# 《软件设计与体系结构》复习指导手册

## 课程介绍

《软件设计与体系结构》是“软件工程”专业（本科）的一门统设必修课，并作为其他专业的选修课。本课程的主要任务是讲授软件设计和软件结构方面的基本思想、知识、原理和方法，包括软件设计与软件体系结构在软件工程中的地位与作用、软件设计基本方法与原则、统一建模语言UML2.0、面向对象的软件设计方法、软件体系结构风格与设计模式、基于构件的软件体系结构、软件体系结构评估、软件设计的进化等内容。

本课程的教学目的是使学生在软件工程思想的基础上，掌握软件体系结构相关知识、原理和各种类型设计模式的基本结构，对现有软件体系结构有比较深入的理解，能够从系统结构角度分析现有软件系统，能够运用具体的软件设计方法解决实际的软件工程问题，并能利用所学到的软件系统结构知识高效地设计软件系统，为从事大型复杂软件开发打下良好的基础。

## 考核说明

1. 考核对象

国家开放大学软件工程专业（本科）学生。

2. 启用时间

2020年秋季学期。

3. 考核目标

通过考核使学生在软件工程思想的基础上，掌握软件体系结构相关知识、原理和各种类型设计模式的基本结构，对现有软件体系结构有比较深入的理解，能够从系统结构角度分析现有软件系统，能够运用具体的软件设计方法解决实际的软件工程问题，并能利用所学到的软件系统结构知识高效地设计软件系统，为从事大型复杂软件开发打下良好的基础。

4. 考核依据

本课程考核说明是依据国家开放大学“软件设计与体系结构”课程教学大纲、文字教材《软件设计与体系结构》（高等教育出版社2017年12月第2版）制定的。本课程考核说明是课程考核命题的基本依据。

5. 考核方式及计分方法

本课程考核采用形成性考核与终结性考核相结合的方式。形成性考核占课程综合成绩的30%，终结性考核占课程综合成绩的70%。课程考核成绩统一采用百分制，即形成性考核、终结性考核、课程综合成绩均采用百分制。

