**软件工程**

（1）构成计算机软件的是(D)。

A）源代码

B）程序和数据

C）程序和文档

D）程序、数据及相关文档

（2）软件是指D

A）程序

B）程序和文档

C）算法加数据结构

D）程序、数据与相关文档的完整集合

（3）下列描述中正确的是D

A）程序就是软件

B）软件开发不受计算机系统的限制

C）软件既是逻辑实体，又是物理实体

D）软件是程序、数据与相关文档的集合

（4）软件按功能可以分为应用软件、系统软件和支撑软件（或工具软件）。下面属于应用软件的是A

A）学生成绩管理系统

B）C语言编译程序

C）UNIX操作系统

D）数据库管理系统

（5）软件按功能可以分为：应用软件、系统软件和支撑软件（或工具软件）。下面属于应用软件的是C

A）编译程序

B）操作系统

C）教务管理系统

D）汇编程序

（7）下面描述中，不属于软件危机表现的是A

A）软件过程不规范

B）软件开发生产率低

C）软件质量难以控制

D）软件成本不断提高

（8）下列关于软件工程的描述中正确的是C

A）软件工程只是解决软件项目的管理问题

B）软件工程主要解决软件产品的生产率问题

C）软件工程的主要思想是强调在软件开发过程中需要应用工程化原则

D）软件工程只是解决软件开发中的技术问题

（9）下面不属于软件工程的3个要素是D

A）工具

B）过程

C）方法

D）环境

（10）下面不属于软件工程过程的4种基本活动A

A）软件规格说明

B）软件开发

C）软件演进

D）软件测试

（11）软件生命周期是指A

A）软件产品从提出、实现、使用维护到停止使用退役的过程

B）软件从需求分析、设计、实现到测试完成的过程

C）软件的开发过程

D）软件的运行维护过程

（12）软件生命周期中的活动不包括A

A）市场调研

B）需求分析

C）软件测试

D）软件维护

（13）软件生命周期可分为定义阶段、开发阶段和维护阶段，下面不属于开发阶段任务的是C

A）测试

B）设计

C）可行性研究

D）实现

（14）下列选项中不属于软件生命周期开发阶段任务的是C

A）软件测试

B）概要设计

C）软件维护

D）详细设计

（15）软件生命周期可分为定义阶段，开发阶段和维护阶段。详细设计属于B

A）定义阶段

B）开发阶段

C）维护阶段

D）上述三个阶段

（16）下列描述中正确的是A

A）软件交付使用后还需要再进行维护

B）软件工具交付使用就不需要再进行维护

C）软件交付使用后其生命周期就结束

D）软件维护是指修复程序中被破坏的指令

（17）下面不属于软件设计原则的是C

A）抽象

B）模块化

C）自底向上

D）信息隐藏

（18）下面不属于软件需求分析阶段主要工作的是A

A）需求变更申请

B）需求分析

C）需求评审

D）需求获取

（19）下面不属于需求分析阶段任务的是D

A）确定软件系统的功能需求

B）确定软件系统的性能需求

C）需求规格说明书评审

