计算机学院

北華航天工業學院
NORTH CHINA INSTITUTE OF AEROSPACE ENGINEERING

第5章 面向对象技术-Ⅰ

2021-2022-1

本讲目标

- **面向对象的基本概念**
  - 面向对象的设计思想
  - 类之间的关系
  - 类和对象
- **对象的创建、使用、销毁**
  - 构造方法
- **面向对象的特征**
  - 封装
  - 继承
  - 多态

面向对象的基本概念

面向对象的设计思想

面向对象是一种新兴的程序设计方法,或者是一种新的程序设计规范(paradigm),其基本思想是使用对象、类、继承、封装、多态、消息等基本概念来进行程序设计。

从现实世界中客观存在的事物（即对象）出发来构造软件系统，并且在系统构造中尽可能运用人类的自然思维方式。

开发一个软件是为了解决某些问题，这些问题所涉及的业务范围称作该软件的问题域。

面向对象的特征

- **封装性（Encapsulation）：**

  将相关的数据及其操作结合在一起，使除了该对象的方法以外的其他方法，尽量不能使用这些数据、改变这些数据的状态。也称为消息隐藏。

  其有两层含义：

  - 相关的数据及其操作结合在一起，形成一个不可分割的独立单位（即对象）。
  - 信息隐蔽，即尽可能隐蔽对象的内部细节，对外形成一个边界〔或者说形成一道屏障〕，只保留有限的对外接口使之与外部发生联系。

面向对象的特征

◆继承

特殊类的对象拥有其一般类的全部属性与服务，称作特殊类对一般类的继承。例如，轮船、客轮；人、大人。一个类可以是多个一般类的特殊类，它从多个一般类中继承了属性与服务，这称为多继承。例如，客轮是轮船和客运工具的特殊类。在java语言中，通常我们称一般类为父类（superclass,超类），特殊类为子类(subclass)。

面向对象的特征

◆多态

对象的多态性是指在一般类中定义的属性或服务被特殊类继承之后，可以具有不同的数据类型或表现出不同的行为。这使得同一个属性或服务在一般类及其各个特殊类中具有不同的语义。例如：“几何图形”的”绘图”方法，”椭圆”和”多边形”都是”几何图”的子类，其”绘图”方法功能不同。

面向对象的基本概念

◆ 对象的基本概念

对象是系统中用来描述客观事物的一个实体，它是构成系统的一个基本单位。一个对象由一组属性和对这组属性进行操作的一组服务组成。

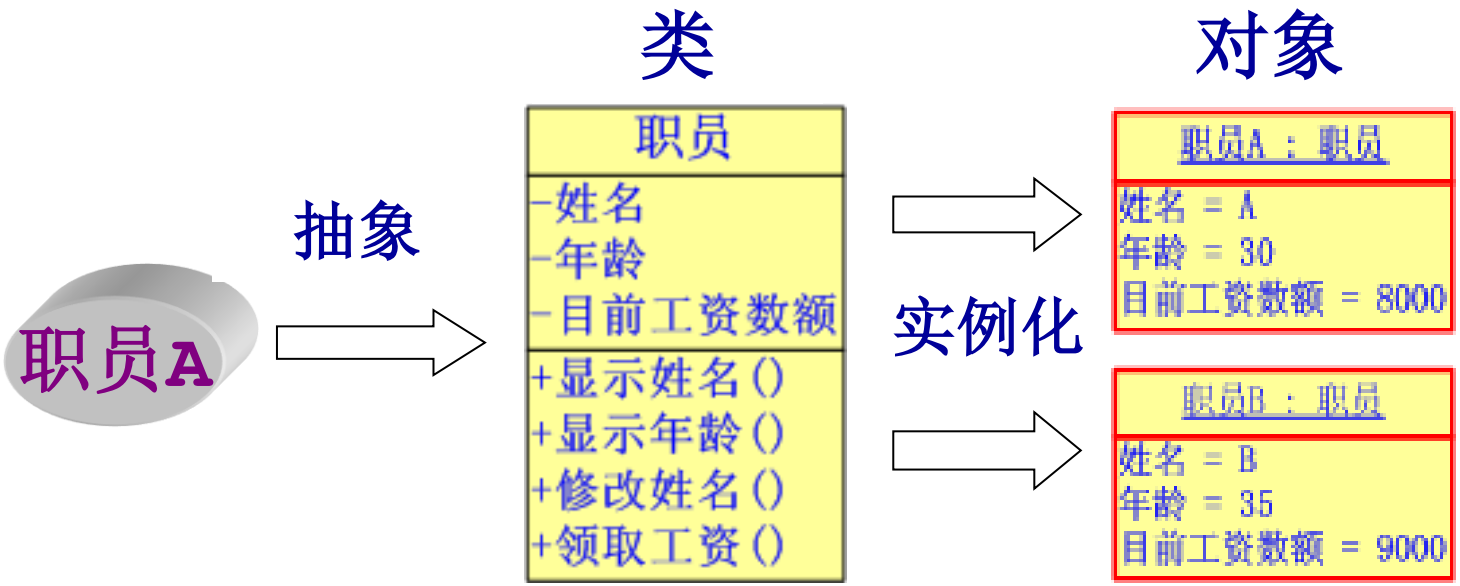
- **对象通过“属性（attribute）”和“方法（method）”来分别对应事物所具有的静态属性和动态属性。**