课程综合成绩达到60分及以上（及格），可获得本课程相应学分。

考核方式相关信息以国家开放大学当学期发布的考试安排文件为准。

6. 终结性考核

（1）考核目的

终结性考核是在形成性考核的基础上，对学生学习情况和学习效果进行的一次全面检测。

（2）命题原则

第一，本课程的考试命题严格控制在教学大纲规定的教学内容和考核要求的范围之内。

第二，按掌握、理解、了解三个层次命题。

第三，每份试卷所考的内容，覆盖本课程教材所学内容的70%以上章节。

第四，试题难度适中。一般来讲，可分为容易、适中、较难三个程度，所占比例大致为：容易占40%，适中占40%，较难占20%。

（3）考核手段

网考

（4）考核方式

闭卷

（5）考核时限

90分钟

（6）每套试卷的试题类型、数量及分值

1. 单项选择题：12道题。
2. 多项选择题：10道题。
3. 简答题：3道题。
4. 应用题：0 （考题不包含应用题）

试卷试题数合计25道题；试卷总分数100分。

## 模拟题

### 单选题

1. 关于瀑布模型的描述，下面不正确的是（ ）。

【A.】瀑布模型的核心思想是按工序将问题化简，将功能的实现与设计分开，便于分工协作

【B.】瀑布模型采用结构化的分析与设计方法，以便将逻辑实现与物理实现分开

【C.】瀑布模型需建立一个可以运行的软件原型，该原型向用户展示待开发软件的全部或部分功能和性能

【D.】瀑布模型将软件生命周期划分为制定计划、需求分析、软件设计、程序编写、软件测试和运行维护等六个基本活动，并规定了自上而下、相互衔接的次序

【答案】C

1. 关于快速原型模型，下面不正确的是（ ）。

【A.】快速原型首先需建立一个可以运行的软件原型，该原型向用户展示待开发软件的全部或部分功能和性能

【B.】快速原型模型能够实现客户或未来的用户与系统的交互，用户对该原型进行测试评定，给出具体改进意见以丰富细化软件需求

【C.】开发人员根据用户意见对软件进行修改完善，直至用户满意认可之后，进行软件的完整实现及测试、维护

【D.】快速原型模型因为要求快速建立，增加了由于软件需求不明确带来的开发风险

【答案】D

1. 以下哪项是螺旋模型的优点 ( )。

【A.】采用螺旋模型需要具有相当丰富的风险评估经验和专门知识，在风险较大的项目开发中，如果未能够及时标识风险，势必造成重大损失

【B.】对可选方案和约束条件的强调既有利于已有软件的重用，也有助于把软件质量作为软件开发的目标

【C.】过多的迭代次数会增加开发成本，延迟提交时间

【D.】增加了过多测试或测试不足所带来的风险

【答案】B

1. 关于增量模型，下面不正确的是（ ）。

【A.】增量模型的各个构件是一次性并入已有的软件体系结构，因此不需要软件具备开放式的体系结构

【B.】作为瀑布模型和快速原型模型的综合，能在较短的时间内向用户提交可完成部分工作的产品

【C.】将待开发的软件系统模块化，可以分批次地提交软件产品，使用户可以及时了解软件项目的进展

【D.】以组件为单位进行开发降低了软件开发的风险。一个开发周期内的错误不会影响到整个软件系统的开发

【答案】A

1. 以下哪项不属于软件工程的三要素（ ）。

【A.】方法

【B.】工具

【C.】过程

【D.】质量

【答案】D

1. 关于软件工程中结构化方法的设计原则，以下描述错误的是（ ）。

【A.】每个模块执行一个功能，每个模块只有一个入口和一个出口

【B.】每个模块由过程语句或函数方式等归其上级模块调用

【C.】应当构造内部联系紧密的模块，降低模块间的联系

【D.】结构化设计采用自下而上的模块化设计方法

【答案】D

1. 以下哪项不是面向对象方法的主要特征（ ）。

【A.】封装性

【B.】继承性

【C.】多态性

【D.】适应性

【答案】D

1. 耦合是模块之间依赖程度的度量。以下关于耦合的说法，错误的是（ ）。

【A.】如果两模块中任一个都不依赖于对方能独立工作，则称这两模块为非直接耦合

【B.】如果两模块都与同一个数据结构有关，则称这两个模块为特征耦合

【C.】当若干模块均与同一个外部环境关联，它们之间存在内容耦合

【D.】当若干模块通过全局的数据环境相互作用时，它们之间存在公共耦合

【答案】C

1. 关于软件体系结构核心模型的组成元素，以下说法错误的是（ ）。

【A.】连接件用来表示构件之间的交互

【B.】配置用来表示构件和连接件的拓扑逻辑和约束

【C.】构件作为封装实体，不需要借助其接口即可与外部环境交互

【D.】构件用来表示系统中主要的计算元素和数据存储

【答案】C

1. 关于软件体系结构描述语言，以下说法错误的是（ ）。

【A.】作为一种形式化语言，它为软件系统的概念体系结构建模提供了具体语法和概念框架

【B.】基于底层语义的工具为体系结构的表示、分析、进化、细化、设计过程等提供支持

【C.】软件体系结构描述语言的三个基本元素包括构件、连接件、体系结构配置

【D.】编程语言、面向对象的建模符号、形式化说明语言均属于软件体系结构描述语言

【答案】D

1. 关于调用/返回风格的体系结构中的主-子程序体系结构风格，以下说法错误的是（ ）。

【A.】由于单线程控制，计算的顺序得以保障

【B.】有用的计算结果在同一时刻会产生多个

【C.】单线程的控制可以直接由程序设计语言来支持

【D.】子程序的正确性与它调用的子程序的正确性有关

【答案】B

1. 关于调用/返回风格的体系结构中的分层体系结构风格，以下说法错误的是（ ）。

【A.】支持基于抽象程度递增的系统设计，使得设计者可以把复杂系统按递增的步骤进行分解

【B.】支持功能增强，每一层至多和相邻的上下层交互，因此功能的改变最多影响相邻的上下层

【C.】每个系统都可以很容易地划分为分层体系结构

【D.】支持复用，只要提供的服务接口定义不变，同一层的不同实现可以交换使用

【答案】C

1. 在进行软件设计过程中，以下关于“逐步求精”与“抽象”的说法，错误的是（ ）。

【A.】过程抽象是把完成一个特定功能的动作序列抽象为一个过程名和参数表，以后通过指定过程名和实际参数调用此过程

【B.】数据抽象是把一个数据对象的定义抽象为一个数据类型名，用此类型名可定义多个具有相同性质的数据对象

【C.】逐步求精的主要思想是，针对某个功能的宏观描述用逐步求精的方法不断地分解，逐步确立过程细节，直至该功能用程序语言描述的算法实现为止

【D.】逐步求精的每一步是用更为简单的描述代替上一层次的描述

【答案】D

1. 信息隐藏是使模块化开发活动更加简单的重要因素。以下关于信息隐藏的说法错误的是（ ）。

【A.】为达到信息隐藏的目的，模块所含信息对于那些不需要这些信息的模块不可访问

【B.】为达到信息隐藏的目的，每个模块只完成一个相对独立的特定功能

【C.】为达到信息隐藏的目的，每个模块可与其他模块共同完成特定功能

【D.】为达到信息隐藏的目的，模块之间仅仅交换那些为完成系统功能必须交换的信息

【答案】C

1. 内聚是一个模块内部各成分之间关联程度的度量。以下哪项不属于低等级内聚（ ）。

【A.】偶然性内聚

【B.】逻辑性内聚

【C.】时序内聚

【D.】过程性内聚

【答案】D

1. 耦合是模块之间依赖程度的度量。以下关于耦合的说法，错误的是（ ）。

【A.】如果两模块中任一个需依赖对方才能工作，则称这两模块为非直接耦合

【B.】如果两模块间通过参数交换信息，而信息仅限于数据，则称这两模块为数据耦合

【C.】若两个模块传递的信息中含有控制信息，则称这两模块耦合度为控制耦合

【D.】当一个模块使用另一模块内部的数据或控制信息，或一个模块直接转移到另一模块内部，则称这两模块为内容耦合

【答案】A

1. 以下关于软件结构评价的说法，错误的是（ ）。

【A.】一个软件的深度说明其控制的层数

【B.】一个软件的宽度说明其控制的层数

【C.】一个模块的扇出率指该模块直接控制的其他模块数

【D.】一个模块的扇入率指能直接控制该模块的模块数

【答案】B

1. 以下关于软件结构评价的说法，错误的是（ ）。

【A.】一个软件的深度说明其控制的层数

【B.】一个软件的宽度说明其控制的跨度

【C.】一个模块的扇出率指该模块直接控制的其他模块数

【D.】一个模块的扇入率指该模块直接控制的其他模块数

【答案】D

1. 关于用户界面分析和设计模型，以下说法错误的是（ ）。

【A.】用户模型给出了系统用户的介绍

【B.】设计模型用以理解潜在的系统用户

【C.】心智模型是用户对所使用系统的想象

【D.】实现模型结合了基于计算机的系统外部显示以及所有描述系统语法和语义的信息

【答案】B

1. 以下哪项不属于常用的用户界面原型构建方式( )。

【A.】脚本驱动方式

【B.】可视化的程序语言

【C.】手工勾画

【D.】基于因特网的原型

【答案】D

1. 在UML2.0建模机制中，（ ）属于结构建模。

【A.】活动图

【B.】类图

【C.】顺序图

【D.】通信图

【答案】B

1. 在UML2.0建模机制中，（ ）既属于行为建模又可以归属为交互图。

【A.】活动图

【B.】用例图

【C.】顺序图

【D.】状态图

【答案】C

1. 在UML2.0类图中，关于“关联关系”描述错误的是（ ）。

【A.】关联关系可以具有方向性

【B.】关联未必是双向的，必须用“有向连线”表示

【C.】在关联关系上可以写明关联名称，每个类在关联中的角色以及两个类的实例在实际关联中的数量对应关系

【D.】在关联关系表示中，用星号“\*”表示多个，用n表示n个实例，n..m表示数量的范围

【答案】B

1. 在UML2.0类图中，下面哪个符号表示泛化（继承）关系（ ）。

【A.】

【B.】

【C.】

【D.】

【答案】B

1. 面向对象的设计任务不包括以下哪个内容（ ）。

【A.】系统整体设计

【B.】对象设计

【C.】功能设计

【D.】在消息设计

【答案】C

1. 面向对象的设计任务不包括以下哪个内容（ ）。

【A.】系统整体设计

【B.】对象设计

【C.】功能设计

【D.】在消息设计

【答案】C

1. 关于面向对象程序设计的特征，以下说法错误的是（ ）。

【A.】每个对象有自己的惟一标识，以便区别属于同一个类的不同对象

【B.】所有待处理的内容都表示成对象

【C.】对象之间依靠相互发送信息或响应信息实现通信

【D.】每个对象的标识不是唯一的

【答案】D

1. 关于面向对象程序设计的基本概念，以下说法错误的是（ ）。

【A.】抽象是解决任何问题所采用的基本策略，是人类认识世界的本能形式

【B.】封装是指将现实世界中某个客体的属性与行为聚集在一个逻辑单元内部的机制

【C.】对象是用来描述现实世界中客体的部件，是面向对象软件系统在运行时的基本单位

【D.】对象是面向对象程序的惟一构造单位，也是抽象数据类型的具体实现

【答案】D

1. 关于面向对象程序设计的基本概念，以下说法错误的是（ ）。

【A.】抽象是指从许多事物中，舍弃本质属性，抽取个别的、非本质的属性的过程

【B.】消息是一个对象要求另一个对象实施某项操作的请求

【C.】继承是类之间的一种常见关系，它为共享数据和操作提供了良好的机制

【D.】不同的类对象收到同一个消息可以产生完全不同的响应效果,这种现象叫做多态

【答案】A

1. 在设计模式中，（ ）的核心思想是：定义一个用于创建产品对象的工厂接口，由子类决定生产什么产品。

【A.】工厂方法

【B.】单例

【C.】组合

【D.】代理

【答案】A

1. 在设计模式中，（ ）的核心思想是，一个类必须自行创建一个实例且仅能有一个实例，并向系统提供该实例。完成上述行为的操作是将类的构造方法声明为私有类型，就可以防止用户利用类的构造方法创建出多个实例。

【A.】工厂方法

【B.】单例

【C.】组合

【D.】代理

【答案】B

1. 在设计模式中，（ ）以树形结构将多个对象组合起来，用来表示“整体-部分”的层次结构，其核心思想是通过定义抽象构件类，既可用来表示叶子，又可用来表示容器，用户针对抽象构件类进行编程，能够对叶子和容器进行容易处理。

【A.】工厂方法

【B.】单例

【C.】组合

【D.】代理

【答案】C

1. 在设计模式中，（ ）的核心思想是，引入一个新的代理对象，该对象在客户端对象和目标对象之间起到桥梁的作用，它隐藏客户不能看到的内容和服务或者为客户添加需要的额外服务。

