中南大学考试试卷

2017--2018 学年 上 学期　　时间：90 分钟　试卷类型：A 卷

设计模式 课程 32 学时 2 学分 考试形式：闭 卷

专业年级： 软件工程 2015　总分 100 分，占总评成绩 60%

注：1. 此页不作答题纸，请将答案写在答题纸上

2. 试题、答题纸一并交

1. (3 分)Functions that use pointers or references to base classes must be able to use objects of derived classes without knowing it. This is a definition of (　　).

A. Open-Closed Principle　　**B. Liskov Substitution Principle**

C. Single Responsibility Principle　　D. Interface Segregation Principle

2. (3 分)类 ClassA 包含 ClassB 类型的对象，但不能控制 ClassB 对象的生命周期，以下哪个类图正确表示 ClassA 与 ClassB 之间的关系？

A.

B.

C.

D.

3. (3 分)Security is an important quality of software, which pattern can prevent illegal access to sensitive data?

A. Facade　　B. Singleton

C. Proxy　　D. Adapter

4. (3 分)以下关于面向对象设计原则的描述，正确的是(　　)。

A. 类之间尽量使用抽象方式耦合

B. 尽可能合并类的职责

C. 接口与实现不可分割

D. 优先使用继承而非组合

5. (3 分)Which of the following statements is false for the Facade pattern?

A. Facade provides a simplified interface to a large body of a system.

B. Facade defines a higher-level interface that makes the subsystem easier to use.

C. Facade prevents applications from using subsystem classes if they need to.

D. Facade promotes weak coupling between the subsystem and its clients.

6. (3 分)某图形绘制软件提供了多种不同类型的图形，例如圆形、三角形、长方形等，并为每种图形提供了多种样式，例如平面图形、立体图形等。该软件还需经常增加新的图形及新的图形样式，可采用_____模式设计该图形绘制软件。

【Bridge/桥接】

7. (3 分)We can have multiple catch blocks in a try-catch block code in Java. When any exception occurs in the try block, it will send to the first catch block to process. If the catch block is not able to process it, it forwards the request to the next catch block. This is an example of _____ pattern.

【Chain of Responsibility/职责链】

8. (3 分)Encapsulate a request as an object, thereby letting you parameterize clients with different requests, queue or log requests, and support undoable operations. This is the definition of _____ pattern.

【Command/命令】

9. (3 分)The structure of an algorithm does not change, but small well-defined parts of its operation are handled in subclasses. This is a description of _____ pattern.

【Template Method/模板方法】

10. (3 分)Each unit should have only limited knowledge about other units: only units "closely" related to the current unit. This is the definition of _____. The Facade pattern is a great example of this principle.

【迪米特法则/最少知识原则/Law of Demeter/Least Knowledge Principle】

11. (8 分)在某 FPS(First-Person Shooting Game，第一人称射击) 游戏中提供了多个不同的游戏场景。在每一个游戏场景中，提供了对应的地图(Map)、天气(Weather)和游戏背景音乐(Sound)等。请选择一种合适的设计模式对游戏场景进行设计，使得当用户选择游戏场景时，该场景所对应的地图、天气和背景音乐能够同时出现；此外，还可以方便地在该游戏中增加新的游戏场景。请给出该设计模式的名称和定义。

【参考答案】

模式名称：抽象工厂模式。

模式定义：提供一个创建一系列相关或相互依赖对象的接口，而无须指定它们具体的类。

【评分标准】 模式名称计 4 分；模式定义计 4 分；共 8 分。

12. (8 分)在某 FPS 游戏中，系统可以给所有游戏成员发送通知，例如提示任务执行完毕、发送新的任务提醒、发出敌人袭击警报等。请选择一种合适的设计模式设计该系统通知模块，使得在系统中可以灵活地增加或删除游戏成员。请给出该设计模式的名称和定义。

