# 第6章：面向对象

【考点梳理】

## 考点1、面向对象的概念（★★★★★）

### 【考法分析】

#### 本知识点的考查形式主要是给定相关的一些概念描述判断正误；或根据描述指出对应的概念。

### 【要点分析】

#### 1、基本概念：

#### （1）对象：属性（数据）+方法（操作）+对象ID

#### （2）类（实体类/控制类/边界类）

#### （3）接口：一种特殊的类，他只有方法定义没有实现

#### （4）封装：隐藏对象的属性和实现细节,仅对外公开接口，是一种信息隐藏的机制。

#### （5）多态：不同对象收到同样的消息产生不同的结果。多态实质上是将子类的指针对象或者引用对象传递给父类指针对象后，通过这个父类指针对象调用的函数（此函数在父类中声明为虚函数，且在各个子类中重写这个函数），不是父类中定义的，而是传递进来的子类对象中重写的函数。

#### 过载多态：同一个名（操作符﹑函数名）在不同的上下文中有不同的类型。

#### （6）重载：一个类可以有多个同名而参数类型不同的方法

#### （7）继承与泛化：复用机制。面向对象技术中，继承是父类和子类之间共享数据和方法的机制。这是类之间的一种关系，在定义和实现一个类的时候，可以在一个已经存在的类的基础上来进行，把这个已经存在的类所定义的内容作为自己的内容，并加入若干新的内容。可以存在多重继承的概念，但不同的程序设计语言可以有自己的规定。

#### （8）消息和消息通信：消息是异步通信的

#### 2、面向对象设计原则：

#### （1）单一职责原则：设计目的单一的类

#### （2）开放-封闭原则：对扩展开放，对修改封闭

#### （3）李氏（Liskov）替换原则：子类可以替换父类

#### （4）依赖倒置原则：要依赖于抽象，而不是具体实现；针对接口编程，不要针对实现编程

#### （5）接口隔离原则：使用多个专门的接口比使用单一的总接口要好

#### （6）组合重用原则：要尽量使用组合，而不是继承关系达到重用目的

#### （7）迪米特（Demeter）原则（最少知识法则）：一个对象应当对其他对象有尽可能少的了解

### 【备考点拨】

#### 1、掌握面向对象相关的基本概念。

## 考点2、UML（★★★★★）

### 【考法分析】

#### 本知识点的主要考查形式有：给定图示判断具体属于哪种UML图；给定UML图，判断属于哪一类（结构图或行为图）；给定UML图，指出相关的一些参数；给出一些关于UML的描述，判断正误。

### 【要点分析】

#### 1、UML图分类：

#### 

#### （早期对于用例图也有放到结构图一类）

#### 2、用例图：用例图描述一组用例、参与者及它们之间的关系。

#### 用例之间的关系：

#### 包含关系：其中这个提取出来的公共用例称为抽象用例，而把原始用例称为基本用例或基础用例系：当可以从两个或两个以上的用例中提取公共行为时，应该使用包含关系来表示它们。

#### 扩展关系：如果一个用例明显地混合了两种或两种以上的不同场景，即根据情况可能发生多种分支，则可以将这个用例分为一个基本用例和一个或多个扩展用例，这样使描述可能更加清晰。

#### 泛化关系：当多个用例共同拥有一种类似的结构和行为的时候，可以将它们的共性抽象成为父用例，其他的用例作为泛化关系中的子用例。在用例的泛化关系中，子用例是父用例的一种特殊形式，子用例继承了父用例所有的结构、行为和关系。

#### 3、类图（class diagram）:类图描述一组类、接口、协作和它们之间的关系。在OO系统的建模中，最常见的图就是类图。类图给出了系统的静态设计视图，活动类的类图给出了系统的静态进程视图。

#### 对象图（object diagram）:对象图描述一组对象及它们之间的关系。对象图描述了在类图中所建立的事物实例的静态快照。和类图一样，这些图给出系统的静态设计视图或静态进程视图，但它们是从真实案例或原型案例的角度建立的。

#### 

#### 类之间的关系：

#### 依赖关系：一个事物发生变化影响另一个事物。

#### 泛化关系：特殊/一般关系

#### 关联关系：描述了一组链，链是对象之间的连接。

#### 聚合关系：整体与部分生命周期不同。

#### 组合关系：整体与部分生命周期相同。

#### 实现关系：接口与类之间的关系

#### 4、顺序图（sequence diagram，序列图）。顺序图是一种交互图（interaction diagram），交互图展现了一种交互，它由一组对象或参与者以及它们之间可能发送的消息构成。交互图专注于系统的动态视图。顺序图是强调消息的时间次序的交互图。

#### 

#### 5、活动图（activity diagram）。活动图将进程或其他计算结构展示为计算内部一步步的控制流和数据流。活动图专注于系统的动态视图。它对系统的功能建模和业务流程建模特别重要，并强调对象间的控制流程。

#### http://www.educity.cn/tiku/uploadfiles/2017-10/5105b0624f164a1498ac6caaf96bf745_.png

#### 6、状态图（state diagram）。状态图描述一个状态机，它由状态、转移、事件和活动组成。状态图给出了对象的动态视图。它对于接口、类或协作的行为建模尤为重要，而且它强调事件导致的对象行为，这非常有助于对反应式系统建模。

#### 

#### 7、通信图（communication diagram）。通信图也是一种交互图，它强调收发消息的对象或参与者的结构组织。顺序图和通信图表达了类似的基本概念，但它们所强调的概念不同，顺序图强调的是时序，通信图强调的是对象之间的组织结构（关系）。

#### 

#### 8、构件图（component diagram）。构件图描述一个封装的类和它的接口、端口，以及由内嵌的构件和连接件构成的内部结构。构件图用于表示系统的静态设计实现视图。对于由小的部件构建大的系统来说，构件图是很重要的。构件图是类图的变体。

#### 

#### 9、部署图（deployment diagram）。部署图描述对运行时的处理节点及在其中生存的构件的配置。部署图给出了架构的静态部署视图，通常一个节点包含一个或多个部署图。

#### 

### 【备考点拨】

#### 1、掌握各类UML图的特点和适用情景，能够加以区分。对一些特殊的图的特殊部分能够辨认区分。

## 考点3、设计模式（★★★★★）

### 【考法分析】

#### 1、本知识点的主要考查形式有： 根据图示题干描述和图示指出对应的设计模式（中文或英文形式），指出该设计模式的分类，或该设计模式的适用场景。

### 【要点分析】

#### 1、设计模式的分类：

#### 

#### 2、设计模式应用场景和记忆关键字：

#### （1）创建型模式

#### 

#### （2）结构型模式

#### 

#### （3）行为型模式

#### 

### 【备考点拨】

#### 1、掌握23种设计模式的中英文；

#### 2、掌握23种设计模式的分类；

#### 3、根据关键字，掌握23种设计模式的应用场景、相关描述；

4、了解23种设计模式对应的UML图示。