【A.】工厂方法

【B.】单例

【C.】组合

【D.】代理

【答案】D

1. 在设计模式中，（ ）将一个类的接口转换为用户希望的另一个接口，解决了一些类之间由于接口不兼容而无法一起工作的问题。

【A.】适配器

【B.】桥接

【C.】享元

【D.】外观

【答案】A

1. 在设计模式中，（ ）的核心思想是，通过将列表对象中对列表的访问和遍历从中抽离出来并放入到一个迭代器对象中，提供一种访问聚合对象中所有元素的方法，而不暴露该对象的内部表示。

【A.】工厂方法

【B.】迭代器

【C.】命令

【D.】代理

【答案】B

1. 在设计模式中，（ ）的核心思想是，对象是对数据和函数的封装，当一个类包含了太多的函数（或称操作）时，倾向于将其拆分为多个相互协作的类，每个协作类描述一部分行为，包含原来的一部分数据和函数，但这种拆分有一个副作用，因为各协作对象很可能会共享部分数据，所以需要维护相关对象在数据上的一致性。通过使用Observer模式，能够为相关对象制定一个交互协议，专门用作数据的一致性维护。

【A.】工厂方法

【B.】迭代器

【C.】观察者

【D.】代理

【答案】C

1. 在设计模式中，（ ）的核心思想是，通过定义一些用来封装不同算法的相互独立的类，且每一个类都封装了一个具体的算法并将其逐一封装起来，使得它们之间能够互相替换。

【A.】策略

【B.】迭代器

【C.】组合

【D.】代理

【答案】A

1. 在设计模式中，（ ）的核心思想是，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，便于日后将对象复原到先前保存的状态。

【A.】策略

【B.】备忘录

【C.】组合

【D.】代理

【答案】B

1. 在设计模式中，（ ）表示一个作用于某个对象结构中各元素的操作，它使我们能够无须改变各元素的类而定义作用在这些元素的新操作，由两部分组成：被访问元素和访问者。

【A.】适配器

【B.】迭代器

【C.】观察者

【D.】访问者

【答案】D

1. 数据流图就是用来刻画数据流和转换的信息系统建模技术，它用简单的图形记号分别表示数据流、转换、数据源以及外部实体,下面( ) 是外部实体的图形记号。

【A.】

【B.】

【C.】

【D.】

【答案】A

1. 数据流图就是用来刻画数据流和转换的信息系统建模技术，它用简单的图形记号分别表示数据流、转换、数据源以及外部实体,下面( ) 是转换的图形记号。

【A.】

【B.】

【C.】

【D.】

【答案】B

1. 数据流图就是用来刻画数据流和转换的信息系统建模技术，它用简单的图形记号分别表示数据流、转换、数据源以及外部实体,下面( ) 是数据流的图形记号。

【A.】

【B.】

【C.】

【D.】

【答案】C

1. 数据流图就是用来刻画数据流和转换的信息系统建模技术，它用简单的图形记号分别表示数据流、转换、数据源以及外部实体,下面( ) 是数据源的图形记号。

【A.】

【B.】

【C.】

【D.】

【答案】D

1. 在软件进化策略选择方面，关于遗留系统的评价结果以下说法错误的是( )

【A.】对于低业务价值、低系统质量的系统，因使用这些系统开销较小，只需对其进行常规维护

【B.】对于高业务价值、低系统质量的系统，它们对于业务来说非常重要，不能放弃；但因低质量系统维护起来代价会比较大，应该进行进化这类系统来提髙质量

【C.】对于低业务价值、高系统质量的系统，它们对业务没有太大贡献，但其维护起来代价较小。若变更代价不大，可对这类系统进行常规的维护；若需要变更且代价高昂，则应放弃这些系统

【D.】对于高业务价值、高系统质量的系统，由于业务的需要它们必须要继续使用；同时它们有较高的质量，所以不需要投入大量的资金来更新系统，只需要进行常规维护

【答案】A

1. 在软件体系结构的迭代过程中，( )活动是从文档、代码、专家等方面获取所有可能得到的软件信息或设计要素

【A.】提取

【B.】分类

【C.】合并

【D.】融合

【答案】A

1. 在软件体系结构的迭代过程中，( )活动是把提取出的信息按照不同视点进行分类；除了静态和动态视点的划分方式外，还存在其他更具体的划分。

【A.】提取

【B.】分类

【C.】合并

【D.】融合

【答案】B

1. 在软件体系结构的迭代过程中，( )活动是对分类后每个视点中的信息进行分析和合并，形成一种系统、一致的方式来描述软件关于该视点的特征。

【A.】提取

【B.】分类

【C.】合并

【D.】融合

【答案】C

1. 在软件体系结构的迭代过程中，( )活动是把分别合并后的各个视点进行融合，检查各个视点之间的一致性，并把这些视点统一称为一个整体的软件体系结构描述。

【A.】提取

【B.】分类

【C.】合并

【D.】融合

【答案】D

1. 重构就是在不改变软件系统外部行为的前提下，改善它的内部结构，提高其可理解性，降低其修改成本。其中，( ) 的目标是生成具有相同功能、但质量比原来程序更高的代码。

【A.】文本重构

【B.】逻辑重构

【C.】代码重构

【D.】数据重构

【答案】C

1. 重构就是在不改变软件系统外部行为的前提下，改善它的内部结构，提高其可理解性，降低其修改成本。其中，( ) 从一种格式到另一种格式的转换，以实现空间数据在结构、格式、类型上的统一，以及多源和异构数据的联接与融合。

【A.】文本重构

【B.】逻辑重构

【C.】代码重构

【D.】数据重构

【答案】D

1. 在软件再工程过程模型中，( )是对程序进行分析，以便在比源代码更高的抽象层次上创建程序的某种表示的过程。其需要从现存的程序中抽取数据设计、软件体系结构设计和过程设计等信息。

【A.】库存目录分析

【B.】文档重构

【C.】逆向工程

【D.】代码重构

【答案】C

1. 在软件再工程过程模型中，( )通过用重构工具分析源代码，标注出和结构化程序设计概念相违背的部分，然后重构有问题的代码，复审和测试生成的重构代码以保证没有引入异常并更新代码文档。