面向对象的基本概念

- **类的基本概念**

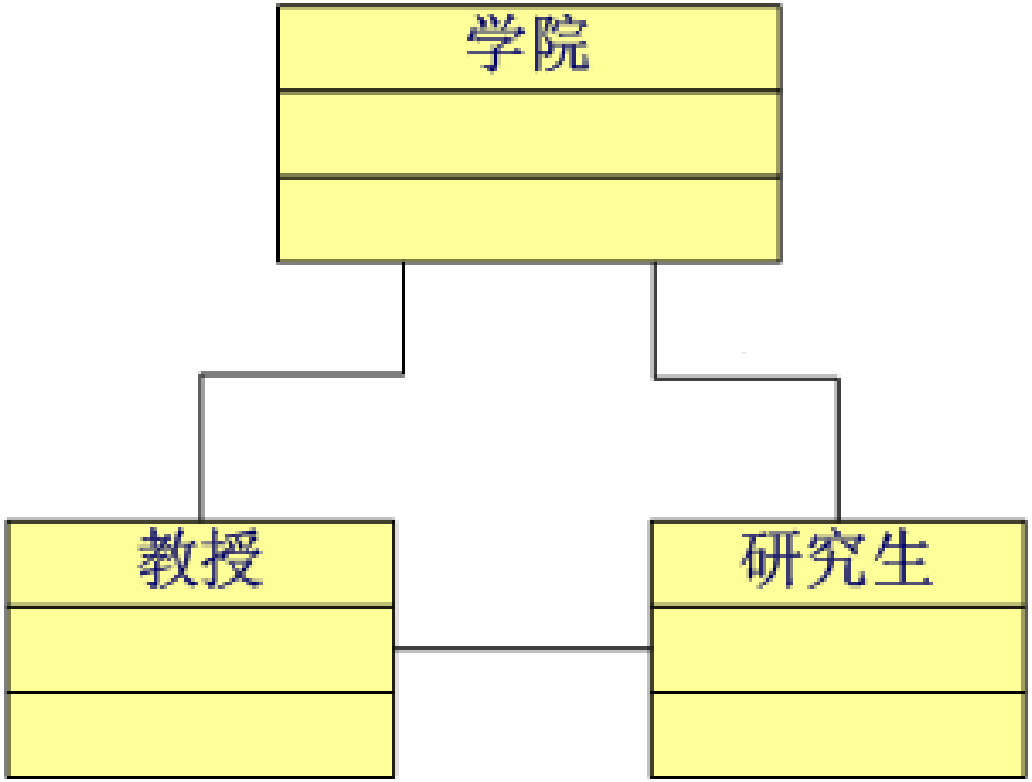
  类是具有相同属性和服务的一组对象的集合抽象，它为属于该类的所有对象提供了统一的抽象描述，定义了同类对象共有的变量和方法。

