河南中医药大学信息技术学院（智能医疗行业学院）智能医学工程专业《计算机程序设计》课程

第04章：面向对象

黄子杰

河南中医药大学信息技术学院（智能医疗行业学院）与河南方和信息科技有限公司 联合建设
河南中医药大学信息技术学院智能医疗教研室
https://aitcm.hactcm.edu.cn
2025/2/16

1

对象

对象是类的实例，具有状态（属性）和行为（方法）。对象是面向对象编程的基本单元。例如，一个 Person 类的对象可以有 name 和 age 属性，并且可以有 walk() 和 talk() 方法。

- 举个例子

想象一下，周杰伦是你最喜欢的音乐明星，他不仅有他的名字和年龄，还有他的音乐才华和其他行为。

这个例子就像是在编程中定义一个对象一样，我们可以用它来描述周杰伦的特质和行为。

2

- 周杰伦的特质（属性）

名字（name）： 周杰伦的名字就是“Jay Chou”。

年龄（age）： 假设他现在35岁（当然，这个信息会随着时间变化）。

这些特质就像对象的“数据”，它们定义了周杰伦的基本信息。在编程中，这些数据是对象的属性（fields），类似于我们对周杰伦的了解。

- 周杰伦的行为（方法）

周杰伦作为一个对象，不仅有名字和年龄，他还可以做很多事情：

演唱（sing）： 周杰伦在舞台上演唱他的新歌，给大家带来欢乐。在程序中，这个行为可以用一个方法表示，比如 sing() 方法。

作曲（compose）： 周杰伦会创作新的音乐作品，这个行为可以用 compose() 方法表示。

参与电影（act）： 他还可能参与电影拍摄，展示他的演技。在编程中，这可以用 act() 方法表示。

- 周杰伦的日常活动

演出：周杰伦可能会在晚会上演出，为观众带来震撼的表演。在编程中，这就像是对象的方法被调用，周杰伦的 sing() 方法让他在舞台上展示才华。

创作：白天，他可能在工作室里创作新歌，打磨音乐作品。这个过程在程序中类似于对象的方法被调用，例如 compose() 方法。

拍戏：他还可能在拍摄电影，与其他演员一起工作。这相当于在程序中执行另一种行为，通过调用 act() 方法来完成任务。

- 总结：

周杰伦就像一个完美的对象，他不仅有基本的“数据”（名字和年龄），还有丰富的“行为”（演唱、作曲和参与电影）。在编程中，你可以创建一个周杰伦的对象，并通过调用他的各种方法来让他展示不同的才艺和活动。这样一来，就像你在现实中了解和享受周杰伦的才华一样，编程中的对象也让我们能够有效地组织和管理各种信息和功能。

4.1 面向对象的设计思想

- 概述：

面向对象编程（OOP）是一种以“对象”为中心的编程思想，它通过模拟现实世界中的实体及其交互来组织代码。在OOP中，类是定义对象的模板，而对象是类的实例。OOP的四大基本原则是：封装、继承、多态和抽象。

- 封装：通过将数据和操作数据的方法封装在一起，隐藏实现细节，仅对外暴露必要的接口。这提高了代码的安全性和可维护性。
- 继承：子类可以继承父类的属性和方法，从而复用代码，同时支持代码的扩展。
- 多态：不同的对象可以以不同的方式响应相同的方法调用，这使得程序更具灵活性和可扩展性。
- 抽象：通过定义抽象类或接口，将通用的行为抽象出来，让子类去实现具体的细节。

封装

- 封装是将对象的状态（属性）和行为（方法）包装在一个类中，并限制外部对这些属性的直接访问。封装通过访问修饰符（如 private）来保护数据，同时提供公共的方法（getter 和 setter）来访问和修改这些数据。