【A.】逆向工程

【B.】正向工程

【C.】数据重构

【D.】代码重构

【答案】D

### 多选题

1. 为了开发出低成本、高质量的软件产品，在软件开发过程中必须遵循哪些软件工程的原则（ ）。

【A.】采取固定的开发模型，不考虑易变需求

【B.】采用适合的设计方法

【C.】提供高质量的工程支持

【D.】重视开发过程的管理

【答案】ABD

1. 软件生命周期包括（ ）等阶段。

【A.】计划

【B.】开发

【C.】测试

【D.】维护

【答案】ABD

1. 瀑布模型的优点主要包括（ ）。

【A.】方便为项目提供了按阶段划分的检查点

【B.】当前一阶段完成后，只需要去关注后续阶段

【C.】瀑布模型提供了一个模板，该模板使得分析、设计、编码、测试和支持的方法可以在该模板下有一个共同的指导

【D.】瀑布模型的各个阶段划分完全固定，阶段之间产生大量的文档，极大地增加了工作量

【答案】ABC

1. 瀑布模型的缺点主要包括（ ）。

【A.】由于开发模型是线性的，凡后一阶段出现的问题需要通过前一阶段的重新确认来解决

【B.】模型缺乏灵活性，无法解决需求模糊或需求经常变动的问题

【C.】最终产品往往反映用户的初始需求而不是最终需求

【D.】瀑布模型的各个阶段划分完全固定，阶段之间产生大量的文档，极大地增加了工作量

【答案】ABCD

1. 快速原型模型的优点主要包括（ ）。

【A.】克服瀑布模型的缺点，减少因软件需求不明确带来的开发风险

【B.】适合预先不能确切定义需求的软件系统的开发

【C.】快速原型模型使总的开发费用降低，时间缩短

【D.】快速原型模型所选用的开发技术和工具不一定符合主流的发展

【答案】ABC

1. 快速原型模型的缺点主要包括（ ）。

【A.】快速原型模型所选用的开发技术和工具不一定符合主流的发展

【B.】快速原型模型建立起来的系统结构加上连续的修改可能会导致产品质量低下

【C.】快速原型模型使用前提是要有展示性的产品原型，一定程度上可能会限制开发人员的创新

【D.】用户不知道快速原型模型是临时搭起来的，也不知道为了使其尽快运行还没考虑软件的总体质量或今后的可维护性问题

【答案】ABCD

1. 从审视角度看，关于软件体系结构的定义，说法正确的是（ ）。

【A.】从概念角度，软件体系结构描述系统的主要构件以及它们之间的关系

【B.】从模块角度，软件体系结构包含功能分解与层次结构

【C.】从运行角度，软件体系结构描述了一个系统的动态结构

【D.】从代码角度，软件体系结构描述了各种代码和库函数在开发环境中的组织

【答案】ABCD

1. 关于刻画软件体系结构的场景视图模型，其缺陷主要包括( )。

【A.】不能充分表达系统的体系结构风格

【B.】不能充分体现数据

【C.】不能充分体现系统要素之间关系

【D.】实现体系结构模型时，缺乏构造视图和建立视图之间关系的指导信息

【答案】ABCD

1. 软件体系结构的描述方法主要包括( )。

【A.】图形表达工具

【B.】模块内连接语言

【C.】基于软构件的系统描述语言

【D.】体系结构描述语言

【答案】ABCD

1. 软件体系结构描述语言的三个基本元素是（ ）。

【A.】构件

【B.】连接件

【C.】体系结构配置

【D.】角色

【答案】ABC

1. 软件体系结构的设计原则主要包括( )。

【A.】合适性

【B.】结构稳定性

【C.】可扩展性

【D.】可复用性

【答案】ABCD

1. 软件体系结构风格是描述某一特定应用领域中系统组织方式的惯用模式,其关键要素包括( )。

【A.】提供一个词汇表

【B.】定义一套配置规则

【C.】定义一套语义解释原则

【D.】定义对基于这种风格的系统所进行的分析

【答案】ABCD

1. 以下关于软件设计可能对最终软件产品质量产生的影响，正确的是( )。

【A.】正确性：软件设计中如果存在错误，会导致软件实现的错误，影响最终软件的正确性

【B.】可靠性：如果软件设计中错误、缺陷较多，那么会导致最终软件的可靠性降低

【C.】运行效率：软件运行的效率可能会受到设计中选择的算法、数据结构的影响

【D.】可移植性：良好的设计使得与运行平台和环境相关的代码最小化，并进行良好的封装和接口定义，这使得软件移植更加容易

【答案】ABCD

1. 以下关于软件设计可能对最终软件产品质量产生的影响，正确的是( )。

【A.】可靠性：如果软件设计中错误、缺陷较多，那么会导致最终软件的可靠性降低

【B.】可维护性：软件可能会因为发生错误或环境变化而进行维护，具有良好结构的软件设计会使得维护人员更容易理解软件结构

【C.】可移植性：良好的设计使得与运行平台和环境相关的代码最小化，并进行良好的封装和接口定义，这使得软件移植更加容易

【D.】可复用性：由于软件设计描述了软件如何分解，子系统和模块之间的接口如何定义，因此对于软件模块的复用具有重要影响

【答案】ABCD

1. 用户界面设计基本原则包括( )。

【A.】用户熟悉程度

【B.】一致性

【C.】使惊讶最小化

【D.】用户帮助

【E.】可恢复性

【F.】用户多样性

【答案】ABCDEF

1. 用户交互实质上就是用户向计算机系统发出命令和相应的数据来完成相应的任务。用户交互方式主要包括( )。

【A.】直接操作

【B.】菜单选择

【C.】表格填写

【D.】命令语言

【答案】ABCD

1. 常用的用户界面原型的构建方式包括( )。

【A.】脚本驱动方式

【B.】可视化的程序语言

【C.】手工勾画

【D.】基于因特网的原型

【答案】ABD

1. 统一建模语言UML的目标是定义一种通用建模语言并对这些建模语言做出简单的说明，供系统建模者理解与使用。其主要特点包括（ ）。

【A.】UML统一了各种方法对不同类型的系统、不同开发阶段以及不同内部概念的各种观点，有效地消除了各种建模语言之间不必要的差异

【B.】UML建模能力比其它面向对象建模方法更加强大，不仅适合于一般系统的开发，还特别适用于对并行和分布式系统进行建模

【C.】UML是一个开发过程

【D.】UML是一种建模语言

【答案】ABD

1. 统一建模语言UML在表达能力、对新技术的包容能力和扩展性等方面具备的优势包括（ ）。

【A.】提供了一种通用、表达能力强的可视化建模语言，用于描述应用问题的需求模型、设计模型和实现模型

【B.】提供了对核心概念的扩展机制，用户可自行加入核心概念中没有提供的概念和符号，可为特定应用领域提出具体的概念、符号表示以及约束

【C.】增强面向对象工具之间的相互可操作性，便于对不同系统进行集成

【D.】支持较高抽象层次开发所需的各种概念，便于系统的重用

【答案】ABCD

1. 在面向对象软件设计过程中，持久数据模型设计主要包括以下几个步骤( )。

【A.】确定设计模型中需要持久保存的类的对象及其属性，其中实体类是主要关注对象

【B.】确定持久存储的数据之间的组织方式

【C.】确定数据模型中的操作行为

【D.】进一步优化持久数据操作的性能

【答案】ABCD

1. 在面向对象软件设计过程中，部署模型设计需要考虑以下几个要素( )。

【A.】最终开发完成的软件包括哪些制品形式

【B.】软件运行环境存在哪些类型的物理节点

【C.】不同节点之间的连接和通信形式是什么

【D.】软件制品应该如何在物理节点上进行部署，即它们的部署映射关系

【答案】ABCD

1. 在面向数据流的软件设计过程中，数据对象的属性包括( )。

【A.】物理属性

【B.】命名性属性

【C.】描述性属性

【D.】引用性属性

【答案】BCD

1. 在实体关系图进行复杂数据建模过程中，为确保模型的一致性并消除数据冗余，分析人员要掌握 ( )。

【A.】数据对象的任何实例对每个属性必须有且仅有一个属性值

【B.】属性是原子数据项，不能包含内部数据结构

【C.】如果数据对象的关键属性多于一个，那么其他的非关键属性必须表示整个数据对象而不是部分关键属性的特征