D）制定软件集成测试计划

（20）在软件生命周期中，能准确确定软件系统必须做什么和必须具备哪些功能的阶段是D

A）概要设计

B）详细设计

C）可行性分析

D）需求分析

（21）在软件开发中，需求分析阶段可以使用的工具是B

A）N-S图

B）DFD图

C）PAD图

D）程序流程图

（22）数据流图中带有箭头的线段表示的是D

A）控制流

B）事件驱动

C）模块调用

D）数据流

（23）在软件设计中不使用的工具是C

A）系统结构图

B）PAD图

C）数据流图（DFD图）

D）程序流程图

（24）数据流图用于抽象描述一个软件的逻辑模型，数据流图由一些特定的图符构成。下面图符名标识的图符不属于数据流图合法图符的是D

A）控制流

B）加工

C）数据存储

D）源和潭

（25）在软件设计中，不属于过程设计工具的是D

A）PDL（过程设计语言）

B）PAD图

C）N-S图

D）DFD图

（26）数据流程图（DFD图）是C

A）软件概要设计的工具

B）软件详细设计的工具

C）结构化方法的需求分析工具

D）面向对象方法的需求分析工具

（27）数据字典（DD) 所定义的对象都包含于A

A）数据流图（DFD图）

B）程序流程图

C）软件结构图

D）方框图

（29）在软件开发中，需求分析阶段产生的主要文档是D

A）软件集成测试计划

B）软件详细设计说明书

C）用户手册

D）软件需求规格说明书

（30）软件需求规格说明书的作用不包括D

A）软件验收的依据

B）用户与开发人员对软件要做什么的共同理解

C）软件设计的依据

D）软件可行性研究的依据

（32）从工程管理角度看，软件设计一般分为两步完成，它们是A

A）概要设计与详细设计

B）数据设计与接口设计

C）软件结构设计与数据设计

D）过程设计与数据设计

（33）下面不属于软件设计阶段任务的是C

A）软件总体设计

B）算法设计

C）制定软件确认测试计划

D）数据库设计

（34）软件设计中模块划分应遵循的准则是B

A）低内聚低耦合

B）高内聚低耦合

C）低内聚高耦合

D）高内聚高耦合

（35）耦合性和内聚性是对模块独立性度量的两个标准。下列叙述中正确的是B

A）提高耦合性降低内聚性有利于提高模块的独立性

B）降低耦合性提高内聚性有利于提高模块的独立性

C）耦合性是指一个模块内部各个元素间彼此结合的紧密程度

D）内聚性是指模块间互相连接的紧密程度

（37）两个或两个以上模块之间关联的紧密程度称为A

A）耦合度

B）内聚度

C）复杂度

D）数据传输特性

（38）在结构化程序设计中，模块划分的原则是D

A）各模块应包括尽量多的功能

B）各模块的规模应尽量大

C）各模块之间的联系应尽量紧密

D）模块内具有高内聚度、模块间具有低耦合度

（39）下面不能作为结构化方法软件需求分析工具的是A

A）系统结构图

B）数据字典(DD)

C）数据流程图(DFD图)

