**最后老师说的思考题**

1. **软件工程热点话题，谈谈自己的认识**
2. **5G的到来对软件工程的影响**
3. **计算机语言的发展从低级语言到高级语言，体现了什么特点：（**老师说首先知道这三句话，然后可以展开描述三句话的意思，最后要知道这些特征的出现与软件工程发展的关系**）**

a.计算机语言对硬件资源的依赖性越来越低。

b.所描述的数据结构越来越复杂

c.所描述的程序特征越来越抽象（抽象的层次和级别是针对cpu的）

**第一部分：**

软件工程是一门方法学：方法、过程、工具。其中包括面向对象方法学，面向过程方法学，基于构建方法学，维护驱动方法学（成本高），面向数据结构方法学。

**第二部分（需求分析部分）：**

1.两种不同方法学比较（分别从方法、过程、工具展开），方法：面向对象和面向过程。

考点：比较分析面向对象和面向过程两种不同方法的本质区别：

a.面向过程程序设计方法采用函数（或过程）来描述对数据的操作，但又将函数与其操作分离开来；面向对象程序设计方法将数据和对数据的操作封装在一起，作为一个整体来处理。

b.面向过程方法以功能为中心来设计功能模块，难于维护；而面向对象程序设计方法以数据为中心来描述系统，数据相对于功能而言具有较强的稳定性，因此更易于维护。

c.面向过程程序的控制流程有程序中预定顺序来决定；面向对象程序的控制流程由运行时各种事件的实际发生来触发，而不再由顺序来决定，更符合实际需要。

d.面向对象程序设计方法可以利用框架产品进行编程。

面向过程性能高，因为面向对象需要实例化类，开销大，缺点是不易维护；

面向对象易维护、易扩展、易复用，缺点是性能比面向过程低。

**过程：**

**什么是瀑布模型，有何特性，为什么有僵化的瀑布模型之说？**

(1) 瀑布模型：它提出了一个系统的、顺序的软件开发方法，从用户需求规格说明开始，通过计划、建模、构建和部署的过程，最终提供一个完整的软件并提供持续的技术支持。

(2) 特性：① 阶段间具有顺序性和依赖性；② 推迟实现的观点，划分逻辑设计与物理设计，尽可能推迟程序的物理实现；③ 质量保证的观点，每个阶段必须完成规定的文档，对其中问题通过复审及早发现，及早解决。

(3) 僵化：对变更的无奈。① 有些类型的系统需求是模糊的；② 项目参与者之间存在通信鸿沟；③ 预先定义的需求可能是过时的。

螺旋模型：有利于已有如案件的重用，有助于把软件质量作为软件开发的一个重要目标，减少了过多测试或测试不足所带来的风险，软件维护与软件开发没有本质区别。

**工具**

宏观：平台/支持软件开发流程

微观：UML图，协助图，顺序图

**功能模型：用例图如何画？**

面向过程工具：数据流图->数据字典->软件结构图

面向对象：类图

设计模型：

为什么需求分析是通向设计的桥梁，这个该如何理解？

软件工程致力于尽可能缩短分析与设计之间的语义断层，使得分析与设计之间进行平滑过渡。需求分析对最初的需求进行细化，并建立模型教会系统做什么和怎么做。形式化方法建模使用数学语言描述需求，，面向对象方法建模在一定程度上实现了分析与设计的无缝连接。

描述用户场景、功能活动、类及类之间的关系、系统和类行为、数据流等（形式化建模、面向对象建模等），为设计提供信息、功能和行为的表示，以便设计出软件的结构、接口和构件（模块）。

UML建模，考点

分析一个题目

经典模型举例（面向对象：1.喷泉模型；面向过程：2.瀑布模型）

**比较分析两种主流的方法学（面向数据流的结构化分析方法和面向对象的分析方法）。**

(1) 方法：面向数据流是一种传统的结构化分析方法，是典型传统的瀑布模型；面向对象是一种把面向对象的思想应用于软件开发过程中，指导开发活动的系统方法，是喷泉模型。

(2) 过程：面向数据流（从哪里来、怎样流、要流去哪里）把一个复杂问题的求解过程分阶段进行，而且这种分解是自顶向下，逐步抽象构建细化数据流图，使得每个阶段处理的问题都控制在人们容易理解和处理的范围内；面向对象是一个重复迭代的过程，生命周期长，稳定性高，其步骤为：沟通、确定类、定义类的层次结构、表现对象与对象的关系、为对象行为建模。