【D.】所有的非关键属性既可以表示整个对象，也可以表示部分属性的特征

【答案】ABC

1. 数据流图必须与描述并组织数据条目的数据字典配套使用。通常，数据字典中的每一条数据条目包含( )。

【A.】在数据流图中标识数据流、数据源或外部实体的名称与别名

【B.】数据类型

【C.】所有以它作为输入流或输出流的转换列表

【D.】如何使用该数据条目的简要说明

【答案】ABCD

1. 在面向对象的数据模型设计中，持久数据模型设计主要包括以下几个步骤（ ）。

【A.】确定设计模型中需要持久保存的类的对象及其属性，其中实体类是主要关注对象

【B.】确定持久存储的数据之间的组织方式

【C.】确定数据模型中的操作行为，例如数据完整性验证、数据读取、存储与更新、数据求和、求数据平均值

【D.】进一步优化持久数据操作的性能，例如使用数据索引、存储过程、触发器等方式

【答案】ABCD

1. 在进行部署模型设计时，一般需要考虑（ ）。

【A.】最终开发完成的软件包括哪些制品形式

【B.】软件运行环境存在哪些类型的物理节点

【C.】不同的节点之间的连接和通信形式是什么

【D.】软件制品应该如何在物理节点上进行部署

【答案】ABCD

1. 在进行部署模型设计时，一般需要考虑（ ）。

【A.】最终开发完成的软件包括哪些制品形式

【B.】软件运行环境存在哪些类型的物理节点

【C.】物理节点应采用本地部署，还是考虑采用云计算环境上进行部署

【D.】软件制品应该如何在物理节点上进行部署

【答案】ABD

1. 以下哪项不属于属性和操作的作用范围有哪几种（ ）。

【A.】public

【B.】protected

【C.】static

【D.】private

【答案】ABD

1. 四位著名的软件工程学者（Erich Gamma, Richard Helm, Ralph Johnson和John Vlissides）在《设计模式：可复用面向对象软件基础》一书中总结出软件工程设计模式的基本要素包括( )。

【A.】模式名称

【B.】问题

【C.】解决方案

【D.】效果

【答案】ABCD

1. 软件设计模式的应用对软件开发所起的重大作用包括( )。

【A.】代码重用性

【B.】节约设计实践，提高设计效率

【C.】可读性

【D.】可扩展性

【答案】ABCD

1. 依照软件设计的目的和范围准则，软件设计模式包括( )。

【A.】创建型设计模式

【B.】结构型设计模式

【C.】行为型设计模式

【D.】创新型设计模式

【答案】ABC

1. 对于遗留软件，可以选择的进化策略包括( )。

【A.】完全放弃该软件

【B.】继续维护系统

【C.】通过软件工程技术转换系统以提高系统的可维护性

【D.】开发一个新系统来代替遗留系统

【答案】ABCD

1. 在软件进化策略选择方面，遗留系统的评价结果根据业务价值和系统质量可分为( )。

【A.】低业务价值，低系统质量

【B.】高业务价值，低系统质量

【C.】低业务价值，高系统质量

【D.】高业务价值，高系统质量

【答案】ABCD

1. 为了评价软件系统的业务价值，需要从以下哪几个主要方面对系统进行考察( )。

【A.】系统的使用

【B.】系统支持的业务过程

【C.】系统的可靠性

【D.】系统的输出

【答案】ABCD

1. 软件再工程是指通过对目标系统的检查和改造，将逆向工程、重构和正向工程组合起来，将现存系统重新构造为新的形式，以开发出质量更高、维护性更好的软件。其具体目标包括( )。

【A.】为追加、增强功能做准备

【B.】提高可维护性

【C.】软件的移植

【D.】提高可靠性

【答案】ABCD

1. 完整软件体系结构的迭代过程包括以下几种活动( )。

【A.】提取：即从文档、代码、专家等方面获取所有可能得到的软件信息或设计要素

【B.】分类：把提取出的信息按照不同视点进行分类

【C.】合并：对分类后每个视点中的信息进行分析和合并，形成一种系统、一致的方式来描述软件关于该视点的特征

【D.】融合：把分别合并后的各个视点进行融合，检查各个视点之间的一致性，并把这些视点统一称为一个整体的软件体系结构描述

【答案】ABCD

1. 对软件实施代码和数据重构的优点主要包括( )。

【A.】改进软件的设计

【B.】提高代码质量，更容易被理解

【C.】帮助尽早发现错误

【D.】可以提高开发速度

【答案】ABCD

1. 软件设计的发展趋势主要包括( )。

【A.】服务性：以人为本的服务是计算机软件开发最基本的目的，是未来发展的趋势

【B.】开放化：只有保持信息以及软件的开放化，企业才能拥有更加全面的视野，取各行业之间的优势，弥补自身的不足

【C.】智能化和网络化：大多数的计算机软件已经朝智能化发展，这代表着计算机软件开发技术在未来向智能化方向转化；在软件开发技术研究中，物联网就是主要的研究内容

【D.】复杂化：计算机软件设计因功能的完备性趋于复杂

【答案】ABC

1. 软件再工程通过改造遗留系统，使系统变得容易理解，进而提高系统的可维护性。具体包括( )。

【A.】系统文档的再建、修改

【B.】更新系统的结构

【C.】用先进的程序设计语言转换源代码

【D.】数据再工程

【答案】ABCD

### 简答题

1. 软件设计包括软件的结构设计、数据设计、接口设计和过程设计。作为一个完整的软件设计方案，请简要阐述软件设计包含的组成要素。

【答案】（1）目标描述。设计方案需要清楚地描述它要解决的问题和将要达到的目标。

（2）设计约束。设计目标的获得通常要在某些约束的限制范围内，这些约束定义了要解决问题的解空间。

（3）设计原理。工程化的设计活动必须以一定的科学原理和技术手段为基础，它们将表明问题可以通过该设计得到合理的解决。

（4）开发规划。工程化的设计活动不仅需要让人知道该设计能解决问题，还需要让人们了解该设计在实际开发中是可行的，以及如何实现该设计。

（5）使用描述。产品经常在某些特定条件下才能被可靠、髙效地使用，并达到最初的设计目标。因此，描述产品如何使用非常重要。

1. 简要阐述软件体系结构描述语言（ADL）的三个基本元素及其作用。

【答案】（1）构件：计算或数据存储单元。

（2）连接件：用于构件之间交互建模的体系结构构造块及其支配这些交互的规则。

（3）体系结构配置：描述体系结构的构件与连接件的连接图。

1. 软件体系结构设计方法是指通过一系列的设计活动，获得满足系统功能性需求，并且符合一定非功能性需求约束的软件体系结构模型。简述不同类型的软件体系结构设计方法及其特点。

【答案】（1）工业驱动的方法。工业驱动的体系结构设计方法从方法的工件描述中提取体系结构描述。工件驱动的体系结构设计方法的例子包括广为流行的面向对象分析和设计方法OMT。

（2）用例驱动的方法。用例驱动的体系结构设计方法主要从用例导出体系结构抽象。一个用例，是指系统进行的一个活动系列，它为参与者提供一些结果值，参与者通过用例使用系统。参与者和用例共同构成了用例模型。

（3）模式驱动的方法。模式驱动的体系结构设计方法从模式导出体系结构抽象。该方法在处理范围广泛的体系结构问题时，模式库可能不够充足，且对模式的选择仅仅依靠通用知识和软件工程师的经验。

（4）领域驱动的方法。在领域驱动的体系结构设计方法中，体系结构抽象是从领域模型导出来的。领域模型可以有很多种不同的表示方法，比如，类、实体关系图、框架、语义网络和规格等。

1. 请阐述软件开发过程中的抽象原则和信息隐藏原则。

【答案】（1）抽象是指抽取事物最基本的特征和行为，忽略非基本的细节，采用分层次抽象的办法，可以控制软件开发过程的复杂性，有利于软件的可理解性和开发过程的管理。

（2）信息隐藏是将模块中的软件设计决策封装起来的技术，是使模块化开发活动更加简单的重要因素。在设计和确定模块时信息隐藏使一个模块的开发者不必看到其他模块的内部，只需知道其接口即可。这使得每个模块的开发人员所要处理的复杂性显著降低。