【参考答案】

模式名称：观察者模式。

模式定义：定义对象之间的一种一对多依赖关系，使得每当一个对象状态发生改变时，其相关依赖对象都得到通知并被自动更新。

【评分标准】 模式名称计 4 分；模式定义计 4 分；共 8 分。

13. (15 分)在某 FPS 游戏中提供了一个游戏管理器(Game Manager)，通过该管理器用户可以对音效(Sound Effect)、场景(Scene)、游戏角色(Role)等对象进行参数设置。为了节约系统资源并且保证对象状态的一致性，在游戏运行时，用户只能打开唯一的一个管理器界面。

根据以上描述，请选择两种合适的设计模式设计该游戏管理器，在实现对多个对象进行统一设置的同时保证游戏管理器的唯一性。

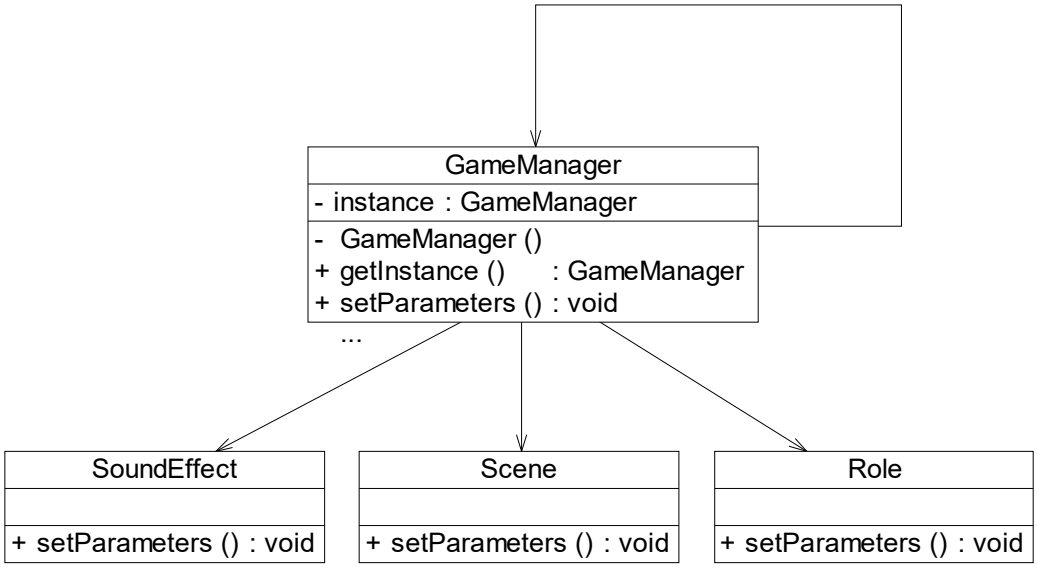
请给出这两种设计模式的名称和定义，并结合实例绘制对应的结构图（类图，类名、方法名和属性名可自行定义）。

【参考答案】

模式名称：外观模式(Facade Pattern)和单例模式(Singleton Pattern)。

模式定义：外观模式：为子系统中的一组接口提供一个统一的入口。外观模式定义了一个高层接口，这个接口使得这一子系统更加容易使用 (Provide a unified interface to a set of interfaces in a subsystem. Facade defines a higher-level interface that makes the subsystem easier to use.)；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来访问这个唯一实例(Ensure a class has only one instance, and provide a global point of access to it.)。

结构图（类图）：



【评分标准】 模式选择正确：4 分（每个 2 分）；模式定义：6 分（每个 3 分）；类图绘制：5 分。

14. (15 分)为了让游戏场景呈现更加逼真的效果，在某 FPS 游戏中可以对场景(Scene)的光照效果等进行渲染(Rendering)。考虑到系统的可扩展性，开发人员可以实现表面渲染(Surface Rendering)和体渲染(Volume Rendering)等算法，也可以调用一些已有的渲染引擎(Render Engine)中的渲染算法。在设计时需要考虑到渲染算法的可复用性，并能够灵活地更换和增加新的渲染效果。

