

大学面向对象课程设计试卷

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，下列哪个概念不是用来描述类与类之间的关系？
 - A. 继承
 - B. 封装
 - C. 多态
 - D. 关联
2. 关于抽象类的描述，哪一项是不正确的？
 - A. 抽象类可以有实例化对象
 - B. 抽象类可以包含抽象方法
 - C. 抽象类可以被继承
 - D. 抽象类不能直接创建对象
3. 下面哪种说法正确表达了接口与抽象类的区别？
 - A. 接口可以包含构造函数
 - B. 一个类可以实现多个接口但只能继承一个抽象类
 - C. 接口中可以包含字段
 - D. 抽象类不允许有具体实现的方法
4. 以下哪个选项最能体现多态性的优点？
 - A. 提高程序运行效率
 - B. 增加代码复杂度
 - C. 减少代码重复性
 - D. 简化系统结构
5. 在设计模式中，工厂模式主要解决的问题是？
 - A. 避免子类过多
 - B. 对象的创建过程解耦
 - C. 提高性能
 - D. 增强安全性
6. 单例模式确保一个类只有一个实例，并提供一个全局访问点给该实例。实现单例模式的关键步骤是什么？
 - A. 私有构造函数
 - B. 静态方法
 - C. 公共属性
 - D. 继承

7. 观察者模式属于哪种类型的设计模式？

- A. 创建型
- B. 结构型
- C. 行为型
- D. 策略型

8. 适配器模式的主要目的是什么？

- A. 替换旧的API
- B. 优化算法效率
- C. 使不兼容的对象能够协同工作
- D. 管理资源分配

9. 装饰者模式与继承相比，在扩展功能方面有什么优势？

- A. 更易于添加新功能
- B. 减少了代码量
- C. 提高了执行速度
- D. 增强了安全性

10. 桥接模式主要用于解决什么问题？

- A. 减少类的数量
- B. 解除抽象和实现之间的耦合
- C. 提高模块间的通信效率
- D. 降低内存消耗

二、填空题（共5题，每题3分，共15分）

1. 在面向对象编程中，_____是指一个类可以从另一个类派生出新的类。

2. 如果一个类中有未实现的方法，则该类必须声明为_____。

3. _____模式允许你定义一系列可互换使用的算法，并将每种算法分别封装起来。

4. _____模式提供了对‘一对多’依赖关系的对象进行一致处理的方式，其中一个对象的状态发生变化时，所有依赖于它的对象都会得到通知。

5. 在Java中，如果你想确保一个类只能有一个实例存在，并且提供了一个全局访问点来获取这个唯一的实例，那么你应该使用_____模式。

三、解答题（共3题，每题10分，共30分）

1. 请解释什么是封装，并举例说明如何在Java中实现封装。

2. 简述代理模式的工作原理，并给出至少两种应用场景。

3. 详细描述工厂方法模式，并与简单工厂模式进行对比。

参考答案及解析

一、选择题答案

1.

答案：B

解析：

封装是面向对象的一个基本特性，它指的是将数据和操作这些数据的方法捆绑在一起，并且隐藏了实现细节。而继承、多态、关联都是描述类之间关系的概念。

2.

答案：A

解析：

抽象类是用来定义其他类的基础类，它自身不能被实例化。但是它可以拥有抽象方法以及非抽象的方法。

3.

答案：B

解析：

Java等语言中，一个类只能继承自一个父类（包括抽象类），但可以实现多个接口。接口内不允许有构造器或字段，所有方法默认为public abstract；而抽象类则更加灵活，可以有具体的成员变量及方法。

4.

答案：C

解析：

多态允许我们使用统一的方式处理不同类型的对象，从而减少代码中的冗余部分，使得程序更加简洁易维护。

5.

答案：B

解析：

工厂模式通过提供一个创建对象的接口来让子类决定实例化哪一个类，从而达到将对象的创建逻辑与使用逻辑分离的目的。

6.

答案：A

解析：

单例模式的核心在于控制类的实例化次数，通常通过将构造函数设为私有(private)并提供一个静态方法来获取唯一实例来实现这一点。

7.