1. 请阐述软件开发过程中模块化设计原则的要点。

【答案】（1）模块是程序中逻辑上相对独立的成分，它是一个独立的编程单位，应有良好的接口定义，模块化有助于信息隐藏和抽象，有助于表示复杂软件系统。

（2）模块的大小要适中，模块过大会导致模块内部复杂的增加，不利于模块的调试和重用，也不利于对模块的理解和修改。

（3）模块太小会导致整个系统的表示过于复杂，不利于控制复杂性。

1. 请阐述软件界面设计的基本原则。

【答案】（1）用户熟悉程度：界面应该采用经常使用系统用户熟悉的术语和概念。

（2）一致性：界面必须一致，在任何可能的情况下，相同的操作应该以同样的方式被激活。

（3）使惊讶最小化：尽量避免使用户对系统的行为感到惊讶。

（4）可恢复性：界面应该为用户提供错误恢复机制。

（5）用户帮助：界面应该在错误发生时提供有意义的反馈，并提供上下文敏感用户帮助系统。

（6）用户多样性：界面应该为不同类型的用户提供恰当的交互方式。

1. 简要描述基于UML的分析与设计过程。

【答案】（1）用例分析与设计

（2）概念模型与顶层架构设计

（3）用户界面设计

（4）数据模型设计

（5）设计精化

（6）类设计

（7）部署模型设计

1. 在创建用户需求的数据流模型的过程中，简要描述分析人员应遵循的原则。

【答案】（1）首先建立顶级数据流图，其中只含有一个代表目标软件系统整体处理功能的转换。

（2）对用户需求的文字描述进行语法分析，其中的名词和名词短语构成潜在的外部实体、数据源或数据流、动词构成潜在的处理功能。

（3）采用通常的功能分解方法，按照“强内聚，松耦合”原则，逐个对处理功能进行精化，与此同时逐步完成对数据流的精化，并针对被精化的处理功能生成下一级数据流图。

（4）在精化过程中，必须维持各级数据流图的平衡。

（5）精化过程应适可而止，避免涉及软件设计细节。

1. 在面向对象的软件设计中，类设计的目的是对各种设计模型中出现的类进行细化设计，以使它们精细至能够直接提交给软件构造阶段进行编码实现。请简述类设计的主要任务。

【答案】（1）对类的属性与操作进行精化。

（2）对类的对象实例在其生存周期中对外部消息的响应和状态变化过程进行建模。

（3）对类中重要操作的实现过程成算法进行描述。

1. 简述启发式设计策略的主要经验。

【答案】（1）改造程序结构，减小耦合度，提高内聚度。

（2）改造程序结构,减少高扇出,在增加程序深度的前提下追求高扇入。

（3）改造程序结构,使任一模块的作用域在其控制域之内。

（4）改造程序结构，减少接口的复杂性和和冗余程度，提高协调性。

（5）模块功能应该可预言，避免对模块施加过多限制。

（6）改造程序结构，追求单入口单出口的模块。

（7）为满足设计或可移植性的要求，把某些软件用包的形式封装起来。

1. 简述软件设计模式的应用对软件开发的重要作用。

【答案】（1）代码重用性。大部分模式都兼顾了系统的可重用性，这使得开发人员可以更好地重用一些已有的设计方案、功能模块甚至一个完整的软件系统，提高最佳设计实践的复用性。

（2）节约设计实践，提高设计效率。模式是众多专家从诸多优秀的软件系统中总结出的设计方案，使用这些设计方案能为开发人员提供设计思路，有助于开发人员提高开发和设计效率，帮助设计新手快速系统地学习专家的成功设计经验。

（3）可读性。模式为设计人员提供了一套统一的设计词汇，包括设计、管理和组织方面，同时也为设计人员提供了一个规范和描述抽象事物的标准，以便开发人员之间的沟通和交流。

（4）可扩展性。随着软件规模的日益增大，软件寿命逐渐延长，软件系统的可扩展性也越来越重要，许多模式将有助于提高系统的灵活性和可扩展性，让开发人员在不修改或者在现有系统的基础上根据实际情况增加、删除或者替换相应的功能模块。

1. 软件设计模式能够帮助软件设计者更快地完成面向对象的软件设计。请简述软件设计模式选择的具体步骤。

【答案】人们在选择模式时可遵循以下步骤和原则。

（1）理解问题需求。问题的需求是进行模式选择的基础，通过对需求的分析可以找到多个模式，并将其形成相应的模式组。

（2）研究组内模式。需求分析得出的组内模式存在一些共同特性，但是不同的模式都有其独特的使用动机、意图和使用条件等等，因此对组内模式进行研究，研究模式之间的相互关联具有一定的实际意义。

（3）考虑设计模式是如何解决设计问题的。在系统的设计过程中会产生许多变化因素。为了支持这种变化，应明确设计模式在设计中支持哪些可变化因素，而这些可变因素是否会导致系统必须进行重新设计，基于此可找到所需的设计模式以尽量避免引起重新设计。

1. 软件设计模式能够帮助软件设计者更快地完成面向对象的软件设计。请简述软件设计模式使用的方法与原则。

【答案】（1）首先理解所选择的模式。特别注意模式的适用条件和使用效果部分，确定该模式是否适合对此类问题的求解。

（2）研究模式的结构部分、参与者部分和协作部分。确保对这个模式的类、对象以及它们之间的关联关系有一定的了解。

（3）看代码示例部分。研究该模式代码形式的具体例子将有助于设计人员更好的理解和实现模式。

（4）选择模式参与者的名字，使它们在应用上下文中有意义。

（5）定义类。声明它们之间的接口，建立它们的继承关系，定义代表数据和对象引用的实例变量。

（6）定义模式中专用于应用的操作名称。应用中的类、变量和对象的名称定义都会依赖于应用所涉及到的实体，对模式名称的定义也应该和实际中的具体操作相关联，要尽量和实际中的名字一致。

（7）实现执行模式中责任和协作的操作。实现部分提供线索指导人们进行实现。

1. 简述软件再工程的定义及突出特征。

【答案】（1）软件再工程是指通过对目标系统的检查和改造，其中包括设计恢复(库存目录分析)、再文档、逆向工程、程序和数据重构以及正向工程等一系列活动，旨在将逆向工程、重构和正向工程组合起来，将现存系统重新构造为新的形式，以开发出质量更高、维护性更好的软件。

（2）软件再工程有两个突出特征:一是比一次软件工程更迫切地需要计算机辅助支持，二是测试工作比例远大于一次软件工程。前者在再工程方法学研究和软件模式运动推动下可以找到自动化解决方案，后者则须强化对测试方法学体系的研究。

1. 简述软件再工程的具体目标。

【答案】（1）为追加、增强功能做准备。软件再工程通过对原系统用新的设计思想加以重新实现。使今后能方便地对系统进行功能的追加和增强。

（2）提高可维护性。经过不断维护日趋复杂的系统．经过软件的再工程和再设计，其模块划分会更合理，接口定义更清晰，文档更齐全，从而更易维护。

（3）软件的移植。软件再工程将一些优秀软件移植到新硬件平台、操作系统或语言环境中，从而使它们能够利用新环境的新特性，更好地发挥作用。

（4）提高可靠性。软件的维护修改容易引起“波动效应”，即修改一处，波及多方。软件的再工程可较好地解决这一问题。

1. 简述软件设计的发展与趋势。

【答案】（1）服务性：计算机软件技术的开发和发展主要是通过相关人员的操作完成的，要达到的目的是为了给人类提供最好的服务。以人为本的服务是计算机软件开发最基本的目的，是未来发展的趋势。

