填空题：

1. A generic process framework for software engineering encompasses five activities：

Communication、Planning、Modeling、Construction、Deployment

1. Prescriptive process model

The waterfall model、incremental process model、evolutionary process model（prototyping）、the spiral model、concurrent model

1. The design model

The design model has four major elements: data、architecture、 interface 、components-level、deployment-level design model

1. Architecture Design

Systems that interoperate with the target system are represented as : Superordinate system,

Subordinate system, Peer-level systems, Actors-entities

1. Test strategies for conventional software

Unit testing, Integration testing, Regression testing, Smoke testing.

1. Validation Testing

Validation Testing begins at the culmination of integration testing,

1. System Testing

System testing is actually a series of different tests whose primary is to fully exercise the computer-based system;

简答题：

1. 惯用过程模型的类型以及优点、缺点、特点（不/适用于和种开发环境）
2. 瀑布模型：
   1. 优点：当从沟通到部署采用线性工作流方式的时候，可以清除的了解问题的需求。
   2. 缺点：实际项目很少遵照其提出的顺序；客户难以清楚的描述所有的需求（需要明确需求）；到最后才能有可执行的程序（耐心）；
   3. 工作场景：适用于需求确定、工作采用线性的方式完成。
3. 增量过程模型
   1. 优点：基于产品核心功能完成后（满足基本需求），可以添加许多的附加特性可以更好的满足客户需求。
   2. 缺点:
   3. 工作场景：适用于迫切需要提供一套功能有限的软件产品，在后续版本中细化、扩展。
4. 原型模型
   1. 优点：更好的理解需要做什么。
   2. 缺点:未考虑整体软件质量和长期的可维护性
   3. 工作场景：客户只提出了基本的功能，未详细定义功能及特性需求。
5. 螺旋模型：演进式软件过程模型
   1. 优点：具有快速开发越来越完善软件版本的潜力。
   2. 缺点：不能说明演进的方法是可控的；容易存在潜在的风险
   3. 工作场景（特点）：采用循环的方式逐步加深系统定义和实现的深度，同时降低风险；确定一系列的里程碑，确保利益相关者都支持可行的和令人满意的系统解决方案。
6. 协同模型
   1. 工作场景：可用于所有类型的软件开发；适合不同的工程团队共同开发的系统工程项目。
7. 数据建模概念
   1. 数据对象和面对对象类
      1. 数据对象定义了一个复合的数据项，合并一组独立的数据项并为数据项集合命名
      2. 面对对象类封装了数据对象，对这些属性定义数据的操作进行合并。
8. 类-职责-协作者模式（分类）
9. 实体类，模型或业务类，包含对用户来说很重要的信息。
10. 边界类用于创建用户可见的和在使用软件时交互的接口，管理实体对象对用户的表示方式。
11. 控制类自始至终管理“工作单元“。管理
    * 1. 实体类的创建和更新
      2. 当边界类从实体对象获取信息后的实例化
      3. 对象集合间的复杂通信
      4. 对象间或用户和应用系统间交换数据的确认。
12. 设计过程

软件设计是一个迭代的过程。软件质量指导原则和属性

1. 设计应展示出这样一种结构
   1. 已经使用可识别的体系结构风格或模式创建
   2. 由展示出良好设计特征的构建构成
   3. 能够以演化的方式实现，从而便于实现和测试
2. 设计应该模块化
3. 设计应该包括数据、体系结构、接口和构件的清晰表示
4. 设计应该导出数据结构，其适用于要实现的类，从可识别的数据模式提取
5. 设计应该导出显示独立功能特征的构件
6. 设计应该导出接口，降低了构件之间以及与外部环境连接的复杂性
7. 设计的导出应根据软件需求分析过程中获取的信息采用可重复的方法进行
8. 应使用能够有效传达其意义的表示法来表达设计
9. 设计基于类的基本构件

有4种适用于构件级设计的基本设计原则

1. 开闭原则（The open-closed principle，OCP）
2. Liskov替换原则（Liskov Substitution Principle，LSP）
3. 依赖倒置原则（Dependency Inversion Principle，DLP）
4. 接口分离原则（Interface Segregation Principle，ISP）
5. 白盒测试、黑盒测试
6. 什么是软件过程？
7. 软件过程中，有哪些共同的，基本的活动？
8. 如何建立过程模型？什么是过程模式？
9. 什么是惯用过程模型？哪些优缺点？\*\*
10. 每一种不同的过程模型都可以用不同的工作流来描述，工作流描述了活动、动作和任务如何按顺序组织。过程模式用来解决软件过程中遇到的共性问题
11. 类瀑布模型和V模型的顺序过程是经典的软件过程模型
12. 增量过程模型采用迭代的方式工作。
13. 演化过程模型认识到大多数软件工程的迭代、递增特性，其设计目的是为了适应变更
14. 开发过程模型未软件团队提供了过程模型中的重叠和并发元素的描述方法
15. 软件工程的敏捷理念强调了4个关键问题：自我组织团队对所开展工作具有控制力的重要性；团队成员之间以及开发参与者与客户之间的交流与合作；对“变更代表机遇“的认识；强调快速软件交付以让客户满意
16. 极限编程（XP）是应用最广泛的敏捷过程，按照计划、设计、编码和测试四个框架组织活动
17. 如何创建需求模型？包含哪些元素？
18. 好的设计包含哪些元素？
19. 如何提高高质量软件部分？
20. 在设计用户接口时使用什么概念、模型及方法？
21. 精化阶段进一步把需求扩展为分析模型--基于场景、类、行为和面向数据流的模型远射的集合
22. 需求建模的目标是创建各种表现形式，用其描述什么是客户需求，建立生成软件设计的基础，一旦软件建立就能定义一组可被确认的需求。
23. 需求模型为系统级表示层和软件设计之间的间隔构造了桥梁
24. 可以使用各种UML建模方法定义类之间的层次、关系、关联、聚合和依赖。P109
25. 行为建模描述了动态行为。行为模型采用基于场景、面向流程、基于类的输入，从而把分析类和系统的优点作为一个整体来描述
26. 设计哦行包含始终不同的元素：体系结构元素、接口设计元素、构件级元素、部署级元素
27. 体系结构设计由4部完成：系统必须表示在相应的环境中、确定一系列的顶层抽象（原型）、设计开始向实现移动、开发体系机构特定示例并在真实世界中验证设计
28. 从3个不同的视角进行构件设计：面向对象的视角、传统的视角、过程视角
29. 高质量软件的一般特性是什么？
30. 什么是软件质量保证？
31. 软件测试要用什么策略？

设计大题

1. E-R图



学会分析多个任务形式任务表

描述表

从图的左上角出房，对图进行描述

1. UML

了解其关系，六条线的用途及其含义（见word）

1. 白盒测试（）和你记得差不多