- **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**



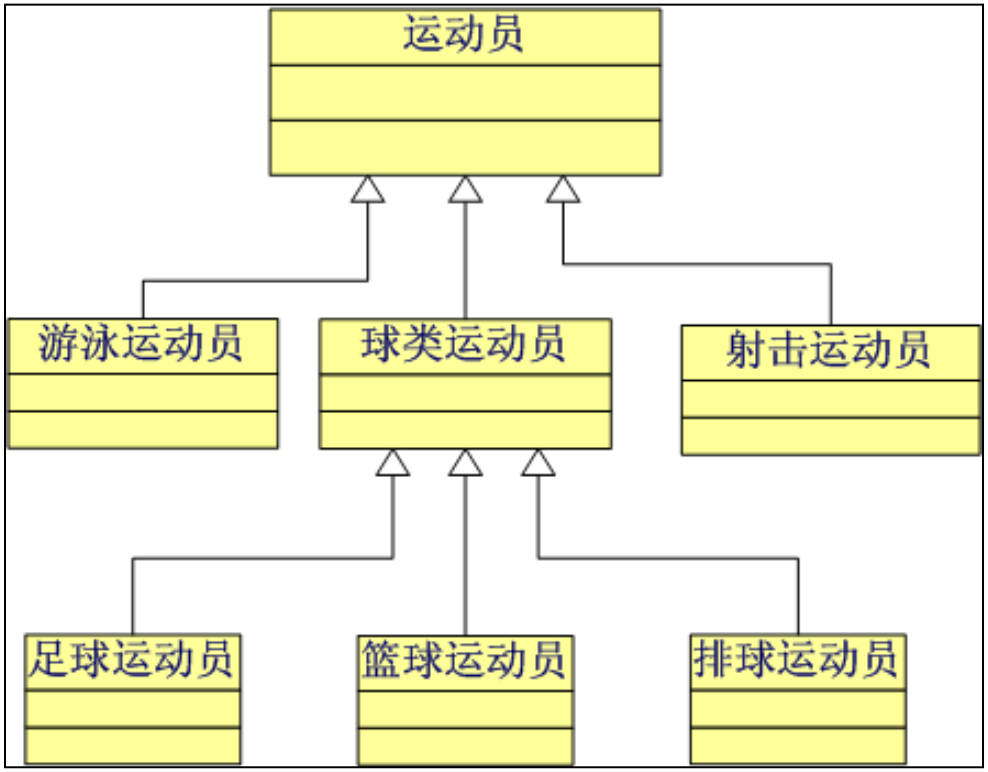
面向对象的基本概念

- 关联关系



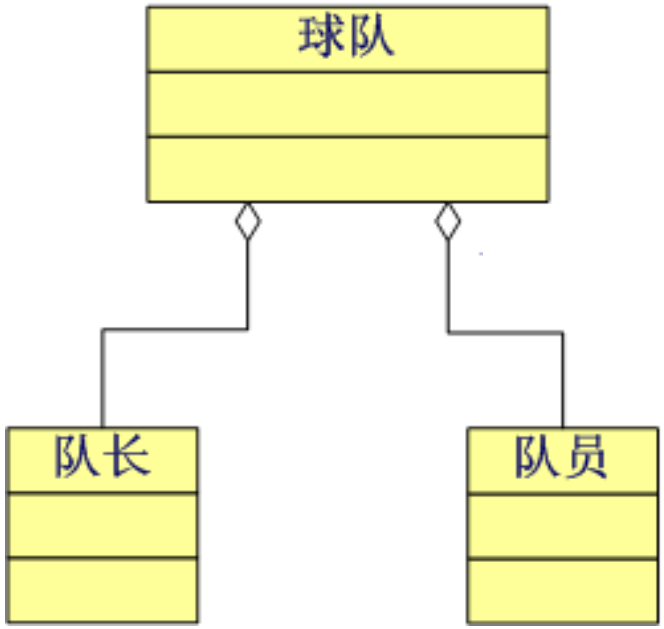
面向对象的基本概念

- 继承关系(XX是一种XX)