（2）开放化：从知识的角度来看，计算机技术属于高水平知识。长远发展眼光来看，只有保持信息以及软件的开放化，企业才能拥有更加全面的视野，取各行业之间的优势，弥补自身的不足，从而促进社会经济取得更深层次的发展。

（3）智能化和网络化：现阶段,大多数的计算机软件已经朝智能化发展，这代表着计算机软件开发技术在未来向智能化方向转化。在软件开发技术研究中，物联网就是主要的研究内容。

1. 开发者需要对遗留软件系统的实际情况进行评价，然后选择最合适的进化策略。简述对于不同的遗留软件，可选择的进化策略有哪些。

【答案】（1）完全放弃该软件。当系统所支持的业务对机构来说已经过时，或者说机构的业务根本不依赖于系统，可采用该策略。

（2）继续维护系统。当系统对机构业务仍然很重要，系统的运行也相对平稳，对系统的变更较少，且请求容易实现时，可以选择这种方案。

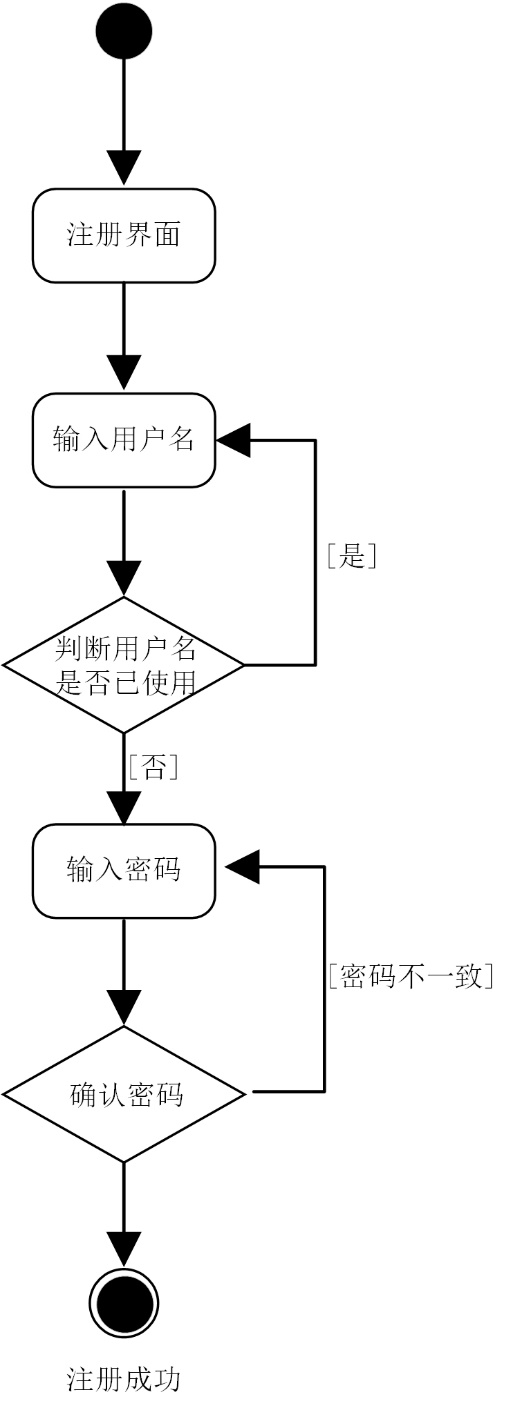
（3）通过软件工程技术转换系统（再工程）以提高系统的可维护性。对软件系统实施再工程以提高可维护性。当系统的质量、结构已经由于太多的变更变得很差，而且变更还会经常发生，则应该选择这个方案。

（4）开发一个新系统来代替遗留系统。当遗留系统由于变更或者运行环境的变化变得无法继续运行时，可以选择这个方案。同时，必须考虑开发新系统的成本和风险要在可以接受的范围内。

### 应用题

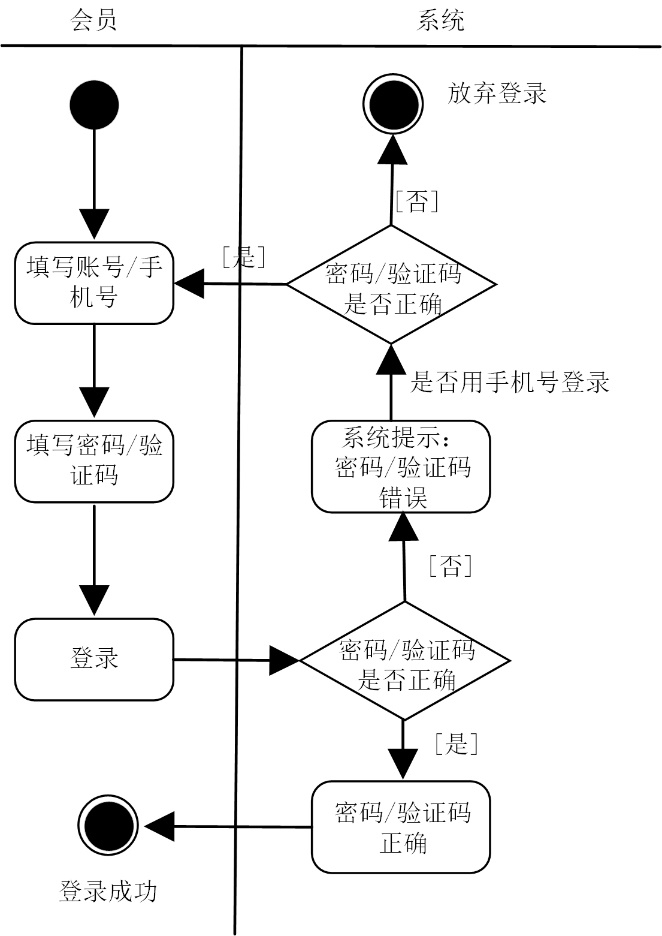
1. 背景：小明不是美团的注册会员，计划今天到美团网进行网上订餐。针对小明这位客户，京东在线服务系统需提供用户注册的服务。针对用户注册这一活动，画出相应的活动图。