D）判定表

（40）下面描述中错误的是A

A）系统总体结构图支持软件系统的详细设计

B）软件设计是将软件需求转换为软件表示的过程

C）数据结构与数据库设计是软件设计的任务之一

D）PAD图是软件详细设计的表示工具

（41）程序流程图中带有箭头的线段表示的是C

A）图元关系

B）数据流

C）控制流

D）调用关系

（42）下面叙述中错误的是A

A）软件测试的目的是发现错误并改正错误

B）对被调试的程序进行"错误定位"是程序调试的必要步骤

C）程序调试通常也称为Debug

D）软件测试应严格执行测试计划，排除测试的随意性

（43）软件测试的目的是D

A）评估软件可靠性

B）发现并改正程序中的错误

C）改正程序中的错误

D）发现程序中的错误

（44）检查软件产品是否符合需求定义的过程称为A

A）确认测试

B）集成测试

C）验证测试

D）验收测试

（45）下列描述中正确的是A

A）软件测试的主要目的是发现程序中的错误

B）软件测试的主要目的是确定程序中错误的位置

C）为了提高软件测试的效率，最好由程序编制者自己来完成软件测试的工作

D）软件测试是证明软件没有错误

（46）下列对于软件测试的描述中正确的是C

A）软件测试的目的是证明程序是否正确

B）软件测试的目的是使程序运行结果正确

C）软件测试的目的是尽可能多地发现程序中的错误

D）软件测试的目的是使程序符合结构化原则

（47）下列描述中正确的是D

A）软件测试应该由程序开发者来完成

B）程序经调试后一般不需要再测试

C）软件维护只包括对程序代码的维护

D）以上三种说法都不对

（48）在黑盒测试方法中，设计测试用例的主要根据是B

A）程序内部逻辑

B）程序外部功能

C）程序数据结构

D）程序流程图

（49）下面属于黑盒测试方法的是C

A）语句覆盖

B）逻辑覆盖

C）边界值分析

D）路径覆盖

（50）下面属于白盒测试方法的是B

A）等价类划分法

B）逻辑覆盖

C）边界值分析法

D）错误推测法

（51）下面不属于软件测试实施步骤的是B

A）集成测试

B）回归测试

C）确认测试

D）单元测试

（52）软件（程序）调试的任务是A

A）诊断和改正程序中的错误

B）尽可能多地发现程序中的错误

C）发现并改正程序中的所有错误

D）确定程序中错误的性质

（53）程序调试的任务是D

A）设计测试用例

B）验证程序的正确性

C）发现程序中的错误

D）诊断和改正程序中的错误

（55）以下关于结构化程序设计的叙述正确的是（B）。

A）结构化程序使用goto语句会很便捷

B）在C语言中，程序的模块化是利用函数实现的

C）一个结构化程序必须由顺序、分支、循环三种结构组成

D）由三种基本结构构成的程序只能解决小规模的问题

（56）以下选项中关于程序模块化的叙述错误的是（A）。

A、可采用自底向上、逐步细化的设计方法把若干独立模块组装成所要求的程序

B、把程序分成若干相对独立、功能单一的模块，可便于重复使用这些模块

C、把程序分成若干相对独立的模块，可便于编程和调试

D、可采用自顶向下、逐步细化的设计方法把若干独立模块组装成所要求的程序

(57) 设计模式具有的优点（A）。

A．适应需求变化 B.程序易于理解 C．减少开发过程中的代码开发工作量  D.简化软件系统的设计

(58)设计模式具有的优点（A）。

A．适应需求变化 B.程序易于理解 C．减少开发过程中的代码开发工作量  D.简化软件系统的设计

（59）设计模式的两大主题是(  D)。

A.系统的维护与开发 B.对象组合与类的继承

C.系统架构与系统开发 D.系统复用与系统扩展

(60) 以下哪些问题通过应用设计模式不能够解决 。（C）

A）指定对象的接口 B）针对接口编程

C）确定软件的功能都正确实现 D）设计应支持变化

(61)

（62）(D）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。

A. Adapter（适配器）模式 B. Bridge（桥接）模式

C. Strategy（策略）模式 D.Facade （外观）模式

(63) Adapter（适配器）模式的意图是（B）。

A. 希望简化现有系统的使用方法。你需要定义自己的借口。

B．将一个无法控制的现有对象与一个特定借口相匹配。

C. 将一组实现部分从另一组使用它们的对象中分离出来。

D．你需要为特定的客户（或情况）提供特定系列的对象。

（64）在观察者模式中，表述错误的是（ c ）

A.观察者角色的更新是被动的。

B.被观察者可以通知观察者进行更新

C.观察者可以改变被观察者的状态，再由被观察者通知所有观察者依据被观察者的状态进行。

D.以上表述全部错误。

（65）关于继承表述错误的是：（ d ）

A.继承是一种通过扩展一个已有对象的实现，从而获得新功能的复用方法。

B.泛化类（超类）可以显式地捕获那些公共的属性和方法。特殊类（子类）则通过附加属性和方法来进行实现的扩展。

C.破坏了封装性，因为这会将父类的实现细节暴露给子类。

D.继承本质上是“白盒复用”，对父类的修改，不会影响到子类。

1.设计模式的原理? (C)

A.面对实现编程

B. 面向对象编程

C. 面向接口编程

D. 面向组合编程

2. 以下对"开-闭"原则的一些描述错误的是?(A)

A. "开-闭"原则与"对可变性的封装原则"没有相似性.

B. 找到一个系统的可变元素,将它封装起来,叫"开-闭"原则

C. 对修改关闭: 是其原则之一

D. 从抽象层导出一个或多个新的具体类可以改变系统的行为,是其原则之一

3.以下不属于创建型模式是? (A)

A. PROXY（代理）                               B.BUILDER（生成器）

C.PROTOTYPE（原型）              D.SINGLETON（单件）

4.以下不属于结构型模式是? (D)

A. COMPOSITE（组合）            B. ADAPTER（适配器）

C. FLYWEIGHT（享元）              D. SINGLETON(单例)

5. 以下不属于行为型模式是? (D )