- 现实例子：

想象你有一个秘密的食谱，你只会给亲密的朋友看，并且用密码锁保护它。

其他人不能随意查看或修改这个食谱，但你可以通过特定的方法来与他们分享。

代码示例：

eg封装.txt

继承

- 继承允许一个类（子类）继承另一个类（父类）的属性和方法。

它促进代码的重用和扩展。

- 现实例子：

想象你有一个动物类（Animal），然后创建了一个更具体的动物类（如猫 Cat）。

猫类继承了动物类的属性和方法，并可以增加自己的特性。

- 代码示例：

eg继承.txt

多态

多态允许不同的类通过相同的接口调用各自不同的实现。它主要通过方法重载（Overloading）和方法重写（Overriding）来实现。

- 现实例子：

想象你有一个远程控制器，可以控制不同的设备（如电视和空调）。虽然你按的都是相同的按钮，但电视和空调会有不同的响应。

- 代码示例：

eg多态.txt

抽象

- 抽象是将对象的共性提取出来，形成一个抽象类或接口。抽象类不能直接实例化，而是作为其他类的基类，提供一组子类应实现的通用方法。

- 现实例子：

想象你有一个“交通工具”抽象概念。虽然不同的交通工具（如汽车和自行车）有不同的具体实现，但它们都可以启动和停止。你定义了一个抽象的交通工具类，具体的交通工具类继承自这个抽象类，并实现具体的启动和停止方法。

- 代码示例：

eg抽象.txt

4.1 面向对象的设计思想

- 生活举例：

面向对象设计：就像你在生活中买一台电脑，尽管你不知道内部的电路是如何工作的，但你知道如何通过按键操作来使用它。这类似于OOP中的封装和抽象。

- 面试题示例：

问：请解释OOP中的四大基本原则。

答：OOP的四大原则是封装、继承、多态和抽象。封装将数据和行为结合在一起，并隐藏实现细节；继承允许一个类继承另一个类的属性和方法；多态允许不同的对象对同一个方法调用有不同的响应；抽象通过定义接口或抽象类来隐藏实现细节。

4.1.1 类的理解

- 生活举例：

类的理解：类就像一个蓝图，比如建筑图纸，它定义了房子的结构和功能，但本身不是房子。
只有通过建造房子（创建对象），图纸才能变成实际的房子。

- 面试题示例：

问：类和对象有什么区别？
答：类是对象的模板，它定义了对象的结构和行为，而对象是类的实例。
类是静态的设计，只有通过实例化才会创建对象。

4.1.2 Java类创建

- 概述：

Java类的创建包括定义类名、属性和方法，并指定类的访问修饰符。
Java类通常存放在一个与类名相同的文件中，并且类名首字母大写。

- 一个 Java 类通常包含以下几个部分：

类名：类的名称，通常以大写字母开头。
属性（字段）：类的成员变量，用于保存对象的状态。
方法：类的功能函数，用于定义对象的行为。
构造方法：用于创建类的实例，并初始化对象的状态。
其他：包括静态方法、静态变量等。

4.1.2 Java类创建

- 生活举例：

Java类创建：就像创建一个特定品牌的汽车模型，我们定义了品牌和速度等属性，以及加速的功能。

- 面试题示例：

问：请描述Java类的基本结构。

答：Java类包括类名、属性和方法，并且通常有一个构造方法来初始化对象。类可以使用访问修饰符来控制其可见性。

4.1.2 Java类创建

- 代码示例：

eg4.1.2.txt

总结

在 Java 中创建类是面向对象编程的基础。类定义了对象的结构和行为，通过属性和方法来描述和操作数据。构造方法用于初始化对象，静态方法和变量提供了与对象无关的功能。通过这些特性，类让我们能够更好地组织和管理复杂的程序。