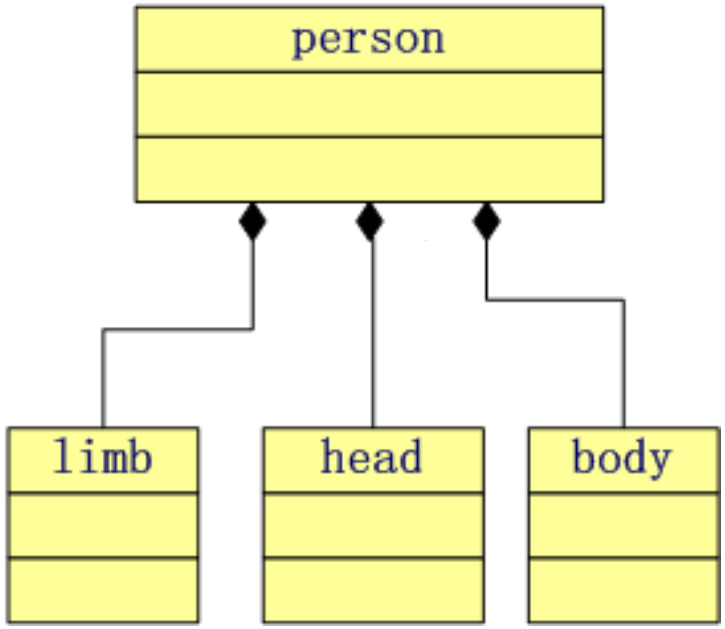


面向对象的基本概念

◆聚合关系（整体和部分）



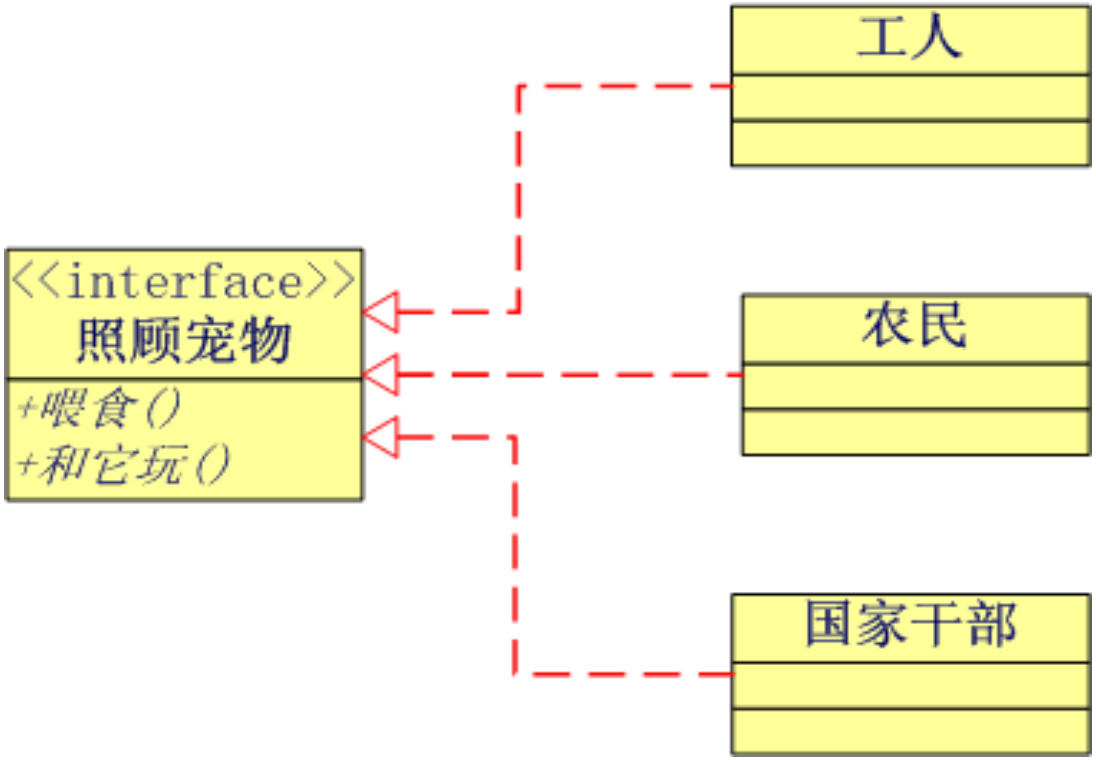
聚集



组合

面向对象的基本概念

◆实现关系



面向对象的基本概念

◆类的语法

- 一个类的实现包括两个部分：类声明和类体

```
[访问限定符] [修饰符] class 类名 [extends 父类名] [implements 接口名列表>]//类声明
{          //类体开始标志
[类的成员变量说明]  //属性说明
[类的构造方法定义]
[类的成员方法定义]  //行为定义
}          //类体结束标志
```