(3) 工具：面向数据流方法分析与描述的工具为：数据流图、数据字典、判定表和判定树等。面向对象方法分析与描述的工具为用例图、类图、状态图等。

方法上：1.以数据为中心，为了数据去封装功能2.为了实现功能去组织数据，以功能为中心，自顶向下。

过程上：1.阶段无缝连接，迭代 2.阶段依赖性，迭代性，在两个过程中加入review，推迟实现。

工具上：1.类图，状态转换图 UML（顺序图，活动图，协作图）2.数据流图、用例图、（数据字典，判定表，判定树）、软件结构图。

再分析两种方法分别适用什么样的大型软件。

2．分析模型到设计模型的映射关系

理解分析分析是通往设计的桥梁？

1. 分析模型
2. 设计模型
3. 关系

3.给出一段关系，进行UML建模

**第三部分（软件设计部分）**

1. **比较C/S，B/S架构的优缺点：**

C/S优点：

(1) C/S架构的界面和操作可以很丰富。

(2) 安全性能可以很容易保证，实现多层认证也不难。

(3) 由于只有一层交互，因此响应速度较快。

C/S缺点：

(1) 适用面窄，通常用于局域网中。

(2) 用户群固定。由于程序需要安装才可使用，因此不适合面向一些不可知的用户。

(3) 维护成本高，发生一次升级，则所有客户端的程序都需要改变。

B/S优点：

(1) 客户端无需安装，有Web浏览器即可。

(2) B/S架构可以直接放在广域网上，通过一定的权限控制实现多客户访问的目的，交互性较强。

(3) B/S架构无需升级多个客户端，升级服务器即可。

B/S缺点：

(1) 在跨浏览器上，B/S架构不尽如人意。

(2) 表现要达到C/S程序的程度需要花费不少精力。

(3) 在速度和安全性上需要花费巨大的设计成本，这是B/S架构的最大问题。

(4) 客户端服务器端的交互是请求-响应模式，通常需要刷新页面，这并不是客户乐意看到的。（在Ajax风行后此问题得到了一定程度的缓解）

体系架构6~7个题目中抽一个

经典的体系结构风格：

数据流体系结构：批处理序列；管道/过滤器。

调用和返回体系结构：主程序/子程序；面向对象风格；层次结构

独立构件风格：进程通讯；事件系统

虚拟机风格：解释器，基于规则的系统

仓库风格：数据库系统；超文本系统；黑板系统

C/S风格

三层C/S风格

B/S风格

2.模块粒度逐渐变大：函数->类->service-> Agent

Saas

SOA

模块自主性/智能性/复制性/繁殖性 逐渐提高，对软件工程体系架构的影响。

**第四部分（软件测试）**

软件测试，一个题目

1.考点：基本概念，软件测试阶段划分（模块->单元测试->集成测试->确认测试->系统设计->验收测试）

问题定义-可行性报告，代码也是一种文档，维护文档，用户手册文档

考点：1.每个测试阶段对应哪些文档？和哪阶段有关

Eg:单元测试与详细设计方法有关，与代码有关，系统测试与哪些有关

2.黑/白盒测试：画程序流程图，程序控制图

1. 计算出软件环境复杂性（参考PPT上的例题）

考点2：白盒测试：基本路径测试，求一组独立路径；针对某组设计测试用例（见PPT，基本路径测试）

**考点3：黑盒测试：考因果图如何转换为判定图，测试用例的含义，要描述出来（见PPT贩卖机的例题）**

**什么是黑盒测试，有哪些测试方法？**

(1) 黑盒测试：把测试对象看做一个黑盒子，测试人员完全不考虑程序内部的逻辑结构和内部特性，只依据程序的需求规格说明书，检查程序的功能是否符合它的功能说明。

(2) 黑盒测试方法：等价类划分，边界值分析，错误猜测法，因果图法

例题如下：

有一个处理单价为5角钱的饮料的自动售货机软件，其规格说明如下：

若投入5角钱或1元钱的硬币，押下〖橙汁〗或〖啤酒〗的按钮，则相应的饮料就送出来。若售货机没有零钱找，则一个显示〖零钱找完〗的红灯亮，这时在投入1元硬币并押下按钮后，饮料不送出来而且1元硬币也退出来；若有零钱找，则显示〖零钱找完〗的红灯灭，在送出饮料的同时退还5角硬币。”

**(1) 分析这一段说明，列出原因和结果  
原因: 1. 售货机有零钱找**

**2. 投入1元硬币**

**3. 投入5角硬币**

**4. 押下橙汁按钮**

**5. 押下啤酒按钮**

**建立中间结点，表示处理中间状态  
11. 投入1元硬币且押下饮料按钮  
12. 押下〖橙汁〗或〖啤酒〗的按钮  
13. 应当找5角零钱并且售货机有零钱找  
14. 钱已付清**

**结果： 21. 售货机〖零钱找完〗灯亮**

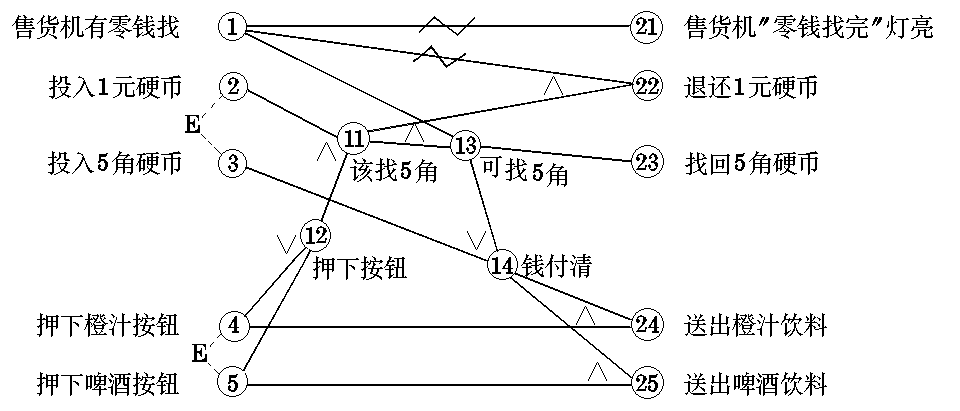
**22. 退还1元硬币**

**23. 退还5角硬币**

**24. 送出橙汁饮料**

**25. 送出啤酒饮料  
(2) 画出因果图。所有原因结点列在左边，所有结果结点列在右边。**

**(3) 由于 2 与 3 ，4 与 5 不能同时发生，分别加上约束条件E。  
(4) 因果图**



（5）转换成判定表