4.1.3 Java中对象的创建和使用

- 概述：

对象是类的实例，通过new关键字来创建。

对象一旦创建，就可以使用类中定义的属性和方法。

- 代码示例：

想象你在创建一个“书”对象。书有标题（属性）、作者（属性）和内容（方法）。你可以创建一本书的实例，并访问其属性和方法。

eg4.1.3.txt

4.1.3 Java中对象的创建和使用

- 生活举例：

对象的创建和使用：就像你用建筑图纸建造一座房子，房子建好后，你可以在其中生活，使用房子的各项功能。

- 面试题示例：

问：对象如何创建？对象和类有什么关系？

答：对象是通过new关键字创建的，是类的实例。类定义了对象的属性和行为，而对象则是类的具体体现。

4.2 属性

- 概述：

在 Java 中，属性（通常称为字段或成员变量）是类的一部分，用于存储对象的状态。每个对象都有自己的属性值，这些属性值决定了对象的具体状态。包括属性的定义、访问修饰符、默认值、封装、常量属性、静态属性、以及现实中的类比和代码示例。

17

4.2.1 属性的定义

- 概述：

属性的定义包括属性的类型、名称及其初始值。

属性可以使用访问修饰符来控制其可见性。

属性在类中定义，通常与类的构造方法和方法一起使用。属性可以是各种数据类型，如 int、double、String 等。它们可以直接在类中声明和初始化。

18

4.2.1 属性的定义

- 代码示例：

eg4.2.1.txt

4.2.1 属性的定义

- 生活举例：

属性定义：就像人的名字和年龄是每个人的属性一样，类的属性定义了对象的状态。

- 面试题示例：

问：类中的属性能否直接访问？如何控制属性的访问权限？

答：类中的属性可以根据其访问修饰符进行访问控制。通过使用private修饰符，可以限制属性的访问权限，并通过getter和setter方法来操作属性。

4.2.2 变量

- 概述：

在 Java 中，变量是用于存储数据的命名内存位置。它们是编程的基本构建块，用于保存程序中各种类型的数据。根据变量的定义位置和使用方式，Java 中的变量可以分为以下几类：