面向对象的基本概念

对类声明的格式说明如下：

◆1）方括号“[]”中的内容为可选项，在下边的格式说明中意义相同，不再重述。

◆2） 访问限定符的作用是：确定该定义类可以被哪些类使用。可用的访问限定符如下：

a)public 表明是公有的。可以在任何Java程序中的任何对象里使用公有的类。该限定符也用于限定成员变量和方法。如果定义类时使用public进行限定，则类所在的文件名必须与此类名相同（包括大小写）

b)private表明是私有的。该限定符可用于定义内部类，也可用于限定成员变量和方法。

c)protected 表明是保护的。只能为其子类所访问。

d)默认访问 若没有访问限定符，则系统默认是友元的 (friendly)。友元的类可以被本类包中的所有类访问。

面向对象的基本概念

对类声明的格式说明如下：

- 3） 修饰符的作用是：确定该定义类如何被其他类使用。可用的类修饰符如下：
  - a)abstract 说明该类是抽象类。抽象类不能直接生成对象。
  - b)final 说明该类是最终类，最终类是不能被继承的。
- 4） class是关键字，定义类的标志（注意全是小写）。
- 5） 类名是该类的名字，是一个Java标识符，含义应该明确。一般情况下单词首字大写。
- 6） 父类名跟在关键字 “extends”后，说明所定义的类是该父类的子类，它将继承该父类的属性和行为。父类可以是Java类库中的类，也可以是本程序或其他程序中定义的类。
- 7） 接口名表是接口名的一个列表，跟在关键字“implements”后，说明所定义的类要实现列表中的所有接口。一个类可以实现多个接口，接口名之间以逗号分隔。如前所述，Java不支持多重继承，类似多重继承的功能是靠接口实现的。

我们先看一个公民类的定义示例。

```
public class Citizen
{
   [ 声明成员变量 ]        //成员变量（属性）说明
   [ 定义构造方法 ]        //构造方法（行为）定义
   [ 定义成员方法 ]        //成员方法（行为）定义
}
```

访问权限修饰符

名称	访问权限修饰符	类本身	子类	本包	所有类
公共	public	√	√	√	√
默认	---	√		√	
保护	protected	√	√	√	
私有	private	√			

面向对象的基本概念

◆属性（成员变量）

- 可以为java中的任何一种数据类型（基本数据类型或引用类型）
- 在定义变量时，可以对其进行初始化；如果不进行初始化，java使用默认值对其进行初始化
- 成员变量的作用域为整个类体

成员变量类型	初始值
byte	0
short	0
int	0
long	0L
float double	0.0F 0.0D
char boolean	'\u0000'（表示为空） False
All reference type	Null

- 声明或定义成员变量的一般格式如下：

[访问限定符] [修饰符]　数据类型 成员变量名 [=初始值];

- 1）访问限定符用于限定成员变量被其它类中的对象访问的权限，和如上所述的类访问限定符类似。

- 2）修饰符用来确定成员变量如何在其他类中使用。可用的修饰符如下：

a)static 表明声明的成员变量为静态的。静态成员变量的值可以由该类所有的对象共享，它属于类，而不属于该类的某个对象。即使不创建对象，使用“类名.静态成员变量”也可访问静态成员变量。

b)final 表明声明的成员变量是一个最终变量，即常量。

- 3）数据类型可以是简单的数据类型，也可以是类、字符串等类型，它表明成员变量的数据类型。

- 类的成员变量在类体内方法的外边声明，一般常放在类体的开始部分。

```
import  java.util.*;
public class Citizen
{
   //以下声明成员变量（属性）
String  name;
String  alias;
String  sex;
Date  brithday;  //这是一个日期类的成员变量
String  homeland;
String  ID;
//以下定义成员方法（行为）
   ……
}
```

- 在上边的成员变量声明中，除出生年月被声明为日期型（Date）外，其他均为字符串型。由于Date类被放在java.util类包中，所以在类定义的前边加上import语句。

面向对象的基本概念

◆**方法定义的一般格式如下：**

```
[访问限定符] [修饰符] 返回值类型   方法名([形式参数表])
[throws 异常表]

{

[ 变量声明 ]              //方法内用的变量，局部变量

[ 程序代码 ]              //方法的主体代码

[ return [ 表达式 ]  ]  //返回语句

}
```