A. COMMAND（命令）

B. STRATEGY(策略)

C. MEMENTO（备忘录）

D. BRIDGE（桥接）

6.以下意图那个是用来描述ABSTRACT FACTORY（抽象工厂)?(A)

A.提供一个创建一系列相关或相互依赖对象的接口，而无需指定它们具体的类。

B.定义一个用于创建对象的接口，让子类决定实例化哪一个类。

C.将一个类的接口转换成客户希望的另外一个接口。

D.表示一个作用于某对象结构中的各元素的操作。

7.以下意图那个是用来描述BUILDER（生成器）?(B)

A.定义一个用于创建对象的接口，让子类决定实例化哪一个类

B.将一个复杂对象的构建与它的表示分离，使得同样的构建过程可以创建不同的表示。

C.保证一个类仅有一个实例，并提供一个访问它的全局访问点。

D.运用共享技术有效地支持大量细粒度的对象。

8.以下意图那个是用来描述FACTORY METHOD（工厂方法）?(C)

A.提供一个创建一系列相关或相互依赖对象的接口，而无需指定它们具体的类。

B.表示一个作用于某对象结构中的各元素的操作。它使你可以在不改变各元素的类的前提下定义作用于这些元素的新操作。

C.定义一个用于创建对象的接口，让子类决定实例化哪一个类。该模式使一个类的实例化延迟到其子类。

D.定义一系列的算法,把它们一个个封装起来, 并且使它们可相互替换。本模式使得算法可独立于使用它的客户而变化。

9.以下意图那个是用来描述PROTOTYPE（原型）?(D)

A.允许一个对象在其内部状态改变时改变它的行为。对象看起来似乎修改了它的类。

B.表示一个作用于某对象结构中的各元素的操作。它使你可以在不改变各元素的类的前提下定义作用于这些元素的新操作。

C.定义对象间的一种一对多的依赖关系,当一个对象的状态发生改变时, 所有依赖于它的对象都得到通知并被自动更新。

D.用原型实例指定创建对象的种类，并且通过拷贝这些原型创建新的对象。

10.以下意图那个是用来描述SINGLETON（单件）?(B)

A.将一个类的接口转换成客户希望的另外一个接口。该模式使得原本由于接口不兼容而不能一起工作的那些类可以一起工作

B.保证一个类仅有一个实例，并提供一个访问它的全局访问点。

C.定义一系列的算法,把它们一个个封装起来, 并且使它们可相互替换。本模式使得算法可独立于使用它的客户而变化。

D.用一个中介对象来封装一系列的对象交互。

11.以下意图那个是用来描述ADAPTER（适配器）?(A)

A.将一个类的接口转换成客户希望的另外一个接口。本模式使得原本由于接口不兼容而不能一起工作的那些类可以一起工作。

B.定义一个用于创建对象的接口，让子类决定实例化哪一个类

C.表示一个作用于某对象结构中的各元素的操作。它使你可以在不改变各元素的类的前提下定义作用于这些元素的新操作。

D.将一个请求封装为一个对象，从而使你可用不同的请求对客户进行参数化；对请求排队或记录请求日志，以及支持可撤消的操作。

12.以下意图那个是用来描述BRIDGE（桥接）?(B)

A.提供一个创建一系列相关或相互依赖对象的接口，而无需指定它们具体的类。

B.将抽象部分与它的实现部分分离，使它们都可以独立地变化。

C.将一个复杂对象的构建与它的表示分离，使得同样的构建过程可以创建不同的表示。

D.动态地给一个对象添加一些额外的职责。

13.以下意图那个是用来描述COMPOSITE（组合）?(C)

A.为其他对象提供一种代理以控制对这个对象的访问。

B.运用共享技术有效地支持大量细粒度的对象。

C.将对象组合成树形结构以表示“部分-整体”的层次结构。

C.将一个复杂对象的构建与它的表示分离，使得同样的构建过程可以创建不同的表示。

14.以下意图那个是用来描述DECORATOR（装饰）?(D)

