型 属性2;  
    ...  
    属性n的类型 属性n;  
  
    //定义方法部分  
    方法1;  
    方法2;  
    ...  
    方法m;  
}
```



如何定义类

- 定义一个类的步骤
 - 1、定义类名
 - 2、编写类的属性
 - 3、编写类的方法



类示例

```
class Person
```

```
{
```

```
    String name;//姓名
```

```
    int age;//年龄
```

```
    String sex;//性别
```

```
    //自我介绍的方法
```

```
    public void introduce(){
```

```
        System.out.println("name="+name+"\tage="+age+"\tsex="+sex);
```

```
    }
```

```
}
```

成员变量

方法

定义类的introduce()方法，
用于输出类相关的信息



如何创建和使用对象

- 使用对象的步骤:

- 创建对象

- 类名 对象名 = new 类名();

```
Person person = new Person();
```

- 引用对象成员：使用“.”进行以下操作

- 引用类的属性：对象名.属性
 - 引用类的方法：对象名.方法名()

```
person.name = "王一一"; //给name属性赋值  
person.introduce();      //调用introduce()方法
```



如何创建和使用对象示例

- 创建“王一一”这个对象

说一说看到什么效果？

```
public class InitialPerson
{
    public static void main(String[] args)
    {
        Person person=new Person();//创建对象
        System.out.println("***初始化变量前***");
        person.introduce();//调用自我介绍的方法
        person.name="王一一";//给属性赋值
        person.age=20;
        person.sex="男";
        System.out.println("\n***初始化变量后***");
        person.introduce();//调用自我介绍的方法
    }
}
```

说一说又看到什么效果？



如何创建和使用对象示例

编写学生类，输出学生相关信息；编写教员类，输出教员相关信息

学生类

姓名

年龄

班级

爱好

显示学员个人信息

教员类

姓名

专业方向

教授课程

教龄

显示教员个人信息



上机练习1—定义管理员类

- 实现思路：
 - 1、定义管理员类Administrator
 - 2、定义其属性和方法
 - 3、定义测试类TestAdministrator
 - 4、创建两个管理员类的对象，并输出他们的信息



上机练习2—更改管理员密码

- **需求说明：**
 - 输入旧的用户名和密码，如果正确，方有权限更新
 - 从键盘获取新的密码，进行更新
- **实现思路：**
 - 1、创建管理员类的对象
 - 2、利用while实现循环执行



局部变量和成员变量

- 区别

- 声明位置不同 类中 方法中
- 作用范围不同： 当前类的方法 当前方法
 - 不同的方法中即使有同名的局部变量，没有关系，互不影响，建议相同
- 内存存放的位置的：栈内存中 堆内存中
- 成员变量有默认值；局部变量没有默认值



引用类型

- Java 语言中除基本类型之外的变量类型都称之为引用类型。



- Java中的对象和数组是通过引用对其操作的。
 - 引用可以理解作为一种受限的指针
 - 指针是可以进行与整数做加减运算的，两个指针之间也可以进行大小比较运算和相减运算。引用不行，只能进行赋值运算。
 - 引用就是一个变量或对象的别名（引用的本质是一个对象）；指针是一个段内存空间的地址(指向存储一个变量值的空间或一个对象的空间)



内存分析

- 栈：
 - 存放：局部变量
 - 先进后出，自下而上存储
 - 方法执行完毕，自动释放空间
- 堆：
 - 存放new出来的对象
 - 需要垃圾回收器来回收: `System.gc()`
- 方法区：
 - 存放：类的信息(代码)、 `static`变量、字符串常量等.



构造方法 2-1

- 对象初始化

```
Penguin pgn = new Penguin();  
pgn.name = "qq";  
pgn.sex = "Q仔";
```

使用构造方法：
`Penguin pgn1 = new Penguin();`

```
class Penguin {  
    // 属性  
    /* 无参构造方法 */  
    public Penguin() {  
        name = "qq";  
        love = 20;  
        sex = "Q仔";  
        System.out.println("执行构造方法");  
    }  
}
```

能否在创建对象的
同时就完成赋值？

构造方法



构造方法 2-2

构造方法

无返回值类型

与类名相同

```
访问修饰符 构造方法名 () {  
    //初始化代码  
}
```

可以指定参数

系统提供默认无参构造方法

```
public Penguin() {  
  
}
```



构造方法重载2-1

- 自定义构造方法

```
public Penguin {} {  
    name = "qq";  
    love = 20;  
    sex = "Q仔";  
}
```

系统不再提供默认无参构造方法

方法名相同

参数项不相同

与返回值、
访问修饰符无关

方法
重载

```
public Penguin (String name,int health,int love,String sex ) {  
    this.name = name;  
    this.health = health;  
    this.love = love;  
    this.sex = sex;
```

this关键字
是对一个对象的
默认引用，
这里用以区分
同名成员变量

```
System.out.println(45);  
System.out.println(true);  
System.out.println("狗在玩耍！");
```



构造方法重载2-2

- 构造方法重载的调用

```
pgn = new Penguin();  
pgn.print();  
pgn = new Penguin("美美", 80, 20, "Q仔");  
pgn.print();
```

```
public Penguin (String name,int health,int love,String sex ) {  
    this.name = name;  
    this.health = health;  
    this.love = love;  
    this.sex = sex;  
}
```

```
public Penguin () {  
    name = "qq";  
    love = 20;  
    sex = "Q仔";  
}
```



构造方法

- 阅读代码，说出运行结果，

```
Penguin pgn3=new Penguin();  
pgn3.print();
```

```
class Penguin {
```

```
    String name = null; //昵称
```

```
    int health = 0; // 健康值
```

```
    String sex = null; // 性别
```

```
    public void Penguin() {
```

```
        health=10;
```

```
        sex="雄";
```

```
        System.out.println("执行构造方法");
```

```
    }
```

```
    public void print() {
```

```
        System.out.println("企鹅的名字是" + name + ",健康值是"  
                            + health + ",性别是" + sex);
```

```
    }
```

```
}
```

调用默认
无参构造方法

有返回值类型，
不是构造方法



构造器

- 构造器定义： constructor 构造方法
 - 一个在创建对象时被自动调用的特殊方法。
- 构造器作用：
 - 为对象进行初始化（成员变量）工作。
- 构造器是一种特殊的方法：
 - 构造器的方法名必须和类名一致！
 - 构造器虽然有返回值，但是不能定义返回类型(返回值的类型肯定是本类)，不能在构造器里调用return。
 - 通过new关键字调用！！
 - 如果我们没有定义构造器，则系统会自动定义一个无参的构造方法。如果已定义则编译器不会添加无参数构造方法！
 - 与普通方法一样，构造方法也可以重载



示例

- 定义一个“点” (Point) 类用来表示二维空间中的点 (有二个坐标)。要求如下:
 - 可以生成具有特定坐标的点对象。
 - 提供可以设置二个坐标的方法。
 - 提供可以计算该“点”距另外点距离的方法。



示例

- 代码实现交换point中两个坐标的值（实现用对象作为参数）



方法调用

- 形参和实参
 - 定义方法的参数是形式参数
 - 调用方法的参数是实在参数
 - 调用方法时要求参数个数相同，类型兼容
- 参数传递
 - 基本数据类型的参数传递
 - 无法通过方法调用改变变量的值
 - 引用数据类型的参数传递
 - 可以通过方法调用改变变量的值

