

大数据

UML占大头，九种xx都要会

简答题30分：软件工程的定义、统一模型各种阶段、各种开发模型有哪些

分析与设计70分：设计思路、写用例/画UML图（常见的9种）

- 用例描述、识别【需求分析-设计-建模】
- 画图元素要对、多写字

计科

一、简答题（30分 5,6个） 分析与设计题（70分）

二、简答题：都是比较大的概念，软件的定义、过程模型及各个阶段、敏捷开发、构建和模块、耦合与内聚

三、分析设计题：比较活，需求分析、识别用例、用例描述、用例叙述、uml设计、建模、画图的元素要对（实线、虚线、形状）、多写字

汇总题型

简答题5、6个共30分

软件的定义、软件统一过程模型及各个阶段、敏捷开发及过程开发模型、构建及模块、相互关联的重要概念、联系的两个概念

分析设计题70分

提出需求、需求分析、识别利益关系者、识别用例、用例设计描述、UML设计、各种建模、活动图、用例图、异常流 分支流

简答题

1. 通用过程模型

通用过程框架定义了5种框架活动——沟通、策划、建模、构建以及部署。

一系列普适性活动——项目跟踪控制、风险管理、质量保证、配置管理、技术评审以及其他活动——贯穿软件过程始终。

2. 软件统一过程模型及各个阶段（非常重要）

统一过程（RUP/UP，Rational Unified Process）是一种以用例驱动、以体系结构为核心、迭代及增量的软件过程模型，由UML方法和工具支持，广泛应用于各类面向对象项目。

1. UP的**起始阶段**包括客户端沟通和策划活动。
2. **细化阶段**包括沟通和通用过程模型的建模活动。
3. **构建阶段**与通用软件过程中的构建活动相同。
4. **转换阶段**包括通用构建活动的后期阶段以及通用部署活动的第一部分。
5. **生产阶段**与通用过程的部署活动一致。

3. 软件的定义（重要）

计算机软件是由专业人员开发并长期维护的工作产品。这些产品包括可以在各种不同容量和不同系统结构的计算机上运行的程序。

软件和硬件具有完全不同的特性：软件不会“磨损”，但是软件退化的确存在。

软件维护要应对变更请求，比硬件维护更为复杂。

4. 什么是软件过程？有哪些常用的过程模型？（重要）

- 过程是事情进行或事物发展所经过的顺序
- 当开发产品或构件系统时，遵循一系列可预测的步骤（即路线图）是非常重要的，它有助于及时交付高质量的产品
- 软件开发中所遵循的路线图就称为“软件过程”

常用的过程模型有：瀑布模型、原型模型、螺旋模型、统一过程模型

5. 软件工程的定义（重要）

软件工程包括过程、一系列方法（实践）和大量工具，专业人员借由这些来构建高质量的计算机软件。

软件工程：将系统化的、规范的、可量化的方法应用于软件的开发、运行和维护，即将工程化方法应用于软件。

6. 软件的生命周期

定义问题、软件开发和软件维护

7. 什么是敏捷开发

是一种软件开发方法论，可以应对客户快速变更的需求。它强调以人为核心，采用迭代的方式，循序渐进的开发软件。

基于敏捷原则进行的软件开发过程，视为敏捷过程。所谓“基于”，是指充分考虑，而不是全部包含。

8. 需求工程是指致力于不断理解需求的大量任务和技术。

需求工程包括七项任务：起始，获取，细化，协商，规格说明，确认和管理。

9. 设计建模原则

1. 设计应可追溯到需求模型。
2. 始终考虑要构建系统的体系结构。
3. 数据设计与处理功能设计同等重要。
4. 接口（内部和外部）的设计必须谨慎。
5. 用户界面设计应适应最终用户的需求。
6. 构件级设计应在功能上独立。
7. 构件应彼此松耦合，并应与外部环境松耦合。
8. 设计表示（模型）应易于理解。
9. 设计应迭代式开发。
10. 设计模型的创建并不排除采用敏捷方法的可能性。

10. 什么是模块化？

模块化是软件的单一属性，它使程序能被智能化地管理。

11. 为什么需要模块化？

- 模块化是为了使一个复杂的大型程序能被人的智力所管理，软件应该具备的惟一属性。
- 如果一个大型程序仅由一个模块组成，它将很难被人所理解。

12. 什么是构件？（重要）

通常来讲，构件是计算机软件中的一个模块化的构造块。

再正式一点，OMG统一建模语言规范是这样定义构件的：系统中模块化的、可部署的和可替换的部件，该部件封装了实现并对外提供一组接口。

13. 什么是体系结构？

体系结构并非可运行的软件。确切地说，它是一种表达，使能够：

- (1) 在满足既定的需求方面下，分析设计有效性；
- (2) 在设计变更相对容易的阶段，考虑体系结构可能的选择方案；
- (3) 降低与软件构造相关的风险

14. 内聚性

传统观点：构件的专一性

面向对象的观点：内聚性意味着构件或类只封装那些相互关联密切，以及与构件或类自身有密切关系的属性和操作。

内聚分类：功能内聚、分层内聚、通信内聚、顺序内聚、过程内聚、时间内聚、偶然内聚

15. 耦合性

耦合性是程序结构中各个模块之间相互关联的度量它取决于各个模块之间接口的复杂程度、调用模块的方式以及那些信息通过接口。

传统观点：一个组件连接到其他组件和外部世界的程度。

面向对象的观点：耦合是类之间彼此联系程序的一种定性度量。

耦合是必然存在的。

耦合程度：内容耦合、共用耦合、控制耦合、标记耦合、数据耦合、例程调用耦合、类型使用耦合、包含或者导入耦合、外部耦合。

16. 黄金规则（非常重要）

Theo Mandel关于界面设计提出的三条黄金规则

1. 把控制权交给用户
2. 减轻用户的记忆负担
3. 保持界面一致

用例描述

用例描述一般包括：简要描述（说明）、前置（前提）条件、基本事件流、其他事件流、异常事件流、后置（事后）条件等等。下面说说各个部分的意思：

简要描述：对用例的角色、目的的简要描述；

前置条件：执行用例之前系统必须要处于的状态，或者要满足的条件；

基本事件流：描述该用例的基本流程，指每个流程都“正常”运作时所发生的事情，没有任何备选流和异常流，而只有最有可能发生的事件流；

其他事件流：表示这个行为或流程是可选的或备选的，并不是总要总要执行它们；

异常事件流：表示发生了某些非正常的事情所要执行的流程；

后置条件：用例一旦执行后系统所处的状态；

用例名称：网站公告发布
用例标识号：202
参与者：负责人
简要说明：负责人用来填写和修改家教网站首页的公告，公告最终显示在家教网站的首页上。
前置条件：负责人已经登陆家教网站管理系统
基本事件流： 1. 负责人鼠标点击“修改公告”按钮 2. 系统出现一个文本框，显示着原来的公告内容 3. 负责人可以在文本框上修改公告，也可以完全删除，重新写新的公告 4. 负责人编辑完文本框，按“提交”按钮，首页公告就被修改 5. 用例终止
其他事件流 A1： 在按“提交”按钮之前，负责人随时可以按“返回”按钮，文本框的任何修改内容都不会影响网站首页的公告
异常事件流： 1. 提示错误信息，负责人确认 2. 返回到管理系统主页面
后置条件：网站首页的公告信息被修改
注释：无