A.运用共享技术有效地支持大量细粒度的对象。

B.用原型实例指定创建对象的种类，并且通过拷贝这些原型创建新的对象。

C.将抽象部分与它的实现部分分离，使它们都可以独立地变化。

D.动态地给一个对象添加一些额外的职责。

15.以下意图那个是用来描述 FACADE（外观）?(A)

A.为子系统中的一组接口提供一个一致的界面，本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。

B.定义一个用于创建对象的接口，让子类决定实例化哪一个类。

C.保证一个类仅有一个实例，并提供一个访问它的全局访问点。

D.在不破坏封装性的前提下，捕获一个对象的内部状态，并在该对象之外保存这个状态。这样以后就可将该对象恢复到原先保存的状态。

16.以下意图那个是用来描述FLYWEIGHT（享元）?(B)

A.将抽象部分与它的实现部分分离，使它们都可以独立地变化。

B.运用共享技术有效地支持大量细粒度的对象。

C.动态地给一个对象添加一些额外的职责。

D.用原型实例指定创建对象的种类，并且通过拷贝这些原型创建新的对象。

17.以下意图那个是用来描述 PROXY（代理）?(C)

A.用原型实例指定创建对象的种类，并且通过拷贝这些原型创建新的对象。

B.运用共享技术有效地支持大量细粒度的对象。

C.为其他对象提供一种代理以控制对这个对象的访问。

D.将一个复杂对象的构建与它的表示分离，使得同样的构建过程可以创建不同的表示。

18.以下意图那个是用来描述CHAIN OF RESPONSIBILITY(职责链)?(D)

A.为子系统中的一组接口提供一个一致的界面，本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。

B.提供一种方法顺序访问一个聚合对象中各个元素, 而又不需暴露该对象的内部表示。

C.在不破坏封装性的前提下，捕获一个对象的内部状态，并在该对象之外保存这个状态。这样以后就可将该对象恢复到原先保存的状态。

D.使多个对象都有机会处理请求，从而避免请求的发送者和接收者之间的耦合关系。

19.以下意图那个是用来描述 COMMAND（命令）?(A)

A.将一个请求封装为一个对象，从而使你可用不同的请求对客户进行参数化；对请求排队或记录请求日志，以及支持可撤消的操作

B.定义一系列的算法,把它们一个个封装起来, 并且使它们可相互替换。本模式使得算法可独立于使用它的客户而变化。

C.为其他对象提供一种代理以控制对这个对象的访问。

D.保证一个类仅有一个实例，并提供一个访问它的全局访问点。

20.以下意图那个是用来描述 INTERPRETER(解释器)?(B)

A.将抽象部分与它的实现部分分离，使它们都可以独立地变化。

B.给定一个语言，定义它的文法的一种表示，并定义一个解释器，这个解释器使用该表示来解释语言中的句子。

C.将一个复杂对象的构建与它的表示分离，使得同样的构建过程可以创建不同的表示。

D.为其他对象提供一种代理以控制对这个对象的访问。

21.以下意图那个是用来描述 ITERATOR(迭代器)?(C)

A.使多个对象都有机会处理请求，从而避免请求的发送者和接收者之间的耦合关系。

B.用原型实例指定创建对象的种类，并且通过拷贝这些原型创建新的对象。

C.提供一种方法顺序访问一个聚合对象中各个元素, 而又不需暴露该对象的内部表示。

D.运用共享技术有效地支持大量细粒度的对象。

22.以下意图那个是用来描述 MEDIATOR(中介者)?(D)

A.提供一种方法顺序访问一个聚合对象中各个元素, 而又不需暴露该对象的内部表示。

B.将抽象部分与它的实现部分分离，使它们都可以独立地变化。

C.定义一个用于创建对象的接口，让子类决定实例化哪一个类。

D.用一个中介对象来封装一系列的对象交互。

23.以下意图那个是用来描述 MEMENTO（备忘录）?(A)

A.在不破坏封装性的前提下，捕获一个对象的内部状态，并在该对象之外保存这个状态。这样以后就可将该对象恢复到原先保存的状态。

B.保证一个类仅有一个实例，并提供一个访问它的全局访问点。

C.将一个请求封装为一个对象，从而使你可用不同的请求对客户进行参数化；对请求排队或记录请求日志，以及支持可撤消的操作

D.提供一种方法顺序访问一个聚合对象中各个元素, 而又不需暴露该对象的内部表示。

24.以下意图那个是用来描述 OBSERVER（观察者）?(B)