根据以上描述，请选择两种合适的设计模式设计该场景渲染模块，一方面保证可以方便地调用已有的渲染算法，另一方面还可以方便地嵌入新的算法。

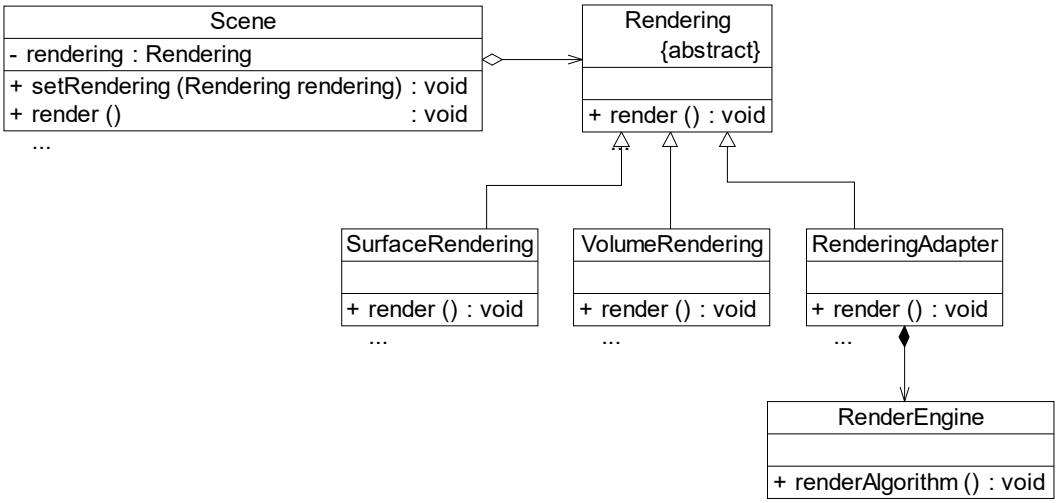
请给出这两种设计模式的名称和定义，并结合实例绘制对应的结构图（类图，类名、方法名和属性名可自行定义）。

【参考答案】

模式名称：策略模式(Strategy Pattern)和适配器模式(Adapter Pattern)。

模式定义：策略模式：定义一系列算法，将每一个算法封装起来，并让它们可以相互替换。策略模式让算法可以独立于使用它的客户变化(Define a family of algorithms, encapsulate each one, and make them interchangeable. Strategy lets the algorithm vary independently from clients that use it.)；适配器模式：将一个类的接口转换成客户希望的另一个接口。适配器模式让那些接口不兼容的类可以一起工作(Convert the interface of a class into another interface clients expect. Adapter lets classes work together that couldn't otherwise because of incompatible interfaces.)。

结构图（类图）：



【评分标准】 模式选择正确：4 分（每个 2 分）；模式定义：6 分（每个 3 分）；类图绘制：5 分。

15. (12 分)In a First-Person Shooting (FPS) Game, a building or a blindage(掩体) is a 3D structure that consists of many 3D Objects such as Cube(立方体), Cylinder(圆柱体), Pyramid(锥体) etc. When we fill a 3D block with color (such as Gray), the same color also gets applied to the Objects in the block. Here a 3D block is made up of different parts and they all have same operations. The parts of a 3D block can be small blocks.

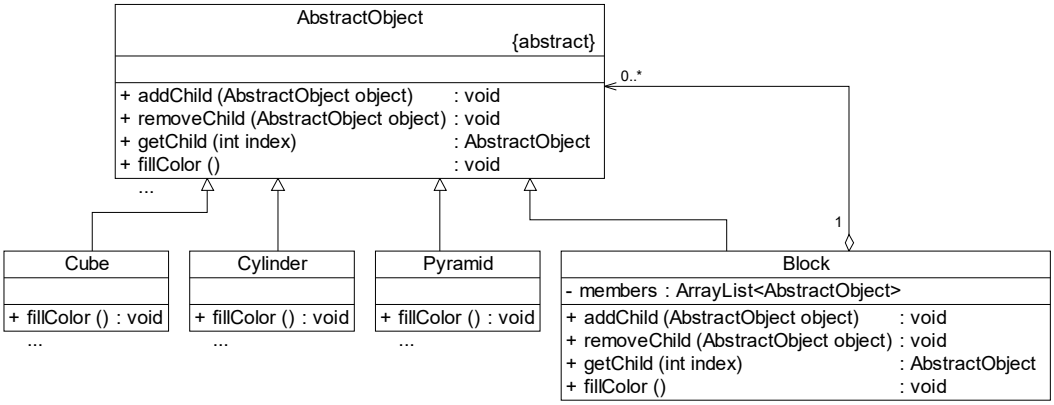
Which design pattern can be used to implement the 3D structure? Give the pattern's definition and draw its structure diagram with this sample.