【答案】



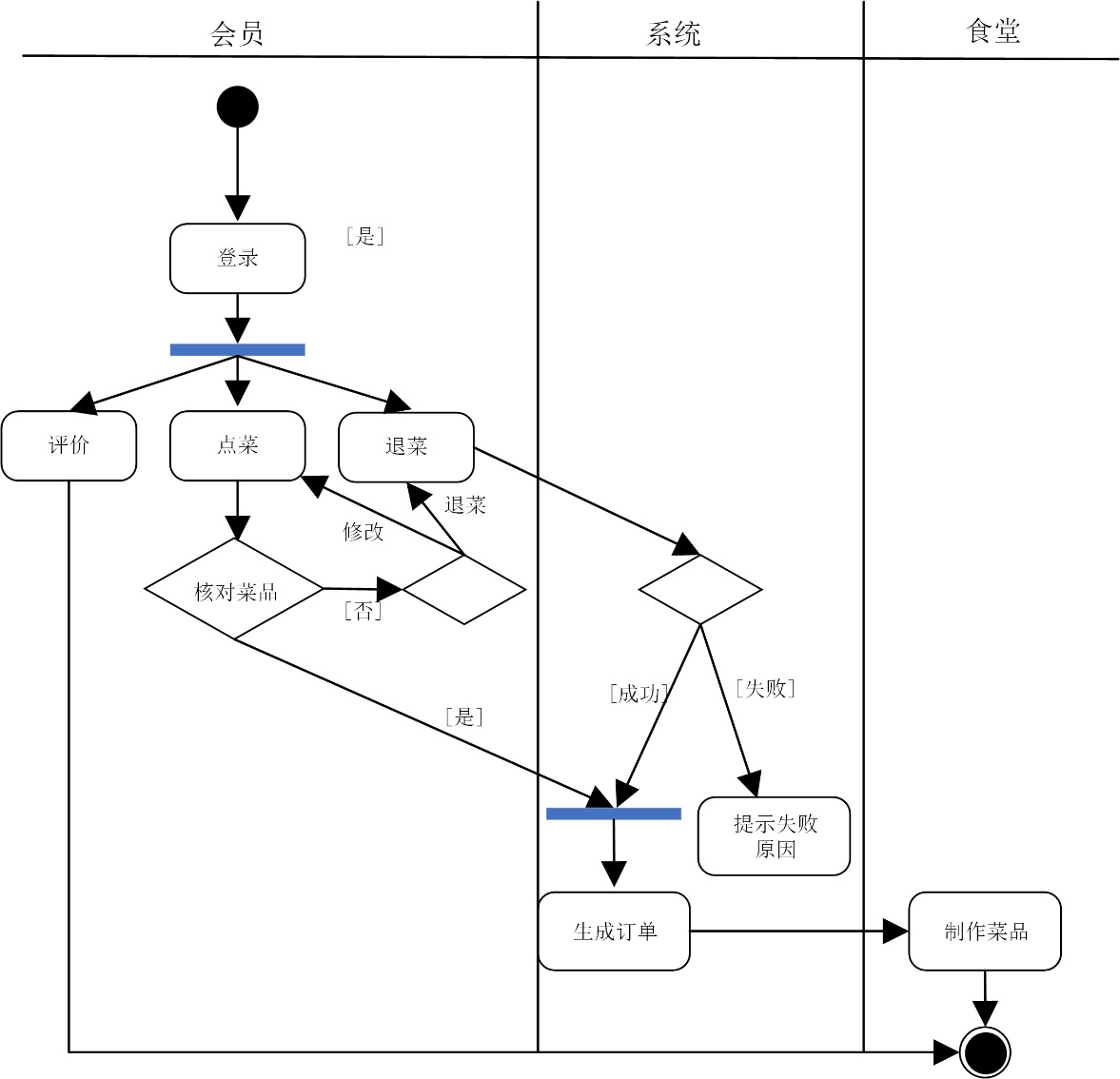
1. 背景：小明作为京东商城的注册会员，计划于双十一期间登录京东商城进行购物。针对用户登录这一活动，画出相应的活动图。

【答案】



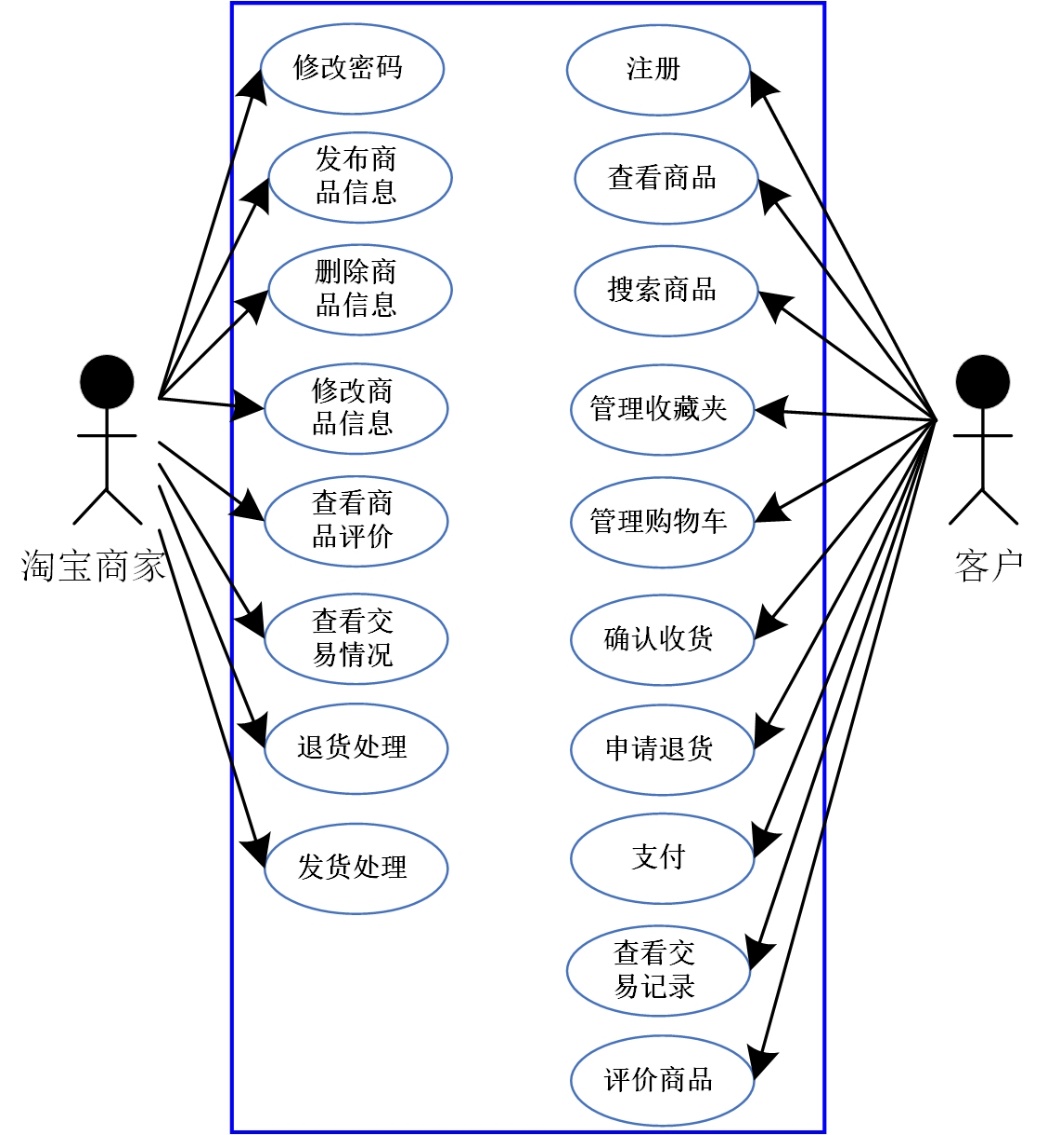
1. 背景：小明计划今天到美团网进行网上订餐。针对小明这位客户，京东在线服务系统需提供需提供网上订餐的服务。针对用户网上订餐这一活动，画出相应的活动图。

【答案】



4. 背景：小明在淘宝网拟选购一件商品，经过商品搜索，最终挑选了小李在淘宝开设的个人店面并购买该商品。针对上述过程，请以淘宝商家和客户为参与者，以发布、修改、删除商品信息，查看商品评价、交易情况，退货处理，发货处理，查看商品，搜索商品，确认收货，申请退货，支付，评价等动作作为功能性任务，画出相应的用例图。

【答案】



5.背景：学校拟建立学生信息管理系统，其功能需求包括四个方面：

（1）系统管理员登录后可以对班级的基本信息进行增加、删除、修改、查询等操作。学校领导登录后可以对班级基本信息进行查询操作。

（2）教师登录后可以对学生的考试成绩进行录入、删除、修改、查询等操作。学生登录后可以对考试成绩进行查询操作。

（3）学生登录后可以了解所有选修课程的具体信息，可以根据自己的需要选择不同课程。系统管理员登录后可以增加、修改、查询、删除选修课程。

（4）系统管理员可以对账号进行创建、设置、查看、删除等操作

请根据上述需求，画出该系统对应的用例图。

【答案】