- 1）访问限定符如前所述。

- 2）修饰符用于表明方法的使用方式。可用于方法的修饰符如下：

a)abstract说明该方法是抽象方法，即没有方法体（只有“{}”引起的空体方法）。

b)final说明该方法是最终方法，即不能被重写。

c)static说明该方法是静态方法，可通过类名直接调用。

- 3）返回值类型应是合法的java数据类型。方法可以返回值，也可不返回值，可视具体需要而定。当不需要返回值时，可用void（空值）指定，但不能省略。

- 4）方法名是合法Java标识符，声明了方法的名字。

- 5）形式参数表说明方法所需要的参数，有两个以上参数时，用“，”号分隔各参数，说明参数时，应声明它的数据类型。

- 6）throws 异常表定义在执行方法的过程中可能抛出的异常对象的列表（放在后边的异常的章节中讨论）。

```
/* 这是一个公民类的定义程序
 * 程序的名字是：Citizen.java
*/
import  java.util.*;
public class Citizen
{
   //以下声明成员变量（属性）
String  name;
String  alias;
String  sex;
Date  brithday;  //这是一个日期类的成员变量
String  homeland;
String  ID;
//以下定义成员方法（行为）
   public String  getName()  //获取名字方法
   {        //getName()方法体开始
      return  name;   //返回名字
   }       //getName()方法体结束
   /***下边是设置名字方法***/
   public void setName(String name)
   {        //setName()方法体开始
      this.name=name;
   }       //setName()方法体结束
   /***下边是列出所有属性方法***/
   public void displayAll()
   {     //displayAll()方法体开始
      System.out.println(“姓名：”+name);
      System.out.println(“别名：”+alias);
      System.out.println(“性别：”+sex);
      System.out.println(“出生：”+brithday.toLocaleString());
      System.out.println(“出生地：”+homeland);
      System.out.println(“身份标识：”+ID);
   }   //displayAll()方法体结束
}
```

Lab5_1:练习写一个职员类，包括属性：年龄、姓名、工资；方法：显示年龄、显示姓名、修改姓名、获取工资

对象的创建、使用、销毁

◆对象的创建

- 用new操作符
- 格式：type objectName=new type([paramlist])

  如：Person p = new Person ()的内存分配情况

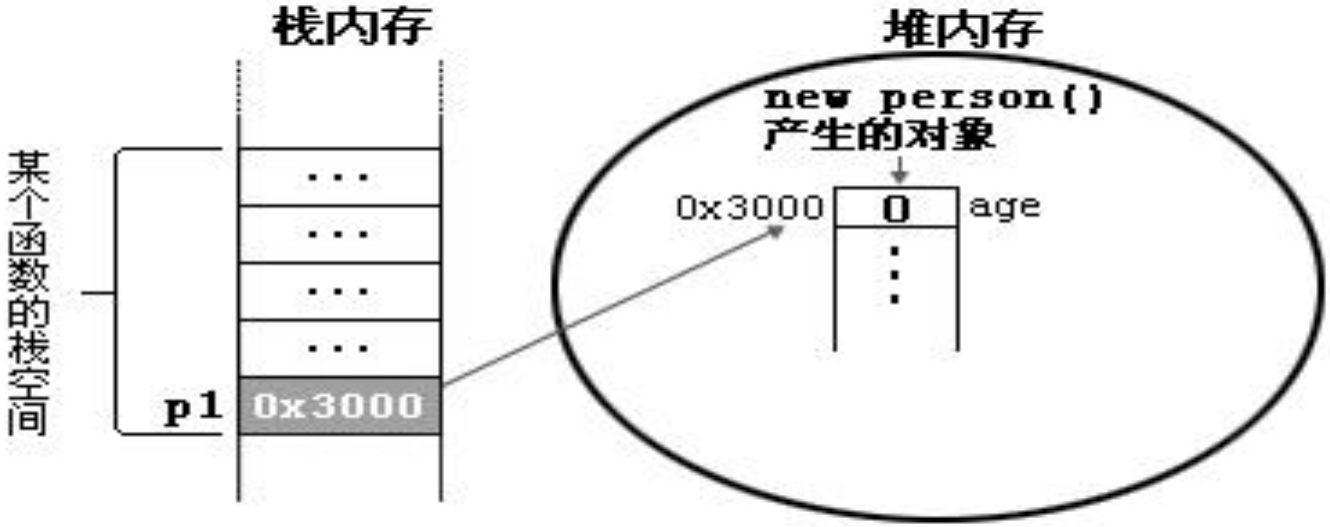
  - p是新对象的引用，其中保存对象在堆中的地址
  - 通过p可以访问堆中的新对象
  - p被称为引用变量
  - 结果：堆中增加了一个Person对象，Java栈中增加了一个引用变量p ， p指向Person对象