A.将抽象部分与它的实现部分分离，使它们都可以独立地变化。

B.定义对象间的一种一对多的依赖关系,当一个对象的状态发生改变时, 所有依赖于它的对象都得到通知并被自动更新。

C.用原型实例指定创建对象的种类，并且通过拷贝这些原型创建新的对象。

D.使多个对象都有机会处理请求，从而避免请求的发送者和接收者之间的耦合关系。

25.以下意图那个是用来描述 STATE（状态）?(C)

A.使多个对象都有机会处理请求，从而避免请求的发送者和接收者之间的耦合关系。

B.提供一种方法顺序访问一个聚合对象中各个元素, 而又不需暴露该对象的内部表示。

C.允许一个对象在其内部状态改变时改变它的行为。对象看起来似乎修改了它的类。

D.在不破坏封装性的前提下，捕获一个对象的内部状态，并在该对象之外保存这个状态。这样以后就可将该对象恢复到原先保存的状态。

26.以下意图那个是用来描述 STRATEGY(策略)?(D)

A.将抽象部分与它的实现部分分离，使它们都可以独立地变化。

B.将一个复杂对象的构建与它的表示分离，使得同样的构建过程可以创建不同的表示。

C.将抽象部分与它的实现部分分离，使它们都可以独立地变化。

D.定义一系列的算法,把它们一个个封装起来, 并且使它们可相互替换

27.以下意图那个是用来描述 TEMPLATE METHOD(模板方法)?(A)

A.定义一个操作中的算法的骨架，而将一些步骤延迟到子类中。

B.为其他对象提供一种代理以控制对这个对象的访问。

C.将抽象部分与它的实现部分分离，使它们都可以独立地变化。

D.使多个对象都有机会处理请求，从而避免请求的发送者和接收者之间的耦合关系。

28.以下意图那个是用来描述 VISITOR（访问者）?(B)

A.定义对象间的一种一对多的依赖关系,当一个对象的状态发生改变时, 所有依赖于它的对象都得到通知并被自动更新。

B.表示一个作用于某对象结构中的各元素的操作。

C.在不破坏封装性的前提下，捕获一个对象的内部状态，并在该对象之外保存这个状态。这样以后就可将该对象恢复到原先保存的状态。

D.用原型实例指定创建对象的种类，并且通过拷贝这些原型创建新的对象。

34.关于模式适用性,在以下情况不适合使用Abstract Factory模式?(D)

A.一个系统要独立于它的产品的创建、组合和表示时。

B.一个系统要由多个产品系列中的一个来配置时。

C.当你要强调一系列相关的产品对象的设计以便进行联合使用时。

D.当一个类希望由它的子类来指定它所创建的对象的时候。

35.关于模式适用性,在以下情况不适合使用FACTORY METHOD模式?(C)

A. 当一个类不知道它所必须创建的对象的类的时候。

B.当一个类希望由它的子类来指定它所创建的对象的时候。

C.当你提供一个产品类库，而只想显示它们的接口而不是实现时。

D.当类将创建对象的职责委托给多个帮助子类中的某一个，并且你希望将哪一个帮助子类是代理者这一信息局部化的时候。

36.关于模式适用性,在以下情况可以使用builder模式?(B)

A.当类只能有一个实例而且客户可以从一个众所周知的访问点访问它时。

B.当创建复杂对象的算法应该独立于该对象的组成部分以及它们的装配方式时。

C.当构造过程必须允许被构造的对象有不同的表示时。

D.一个对象的行为取决于它的状态, 并且它必须在运行时刻根据状态改变它的行为。

37.关于模式适用性,在以下情况不适合使用Factory Method模式?(A)