局部变量（Local Variables）

实例变量（Instance Variables）

类变量（Class Variables，也称为静态变量）

常量（Constants）

1. 局部变量

局部变量是在方法、构造方法或代码块中声明的变量。它们的作用域仅限于所在的代码块或方法内部，一旦代码执行完毕，局部变量就会被销毁。

- 特点：

局部变量必须在使用前初始化。

作用域仅限于定义它们的代码块内。

局部变量不使用任何访问修饰符。

代码示例：

eg局部变量.txt

2. 实例变量

实例变量是在类中声明的，但不在方法或构造方法中。它们的值随对象的创建而存在，并且每个对象有自己的一份实例变量的拷贝。

- 特点：

实例变量在类中声明，并在构造方法中初始化。

它们在类中有一个默认值：例如，数字为 0，布尔值为 false，引用类型为 null。

实例变量可以使用访问修饰符（private、public、protected）来控制访问权限。

- 代码实例：

eg实例变量.txt

3. 类变量（静态变量）

类变量（或静态变量）是在类中使用 static 关键字声明的变量。类变量在类的所有实例中共享，也就是说，无论创建多少个对象，类变量只有一份。

- 特点：

类变量在类加载时初始化。

它们可以通过类名直接访问，也可以通过对象访问。

类变量通常用于存储所有对象共享的数据。

代码示例：

eg静态变量.txt

4. 常量

常量是用 final 关键字声明的变量，一旦赋值后，值就不能再修改。常量通常与 static 关键字一起使用，表示类级别的常量。

- 特点：

常量必须在声明时初始化，或者在构造方法中赋值。

常量的值不能被改变。

常量通常使用全大写字母命名，以区分于普通变量。

- 代码示例：

eg常量.txt

5. 变量的命名规范

- 在 Java 中，变量的命名应该遵循以下规则和规范：

变量名区分大小写。

变量名必须以字母、美元符号（$）、或下划线（_）开头，不能以数字开头。

变量名不能是 Java 的关键字（如 int、class 等）。

变量名应当具有描述性，以便代码的可读性。例如，使用 age 而不是 a，使用 totalCars 而不是 tC。

- 代码示例：

eg变量命名规范.txt

4.2.2 变量

字符串

- 面试题示例：

问：局部变量和实例变量有什么区别？

答：局部变量只在方法或块中有效，而实例变量属于对象，可以在类的整个生命周期内使用。

4.3 方法

String 类

- 概述：

方法是定义在类中的函数，用于执行某种操作或返回某个值。

方法包括方法名、参数列表、返回类型和方法体。

- 注意：

方法必须先创建才可以使用，该过程成为方法定义

方法创建后并不是直接可以运行的，需要手动使用后，才执行，该过程成为方法调用

4.3.1 方法的定义

- 概述：

方法的定义包括访问修饰符、返回类型、方法名、参数列表和方法体。
方法的返回类型可以是任意类型，包括void（无返回值）。

1、无参数方法定义和调用

- 定义格式：

```
public static void 方法名 (  ) {
    // 方法体;
}
```

- 调用格式：

```
方法名();
```

2、带参数方法的定义和调用

- 定义格式：

参数：由数据类型和变量名组成 - 数据类型 变量名
参数范例：int a

```
public static void 方法名 (参数1) {
    方法体;
}
public static void 方法名 (参数1, 参数2, 参数3...) {
    方法体;
}
```

- 调用格式：

```
方法名(参数);
方法名(参数1,参数2);
```


3、带返回值方法的定义和调用

- 定义格式

```
public static 数据类型 方法名 ( 参数 ) {
        return 数据 ;
}
```

- 调用格式

```
方法名 ( 参数 ) ;
数据类型 变量名 = 方法名 ( 参数 ) ;
```

4.3.1 方法的定义

- 代码示例：

eg4.3.1.txt

4.3.1 方法的定义

- 生活举例：

方法定义：就像你每天做饭的过程，方法就是做饭的步骤，参数是食材，返回的菜就是返回值。

- 面试题示例：

问：Java中的方法有哪几部分组成?

答：Java中的方法由访问修饰符、返回类型、方法名、参数列表和方法体组成。

4.3.2 构造方法

- 概述：

构造方法是用于初始化对象的特殊方法，名称与类名相同，无需定义返回类型。

构造方法可以重载以支持不同的初始化方式。

- 1、无参构造方法

无参构造方法是指没有参数的构造方法，它在类中定义一个新的对象实例，但不需要传入任何参数。一般情况下，无参构造方法用来初始化对象的属性，给属性赋默认值。 如果一个类没有定义构造方法，Java 会自动提供一个无参构造方法。如果一个类中已经定义了有参构造方法，但也需要使用无参构造方法，则必须显式地在类中定义一个无参构造方法。

- 2、带参构造方法：

带参构造方法是用来在创建对象的同时为对象的属性赋初值的一种方法，可以通过定义多个带参构造方法来为对象提供不同的初始化方式。如果在编写类时只添加了带参构造方法而未添加无参构造方法，那么编译器会默认为只能使用带参构造方法进行对象的创建，无法使用无参构造方法创建对象。因此，如果需要使用无参构造方法，需要在类里面手动添加一个。相比之下，无参构造方法则是没有参数的构造方法。在Java中，默认情况下，JVM会为每个类提供一个无参构造方法。