对象的创建、使用、销毁

◆ 对象的创建

下图说明了对象在内存中的产生的详细状态

Person p1 = new Person();执行完后的内存状态



对象的创建、使用、销毁

◆ 对象的创建

当一个对象被创建时，会对其中各种类型的成员变量自动进行初始化赋值。除了基本数据类型之外的都是变量类型都是引用类型，如上面的Person及前面讲过的数组。

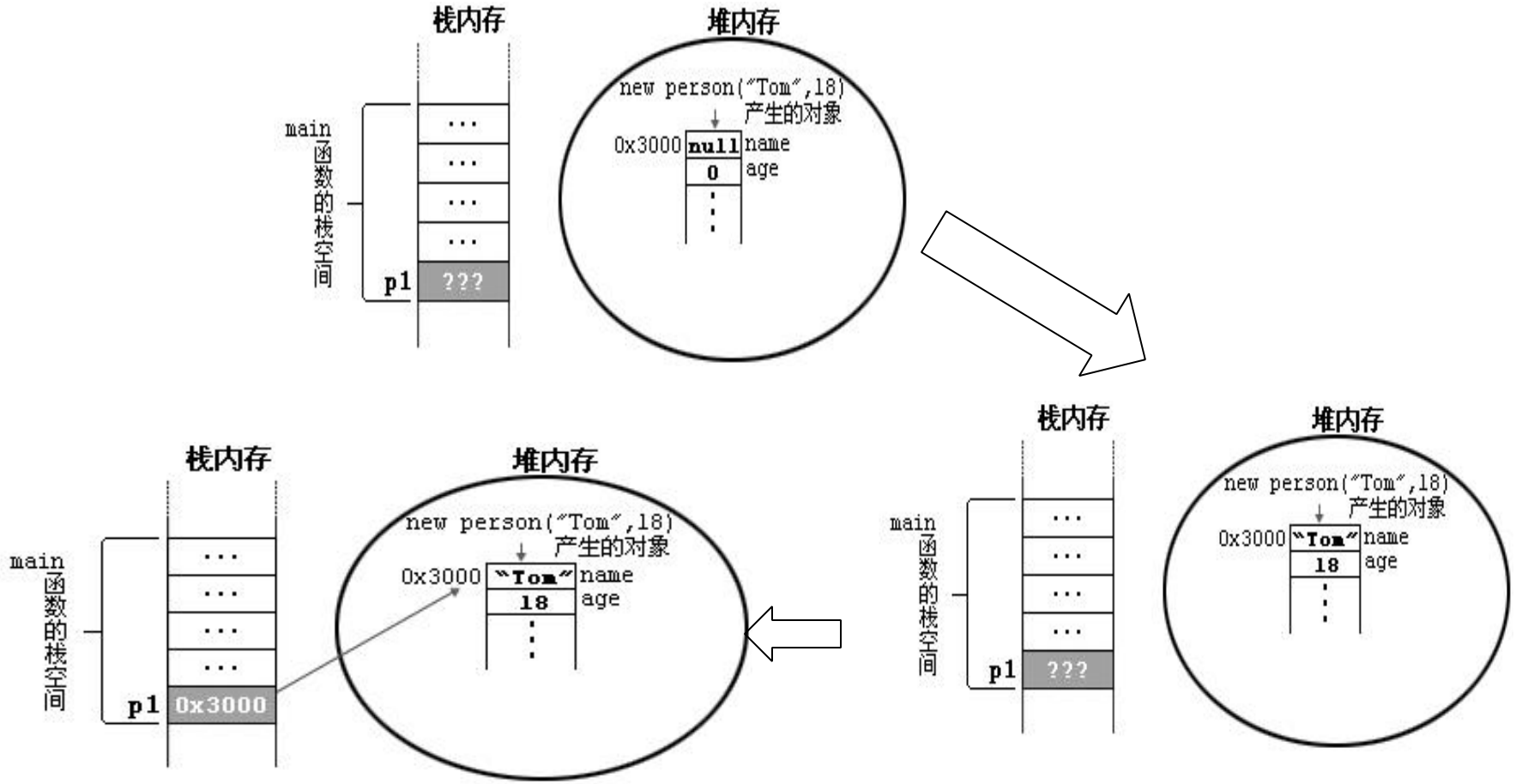
下面是成员变量的初始值

成员变量类型	初始值
byte	0
short	0
int	0
long	0L
float double	0.0F 0.0D
char boolean	'\u0000'（表示为空） False
All reference type	Null

对象的创建、使用、销毁

- 实例化：运算符new为对象分配内存空间，它调用对象的构造方法，返回引用；一个类的不同对象分别占据不同的内存空间。
- 生成：执行构造方法，进行初始化；根据参数不同调用相应的构造方法。

对象的创建、使用、销毁



Person p1=new Person(“Tom”,18)的内存状态变化

对象的创建、使用、销毁

◆对象的使用

- 通过运算符“.”可以实现对变量的访问和方法的调用。变量和方法可以通过设定访问权限来限制其它对象对它的访问 。

格式：objectReference.variable

objectReference是一个已生成的对象，也可以是能生成对象的表达式

- 调用对象的方法

格式：objectReference.methodName([paramlist]);

如： p.getAge();

p.setAge(40);

对象的创建、使用、销毁

◆对象的销毁

当不存在对一个对象的引用时，该对象成为一个无用对象。Java的垃圾收集器自动扫描对象的动态内存区，把没有引用的对象作为垃圾收集起来并释放。

```
System.gc( );
```

当系统内存用尽或调用System.gc( )要求垃圾回收时，垃圾回收线程与系统同步运行。

❖实验：职员类的对象创建、使用

构造方法

- 格式：

[public]　类名（[形式参数列表]）

{

　　[方法体]

}

构造方法

◆ 构造方法的定义

构造方法是一个特殊的方法。Java 中的每个类都有构造方法，用来初始化该类的一个对象。

◆构造方法的特征