A.一次性实现一个算法的不变的部分，并将可变的行为留给子类来实现。

B.当一个类希望由它的子类来指定它所创建的对象的时候。

C.当类将创建对象的职责委托给多个帮助子类中的某一个，并且你希望将哪一个帮助子类是代理者这一信息局部化的时候。

D.当一个类不知道它所必须创建的对象的类的时候。

38.关于模式适用性,在以下情况不适合使用PROTOTYPE模式?(B)

A.当要实例化的类是在运行时刻指定时，例如，通过动态装载

B.当你要强调一系列相关的产品对象的设计以便进行联合使用时

C.为了避免创建一个与产品类层次平行的工厂类层次时

D.当一个类的实例只能有几个不同状态组合中的一种时

39.关于模式适用性,在以下情况适合使用Singleton模式?(C)

A.当一个类不知道它所必须创建的对象的类的时候。

B.当一个类的实例只能有几个不同状态组合中的一种时

C.当这个唯一实例应该是通过子类化可扩展的，并且客户应该无需更改代码就能使用一个扩展的实例时。

D.当一个类希望由它的子类来指定它所创建的对象的时候。

40.关于模式适用性,在以下情况不适合使用Adapter（适配器）模式?(D)

A.你想使用一个已经存在的类，而它的接口不符合你的需求。

B.你想创建一个可以复用的类，该类可以与其他不相关的类或不可预见的类（即那些接口可能不一定兼容的类）协同工作。

B.你想使用一些已经存在的子类，但是不可能对每一个都进行子类化以匹配它们的接口。对象适配器可以适配它的父类接口。

D.如果删除对象的外部状态，那么可以用相对较少的共享对象取代很多组对象。

41.关于模式适用性,在以下情况不适合使用Bridge（桥接）模式?(A)

A.你不希望在抽象和它的实现部分之间有一个固定的绑定关系。例如这种情况可能是因为，在程序运行时刻实现部分应可以被选择或者切换。

B.类的抽象以及它的实现都应该可以通过生成子类的方法加以扩充。这时B r i d g e模式使你可以对不同的抽象接口和实现部分进行组合，并分别对它们进行扩充。

C.对一个抽象的实现部分的修改应对客户不产生影响，即客户的代码不必重新编译。

D.客户程序与抽象类的实现部分之间存在着很大的依赖性。

42.关于模式适用性,在以下情况不适合使用COMPOSITE（组合）模式?(B)

A.你想使用一个已经存在的类，而它的接口不符合你的需求。

B.你想表示对象的部分-整体层次结构。

C.当一个类的实例只能有几个不同状态组合中的一种时

D.一个对象的行为取决于它的状态, 并且它必须在运行时刻根据状态改变它的行为。

43.关于模式适用性,在以下情况不适合使用DECORATOR（装饰）模式?(C)

A.在不影响其他对象的情况下，以动态、透明的方式给单个对象添加职责。

B. 处理那些可以撤消的职责。

C.客户程序与抽象类的实现部分之间存在着很大的依赖性。

D. 当不能采用生成子类的方法进行扩充时。一种情况是，可能有大量独立的扩展，为支持每一种组合将产生大量的子类，使得子类数目呈爆炸性增长。另一种情况可能是因为类定义被隐藏，或类定义不能用于生成子类。

44.关于模式适用性,在以下情况不适合使用FLYWEIGHT（享元）模式?(D)

A.一个应用程序使用了大量的对象。

B.完全由于使用大量的对象，造成很大的存储开销。

C.对象的大多数状态都可变为外部状态。

D.你想使用一个已经存在的类，而它的接口不符合你的需求

45.关于模式适用性,在以下情况不适合使用PROXY（代理）模式?(A)

A.你想使用一个已经存在的类，而它的接口不符合你的需求。

B.根据需要创建开销很大的对象

C.在需要用比较通用和复杂的对象指针代替简单的指针的时候

D.取代了简单的指针，它在访问对象时执行一些附加操作。

46.关于模式适用性,在以下情况不适合使用CHAIN OF RESPONSIBILITY(职责链)模式?(B)

A.有多个的对象可以处理一个请求，哪个对象处理该请求运行时刻自动确定。

B.在需要用比较通用和复杂的对象指针代替简单的指针的时候

C. 你想在不明确指定接收者的情况下，向多个对象中的一个提交一个请求。

D.可处理一个请求的对象集合应被动态指定。

47. 关于模式适用性,在以下情况不适合使用COMMAND（命令）模式?(C)

A.抽象出待执行的动作以参数化某对象,使用过程语言中的回调（c a l l b a c k）函数表达这种参数化机制。

B.Java语言中的:AWT的事件处理

C.在需要用比较通用和复杂的对象指针代替简单的指针的时候

D.一个系统需要支持交易(transaction).一个交易结构封装了一组数据更新命令.