答案：C

解析：

观察者模式是一种行为设计模式，它定义了一种订阅机制，使得当一个对象状态改变时，所有依赖于它的对象都会得到通知并自动更新。

8.

答案：C

解析：

适配器模式的作用是在保持原有接口不变的情况下，转换另一个接口，以便两个原本不兼容的接口能够一起工作。

9.

答案：A

解析：

装饰者模式可以在运行时动态地给一个对象添加新的功能，而不需要修改原有的代码或使用继承带来的固定层次结构限制。

10.

答案：B

解析：

桥接模式旨在将抽象部分与其实现部分分离开来，使得它们都可以独立变化。这样做的好处是可以让不同的抽象和实现自由组合，增加了系统的灵活性。

二、填空题答案

1.

答案：继承

解析：

继承是面向对象编程的基本特性之一，它支持代码重用，并且有助于建立类之间的层次关系。

2.

答案：抽象类

解析：

含有抽象方法的类必须被声明为抽象类，因为抽象方法没有具体实现，需要由子类来完成。

3.

答案：策略

解析：

策略模式定义了一系列算法，并将每个算法封装起来，使它们可以互相替换。这种模式让算法的变化独立于使用算法的客户。

4.

答案：观察者

解析：

观察者模式定义了一种订阅机制，当某个对象状态发生变化时，会自动通知所有关注该对象状态变化的其他对象。

5.

答案：单例

解析：

单例模式保证一个类只有一个实例，并提供一个全局访问点来访问这个实例。这在需要控制资源访问或者共享同一份配置信息时非常有用。

三、解答题答案

1.

答案：封装是面向对象编程(OOP)三大特性之一，其目的是隐藏对象内部的具体实现细节，只对外暴露必要的接口供外部调用。这样做不仅保护了数据的安全性，也降低了程序间的耦合度。在Java中，可以通过设置类成员变量为private访问级别，并通过public修饰符提供getter/setter方法来实现封装。例如：`` `java public class Person { private String name; private int age; public String getName() { return name; } public void setName(String name) { this.name = name; } public int getAge() { return age; } public void setAge(int age) { if (age > 0) { this.age = age; } else { System.out.println("年龄必须大于零!"); } } } `` 上面的例子中，`name`和`age`都是私有的，外界无法直接访问这两个变量，但可以通过提供的`get`和`set`方法来进行读写操作。

解析：

本题考察学生对于封装概念的理解及其在实际编程中的应用能力。通过编写一个简单的例子，展示了如何利用访问控制符加上公共访问方法来实现封装。

2.

答案：代理模式是一种结构型设计模式，它提供了一个代理对象来控制对真实对象的访问。当客户端请求服务时，不是直接向目标对象发出请求，而是通过代理对象转发请求，代理对象可以选择性地执行额外的操作如权限验证、日志记录等，然后再将请求转交给目标对象处理。应用场景示例：1. 远程代理 - 用于表示位于不同地址空间的对象，如Web服务。2. 虚拟代理 - 当创建开销较大的对象时，可以用一个轻量级的代理代替实际对象，直到真正需要时再创建实际对象。

解析：

此题旨在测试学生是否理解代理模式的基本思想及其常见用途。通过列举具体的场景帮助加深理解。

3.

答案：工厂方法模式是一种创建型设计模式，它定义了一个创建对象的接口，但让子类决定实例化哪一

个类。工厂方法模式让类把实例化推迟到子类中进行。 - ****工厂方法模式****: - 定义了一个创建对象的接口，但由子类决定要实例化的类是哪一个。 - 优点：遵循开放封闭原则，即对扩展开放而对修改关闭。 - 缺点：每增加一种产品就需要增加相应的具体产品类和对应的工厂子类。 - ****简单工厂模式****: - 直接根据传入参数的不同返回不同的实例对象。 - 优点：简单易懂，易于实现。 - 缺点：违反了开闭原则，每当需要新增加一种产品时，都需要修改工厂类。两者的主要区别在于工厂方法模式中具体的产品创建是由子类完成的，而简单工厂模式则是集中在一个地方完成所有的对象创建。

解析：

题目要求学生掌握工厂方法模式的概念及其与简单工厂模式之间的差异。通过对两种模式优缺点的比较，帮助学生更好地理解它们各自的适用场景。