它具有与类相同的名称。

它不含返回值。

它不能在方法中用return语句返回一个值。

构造方法只能由new运算符调用 。

重载经常用于构造方法。

访问限定只能使用public或缺省。一般声明为public，如果缺省，则只能在同一个包中创建该类的对象。

构造方法

◆构造方法的作用

当一个类的实例对象刚产生时，这个类的构造方法就会被自动调用，我们可以在这个方法中加入要完成初始化工作的代码。

◆构造方法的一些细节

- 在java每个类里都至少有一个构造方法，如果程序员没有在一个类里定义构造方法，系统会自动为这个类产生一个默认的构造方法，这个默认构造方法没有参数，在其方法体中也没有任何代码，即什么也不做。
- 由于系统提供的默认构造方法往往不能满足编程者的需求，我们可以自己定义类的构造方法，来满足我们的需要，一旦编程者为该类定义了构造方法，系统就不再提供默认的构造方法了。

构造方法

◆构造方法的一些细节

- 声明构造方法，如无特殊需要，应使用public关键字。
- 如果创建子类的对象，则先调用父类的构造方法，然后调用子类的构造方法。

注意：

在构造方法里不含返回值的概念是不同于“void”的，在定义构造方法时加了“void”，结果这个方法就不再被自动调用了。


```
public Citizen(String name,String alias,String sex,Date
brithday,String homeland,String ID)
{

  this.name=name;
  this.alias=alias;
  this.sex=sex;
  this.brithday=brithday;
  this.homeland=homeland;
  this.ID=ID;

 }
```

Lab5_2:修改练习Lab5_2中的职员类，定义其构造函数

总结

计算机学院

北華航天工業學院
NORTH CHINA INSTITUTE OF AEROSPACE ENGINEERING

- 面向对象的基本概念
  - 面向对象的设计思想
  - 类之间的关系
  - 类和对象
- 对象的创建、使用、销毁
  - 构造方法
- 面向对象的特征
  - 封装\继承\多态

计算机学院

北華航天工業學院
NORTH CHINA INSTITUTE OF AEROSPACE ENGINEERING

谢谢！

2021-2022-1