48. 关于模式适用性,在以下情况不适合使用OBSERVER（观察者）模式?(D)

A.当一个抽象模型有两个方面, 其中一个方面依赖于另一方面。将这二者封装在独立的对象中以使它们可以各自独立地改变和复用。

B.当对一个对象的改变需要同时改变其它对象, 而不知道具体有多少对象有待改变。

C. 当一个对象必须通知其它对象，而它又不能假定其它对象是谁。换言之, 你不希望这些对象是紧密耦合的。

D.在不影响其他对象的情况下，以动态、透明的方式给单个对象添加职责。

49. 关于模式适用性,在以下情况不适合使用 STRATEGY(策略)模式?(A)

A. 当一个对象必须通知其它对象，而它又不能假定其它对象是谁。换言之, 你不希望这些对象是紧密耦合的。

B.许多相关的类仅仅是行为有异。“策略”提供了一种用多个行为中的一个行为来配置一个类的方法。

C.需要使用一个算法的不同变体。例如，你可能会定义一些反映不同的空间/时间权衡的算法。当这些变体实现为一个算法的类层次时,可以使用策略模式。

D.算法使用客户不应该知道的数据。可使用策略模式以避免暴露复杂的、与算法相关的数据结构。

50. 关于模式适用性,在以下情况不适合使用 TEMPLATE METHOD(模板方法)模式?(B)

A.一次性实现一个算法的不变的部分，并将可变的行为留给子类来实现。

B.当对一个对象的改变需要同时改变其它对象, 而不知道具体有多少对象有待改变。

C. 各子类中公共的行为应被提取出来并集中到一个公共父类中以避免代码重复。首先识别现有代码中的不同之处，并且将不同之处分离为新的操作。最后，用一个调用这些新的操作的模板方法来替换这些不同的代码。

D.控制子类扩展。模板方法只在特定点调用“hook”操作，这样就只允许在这些点进行扩展。

简答题：

1、设计模式中一般都遵循的原则有什么？

面向抽象原则：设计一个类时，不让该类面向具体的类，而是面向抽象类或接口

开-闭原则：设计应当对扩展开放，对修改关闭

多用组合少用继承原则：设计中避开类继承的缺点，充分使用对象组合的优点

高内聚-低耦合原则

2、简述设计模式4要素

模式名称(pattern name) 一个助记名，它用一两个词来描述模式的问题、解决方案和效果。

问题(problem) 描述了应该在合适使用模式。它解决了设计问题和问题存在的前后因果，它可能描述了特定的设计问题，也可能描述了导致不灵活设计的类或对象结构。

解决方案(solution) 描述了设计的组成成分，它们之间的相互关系及各自的职责和协作方式。

效果(consequences) 描述了模式应用的效果及使用模式应权衡的问题。

3、软件工程基本原理

（1）用分阶段的生命周期计划严格管理。

（2）坚持进行阶段评审。

（3）实行严格的产品控制。

（4）采用现代程序设计技术。（提高软件开发的效率和维护的效率）

（5）结果应能清楚地审查。

（6）开发小组的人员应该少而精。

（7）承认不断改进软件工程实践的必要性。

4、依次写出软件生命周期各个部分

（1）问题定义

（2）可行性研究

（3）需求分析

（4）总体设计和详细设计

（5）编码

（6）测试

（7）软件维护

5、列出三种以上的常见的软件的开发模型

瀑布模型；原型模型；增量模型；螺旋模型；Rational 模型；总体数据库规划模型；

设计题：按照以下E-R图写出数据库表格。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 对应中文 | 关键字 |