【参考答案】

模式名称：组合模式 (Composite Pattern)。

定义：组合多个对象形成树形结构以表示"部分-整体"的结构层次。组合模式对单个对象（即叶子对象）和组合对象（即容器对象）的使用具有一致性。(Compose objects into tree structures to represent part-whole hierarchies. Composite lets clients treat individual objects and compositions of objects uniformly.)

类图：



安全组合模式和透明组合模式都可以。

【评分标准】 模式选择正确：4 分；模式定义：4 分；类图绘制：4 分。

16. (12 分)Use a Factory pattern (such as the Simple Factory, the Factory Method or the Abstract Factory) to illustrate the Open-Closed Principle (OCP). You should give a definition of the OCP, and then show an appropriate class diagram and enough code fragments to state your opinion. Whether the pattern supports the OCP or not, you must give reasons.

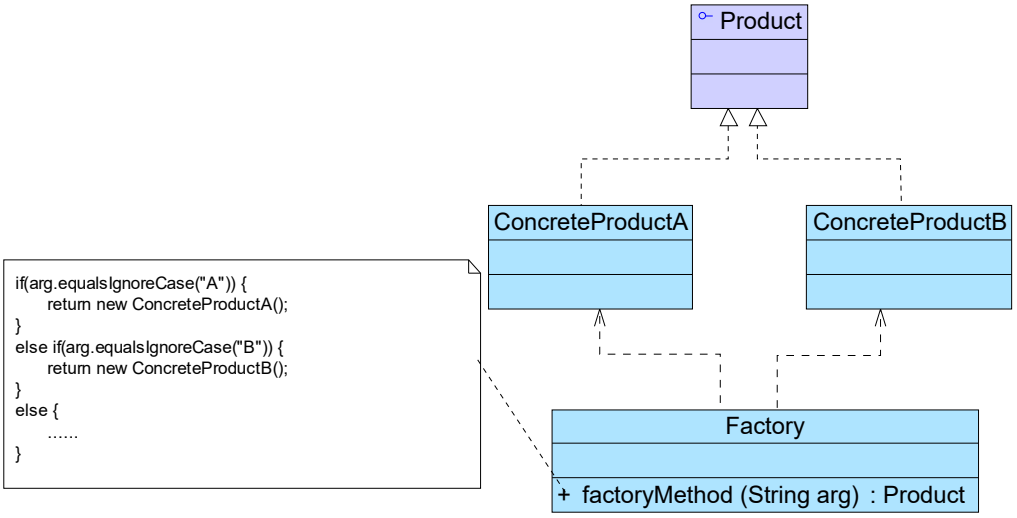
【参考答案】

开闭原则定义：一个软件实体应当对扩展开放，对修改关闭(Software entities should be open for extension, but closed for modification.)。也就是说在设计一个模块的时候，应当使这个模块可以在不被修改的前提下被扩展，即实现在不修改源代码的情况下改变这个模块的行为。