4.3.2 构造方法

- 代码示例

eg4.3.2.txt

4.3.2 构造方法

- 生活举例：

构造方法：购买新车时，你会指定品牌和速度等参数，这就像构造方法初始化对象时指定的属性值。

- 面试题示例：

问：构造方法和普通方法有什么区别？

答：构造方法用于初始化对象，名称与类名相同，没有返回类型，而普通方法用于定义对象的行为，有返回类型。

4.3.3 方法重载

- 概述：

方法重载是指在同一个类中定义多个方法，它们具有相同的名称但参数列表不同。

重载方法根据传入的参数类型和数量进行区分。

4.3.3 方法重载

- 代码示例

eg4.3.3.txt

4.3.3 方法重载

- 生活举例：

方法重载：就像你可以用同一个名字来称呼不同的东西，
比如“苹果”可以指水果或电子产品，具体意义取决于上下文。

- 面试题示例：

问：方法重载和方法重写有什么区别？ （重点）
答：方法重载是同一类中方法名相同但参数不同，方法重写是子类对父类方法的重新定义，方法签名相同。

4.3.4 方法调用

- 概述：

方法调用是通过对象调用其类中定义的方法。
方法调用时可以传递参数，并接收返回值。

4.3.4 方法调用

- 面试题示例：

问：如何调用静态方法与实例方法?
答：静态方法可以通过类名直接调用，实例方法需要通过对象来调用。

- 代码示例

eg4.3.4.txt

4.3.5 方法参数及其传递问题

- 概述：

Java中的方法参数分为值传递和引用传递。
Java中所有的参数传递都是值传递，对于对象引用，传递的是对象的地址。

4.3.5 方法参数及其传递问题

- 面试题示例：

问：Java中参数传递是值传递还是引用传递？

答：Java中参数传递是值传递，即传递的是参数的副本。但对于对象引用，传递的是引用的副本。

- 代码示例

eg4.3.5.txt

4.3.6 理解main方法语法及命令行参数

概述：

main方法是Java程序的入口点。
它的定义是public static void main(String[] args)。
命令行参数是通过args数组传递给main方法的。

4.3.6 理解main方法语法及命令行参数

- 代码示例

eg4.3.6.txt

4.3.6 理解main方法语法及命令行参数

- 生活举例：

main方法：就像启动汽车，点火是整个流程的开始，main方法是Java程序的启动点。

- 面试题示例：

问：为什么main方法必须是static?

答：main方法是程序的入口点，它在类加载时被JVM调用，必须是static以便JVM不需要创建类的实例就可以调用它。

4.3.7 递归算法

- 概述：

递归算法是指方法调用自身来解决问题的方式。
递归通常包括基准情况和递归步骤。

- 直接递归：

方法自己调用自己

- 间接递归：

方法调用其他方法，其他方法又回调方法自己。

- 递归存在的问题？

递归如果没有控制好终止，会出现递归死循环，导致栈内存溢出现象。

4.3.7 递归算法

- 代码示例

eg4.3.7.txt

- 课后题：

计算1-n的阶乘的结果，使用递归思想解决，我们先从数学思维上理解递归的流程和核心点。

eg递归问题答案.txt

4.3.7 递归算法

- 生活举例：

递归：就像照镜子时看到的无限镜像，递归不断调用自身，直到达到基准情况。

- 面试题示例：

问：递归算法的优缺点是什么？

答：递归算法的优点是简洁且容易理解，但如果递归层次过多，会导致栈溢出，使用递归时要注意基准情况。

本章重点

- 面向对象的四大基本原则(继承,封闭,多态,抽象)
- Java 中的变量(局部变量,实例变量,类变量,常量)
- 对象的创建和使用(new)
- 方法的定义和使用(修饰符、返回类型、方法名、参数列表和方法体)
- 构造方法(无参,带参)

本章作业

- 实现一个医生问诊和查房的实例;

要求:

1 程序包含Hospital类、Patient 类、Doctor 类、Ward类

Hospital类为主类,包含一个main方法

2 在这个程序中，Patient类代表病人，有姓名和病情两个属性，以及一个描述病情的方法。Doctor类代表医生，有姓名属性，以及根据病情给出建议的方法， 结合方法重载写出两个场景，一个场景是根据病情字符串给出建议，另一个场景是直接接受一个病人对象，并调用病人的方法来获取病情描述。Ward类代表病房，管理医生和病人，并提供医生查房的功能。最后，在 Hospital 类的 main 方法中，创建医生和病房的实例，添加病人，并模拟医生查房的过程，同时展示方法重载的使用。

信创智能医疗系统研发课程体系

河南中医药大学信息技术学院（智能医疗行业学院）

河南中医药大学信息技术学院（智能医疗行业学院）智能医疗教研室

河南中医药大学医疗健康信息工程技术研究所