

# 大学面向对象课程试卷

考试时间：120分钟

满分：65分

## 一、选择题（共10题，每题2分，共20分）

1. 在面向对象编程中，\_\_\_\_\_是将数据和基于数据的操作封装在一起的实体。
  - A. 类
  - B. 方法
  - C. 属性
  - D. 接口
2. 以下哪个不是继承的主要优点？
  - A. 提高代码复用性
  - B. 减少冗余代码
  - C. 增强程序的可扩展性
  - D. 提高程序运行速度
3. 当一个类实现了一个接口时，意味着该类必须提供什么？
  - A. 所有非抽象方法的具体实现
  - B. 至少一个构造函数
  - C. 静态成员变量
  - D. 私有方法
4. 关于多态性的描述，正确的是：
  - A. 允许子类重写父类的方法
  - B. 仅限于同一包内的类使用
  - C. 使不同的对象能够响应相同的消息
  - D. 以上全部
5. 下面哪项最能体现封装的原则？
  - A. 将相关的类放在同一个文件夹下
  - B. 通过公共方法访问私有字段
  - C. 确保每个类都有一个主方法
  - D. 创建尽可能多的对象
6. 如果两个类之间存在'is-a'关系，则应该使用哪种机制？
  - A. 组合
  - B. 聚合
  - C. 继承
  - D. 关联

7. 下列选项中，哪一个不是Java中的访问修饰符？

- A. public
- B. private
- C. protected
- D. defaulted

8. 在OOP中，\_\_\_\_\_是一种用于描述现实世界概念的工具，它由一组属性和方法组成。

- A. 类
- B. 实例
- C. 变量
- D. 函数

9. 假设有一个Animal类，Dog类从Animal类继承而来。那么，以下说法正确的是？

- A. Dog不能拥有自己的属性
- B. Animal一定是抽象类
- C. Dog可以直接访问Animal的所有方法
- D. Dog类实例化后也可以作为Animal类型处理

10. 在设计模式中，\_\_\_\_\_模式用于定义一个接口，让其他类来实现这个接口。

- A. 工厂
- B. 单例
- C. 策略
- D. 模板方法

## 二、填空题（共5题，每题3分，共15分）

1. 在面向对象编程里，当我们说一个对象是另一个对象的\_\_\_\_\_时，我们指的是后者拥有前者的一个引用。

2. 当一个类实现了多个接口时，这种情况被称为\_\_\_\_\_。

3. \_\_\_\_\_原则指出，软件实体（类、模块、函数等）应该对扩展开放，对修改关闭。

4. 为了保证对象状态的一致性和有效性，开发者常常会在类中编写\_\_\_\_\_方法来验证设置属性时传入的值。

5. \_\_\_\_\_是一种设计模式，它提供了创建对象的最佳方式，同时隐藏了创建逻辑。

## 三、解答题（共3题，每题10分，共30分）

**1. 请解释什么是封装，并给出一个简单的例子说明如何在Java中实现封装。**

**2. 简述抽象类与接口的区别，并讨论何时应该使用抽象类，何时应该使用接口。**

**3. 请详细描述一下单例设计模式，并举例说明其实际用途。**

## 参考答案及解析

### 一、选择题答案

1.

答案：A

解析：

类定义了对象的数据结构以及可以对这些数据执行的操作。

2.

答案：A

解析：

虽然良好的设计可以通过减少重复代码来间接影响性能，但直接提高程序运行速度并不是继承的主要目标。

3.

答案：A

解析：

实现接口要求类为接口中声明的所有方法提供具体的实现。

4.

答案：A

解析：

多态性的一个关键特性就是能够让不同类型的对象以各自的方式响应同一个消息或方法调用。

5.

答案：A

解析：

封装是指隐藏对象内部的状态，并强制外部通过特定的方法来与之交互，这有助于保护数据完整性并简化复杂性。

6.

答案：A

解析：

'is-a'关系通常表示一种继承关系，其中一个类（子类）是从另一个类（父类）派生出来的。

7.

答案：A

解析：

Java中的四个访问级别分别是public、private、protected以及默认（没有关键字）。

8.

答案：A

解析：

类是面向对象编程的基本构建块之一，用来模拟真实世界的实体。

9.

答案：A

解析：

由于Dog是从Animal派生出来的，因此它可以被视为Animal的一种特殊形式。这意味着Dog对象可以被赋值给Animal类型的引用。

10.

答案：A

解析：

策略模式允许客户端在运行时选择算法的行为，而不需要修改包含算法的对象。

## 二、填空题答案

1.

答案：属性

解析：

这是组合关系的一种表现形式，在这种情况下，一个对象作为另一个对象的成员变量存在。

2.

答案：多重继承

解析：

尽管某些语言如Java不支持类之间的多重继承，但它们确实允许一个类实现多个接口。

3.

答案：开闭

解析：

这一原则鼓励程序员通过添加新功能而不是更改现有代码来扩展软件系统的能力。

4.

答案：setter

解析：

Setter方法不仅用于设置属性值，还可以在此过程中加入必要的校验逻辑。

5.

答案：工厂

解析：

工厂模式帮助分离对象的创建过程与其使用过程，使得代码更加灵活且易于维护。

### 三、解答题答案

1.

答案：封装是面向对象编程的核心概念之一，指将数据（属性）和操作数据的方法绑定在一起形成一个单独单元（即类），并且控制对数据的访问。这样做的目的是保护数据不受外部直接修改，同时提供了统一的接口供外部使用。例如，我们可以定义一个名为**Person**的类，其中包含姓名(**name**)和年龄(**age**)两个私有属性，然后提供相应的**getter**和**setter**方法来获取或设置这些属性的值。这样的设计既隐藏了内部实现细节又提供了对外部友好的接口。

解析：

此题考察学生对于封装概念的理解及其实践能力。

2.

答案：抽象类和接口都是用来实现多态特性的手段，但它们之间存在一些重要区别：- 抽象类可以包含具体的方法实现，而接口只能包含方法签名。- **Java**中，一个类只能继承自一个抽象类，但却可以实现多个接口。- 抽象类可用于表示‘**is-a**’关系，而接口更多地用来表达‘**can-do**’关系。选择使用抽象类还是接口取决于具体场景：- 当需要共享代码时，应考虑使用抽象类。- 如果只是想定义行为规范而不关心具体实现，则更倾向于使用接口。

解析：

本题旨在测试考生是否理解抽象类与接口的不同应用场景及其背后的设计理念。

3.

答案：单例模式是一种常用的软件设计模式，它保证一个类只有一个实例，并提供一个全局访问点。实现单例模式通常涉及以下几个步骤：**1.** 将类的构造器设为私有的，防止外部直接创建实例。**2.** 在类内部创建一个静态私有变量，用于保存唯一的实例。**3.** 提供一个静态公有方法返回该实例。实际应用中，单例模式常用于管理应用程序范围内的资源，比如数据库连接池、配置信息加载等场景。因为这类资源往往只需要初始化一次并在整个应用生命周期内保持不变。

解析：

题目意在检验学生对于单例模式工作原理及其实用价值的认识程度。