可以选择简单工厂模式、工厂方法模式或抽象工厂模式。

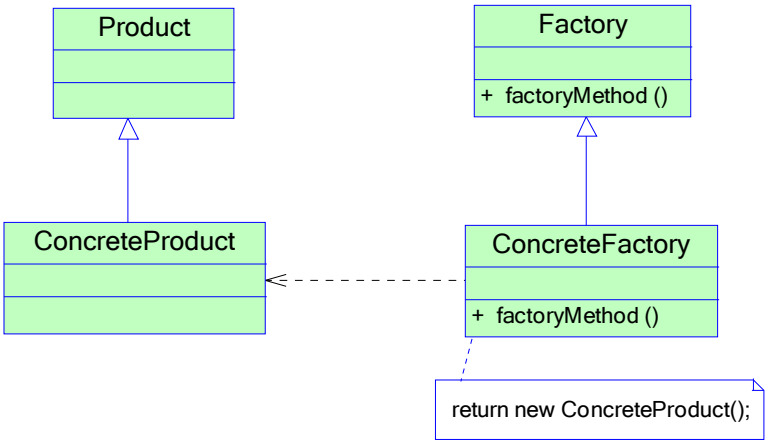
(1) 简单工厂模式：违背了开闭原则，增加新的产品时需要修改工厂类。

简单工厂模式的结构图如下：



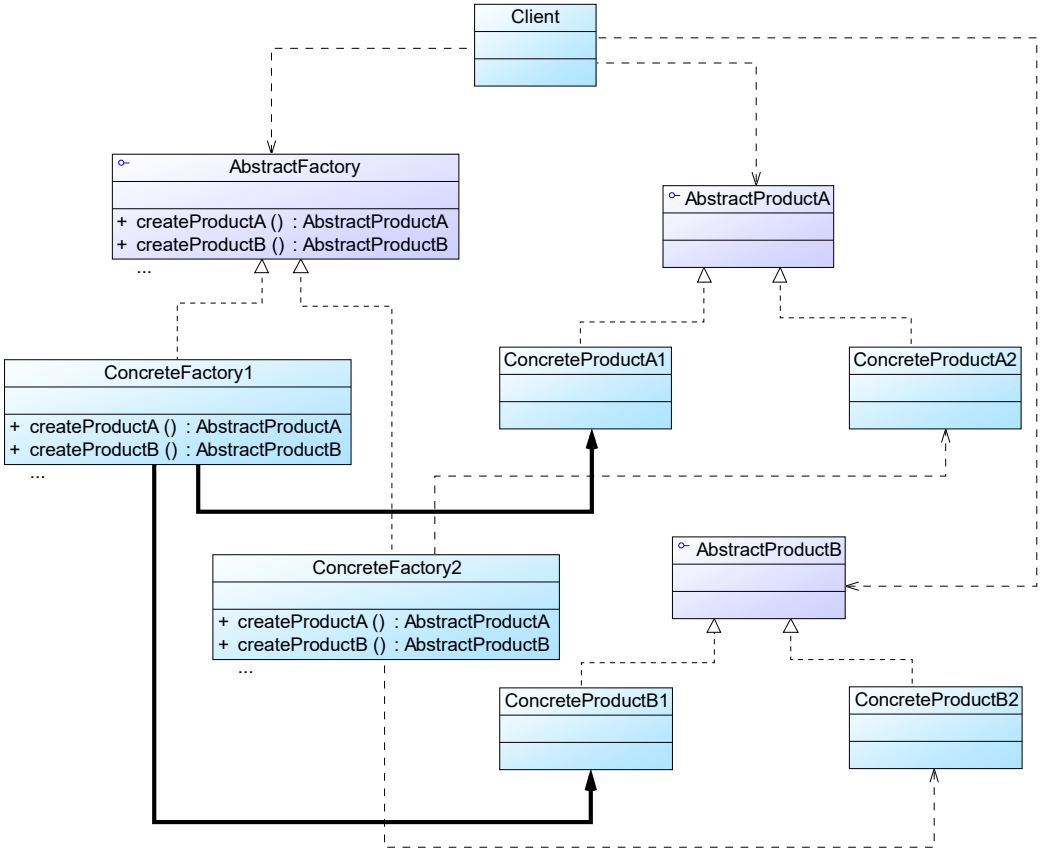
(2) 工厂方法模式：符合开闭原则，增加新的产品只需对应增加一个新的具体工厂类，无需修改源代码。

工厂方法模式的结构图如下：



(3) 抽象工厂模式：具有开闭原则的倾斜性，增加新的产品族符合开闭原则，增加新的产品等级结构违背开闭原则。

抽象工厂模式的结构图如下：



【评分标准】 开闭原则定义：4 分；开闭原则分析：4 分；对应的类图或代码说明：4 分。