**考试范围**

**选择题**

单选 10题 每题2分  
瀑布模型，增量模型，快速原型模型

**判断题**

10题 每题2分  
不是55开

**简答题**

4题 每题5分（作业、练习、测验，不用过分展开，主要是要点）

* 软件危机是什么，产生的原因与解决措施
* 面向对象建模的三个子模型是什么，最核心最基本的是哪个，对复杂问题的模型是由哪五个层次组成
* 类聚和耦合的级别，具体从高到低排序
* 需求分析工作结果有哪两个，分析建模，需求说明书文档，四个方面验证
* 可行性分析的主要研究内容，五个方面
* 面向对象方法的主要优点，5点，标题即可，p205
* 软件文档的分类（两类），分别描述什么，满足什么要求，课本上有
* 过程设计的工具可以分几类，课本中学过的分别是哪类

**应用题**

大概率是开发模型的选择：

* 风险——螺旋
* 准确获取用户需求——快速原型
* 时间需求——增量
* 最快——瀑布

**分析或设计DFD图**

DFD图，符号就四种，见p41

所有的图符都要有名字，数据流，原点，加工，存储都要有名字，加工至少要一进一出，父子图的平衡

**黑白盒设计测试用例**

白盒：语句覆盖，判定覆盖，条件覆盖，判定/条件覆盖。要写预期的路径和结果！

黑盒：等价类划分，边界值分析，看例子与作业，先划分等价类表，再编号，防止遗漏，尽可能完成所有等价类覆盖。每个无效等价类，单独设计测试类。

测试样例：测试数据+预期结果，覆盖了哪个类最好写一下。边界值要作为测试用例，对于测试样例，需要取最接近边界值的测试样例。

**作图题**

盒图、PAD图、判定树

**考题分析**

**第一章**

1. 软件危机的表现：

对软件的开发成本和进度估计不准

用户对已完成软件不满意

软件的质量靠不住

软件常常是不可维护的

软件没有适当的文档资料

软件成本不断上升

软件开发生产率低

1. 什么是软件危机？为什么会产生？怎样消除？

**软件危机是指在计算机软件的开发和维护过程中所遇到的一系列严重问题**。这些问题绝不仅仅是不能正常运行的软件才具有的，实际上，几乎所有软件都不同程度地存在这些问题。

**产生软件危机的原因 ：一、与软件本身的特点有关；二、和软件开发与维护的方法不正确有关。**

我们可以通过了解产生软件危机的原因，澄清错误认识，建立起关于软件开发和维护的正确概念，及采取一系列综合措施来达到消除的目的。**这些措施包括技术措施（方法和工具），组织管理措施，以及运用软件工程理论和实践。**

1. **软件工程方法学包括三个要素：方法、工具和过程**
2. 最常用的软件工程方法学，分别是传统方法学和面向对象方法学
3. **软件生命周期：问题定义、可行性研究、需求分析、总体设计、详细设计、编码和单元测试、综合测试、软件维护**
4. **软件过程模型：瀑布、快速原型、增量、**
5. **螺旋**

**习题1**：假定你被任命为一家软件公司的项目负责人，你的工作是管理该公司已被广泛应用的字处理软件的新版本的开发。由于市场竞争激烈，公司规定了严格的完成期限并且已经对外公布。你打算采取哪种软件生命周期模型？为什么？

**答：**采用增量模型。理由对这个项目的一个重要要求是，严格按照已对外公布了的日期完成产品开发工作。使用增量模型开发软件时可并行完成开发工作，能在较短时间内向用户提交可完成部分的工作的产品，以后根据需要再逐步增加产品功能，因此能加快开发进度。

**习题2：**假定要求你开发一个软件，该软件的功能是把读入的浮点数开平方，所得到的结果应该精确到小数点后4位。一旦实现并测试完之后，该产品将被抛弃。你打算选用哪种软件生命周期模型？请说明你给出选择的理由。

**答：**采用瀑布模型。理由：对这个软件的需求很明确，实现开平方根的算法也很成熟，因此，既无须通过原型来分析需求来验证设计方案。此外，一旦实现并测试完成之后，该产品将被抛弃，因此也无须使用有助于提高软件可维护性的增量模型或螺旋模型来开发该产品。

**习题3：**假定你要为一家生产和销售长筒靴的公司开发一个软件，使用此软件来监控该公司是存货， 并跟踪从购买橡胶开始，到生产长筒靴、发货给各个连锁店，直至卖给顾客的全部过程，以保证生产、销售过程的各个环节供需平衡，既不会停工待料现象，也不会有供不应求现象。你为这个项目选择生命周期模型时使用什么准则为合适？

**答：**选用快速原型模型。

理由：

1. 软件系统使用人员繁杂，需求确定困难；
2. 软件应用环节多，功能变化多，初期需求变化多，需要在修改原型的过程中明确需求。
3. 软件过程四个基本活动：

·P (Plan) : 软件规格说明。规定软件的功能及其运行的限制；

·D (Do) : 软件开发。产生满足规格说明的软件；

·C (Check) : 软件确认。确认软件能够完成客户提出的要求；

·A (Action) : 软件演进。为满足客户的变更要求，软件必须在使用的过程中演进。

**第二章**

1. 可行性研究相关知识点

**在软件开发的早期阶段为什么要进行可行性研究?应该从哪些方面研究目标系统的可行性?**

可行性研究的**目的**是用最小的代价在尽可能短的时间内确定问题是否能解决。如果问题没有可行的解，那么任何花费都是无谓的浪费，因此在软件开发的早期阶段进行可行性研究是很有必要的。

可行性研究主要**技术可行性、经济可行性以及操作的可行性3个方面**加以研究，必要的时候还应该从法律和社会效益等更广泛方面研究每种解法的可行性。

1. 数据字典是描述数据的信息的集合，是对系统中使用的所有数据元素的定义的集合。

**第三章**

1. 结构化分析的准则四条

理解并描述问题的信息域，根据这条建立数据模型。

定义软件应完成的问题，这条准则要求建立功能模型

描述作为外部事件结果的软件行为，这条准则要求建立行为模型

对描述的信息、功能、行为的模型进行分解，用层次的方式展示细节

1. 需求获取面临的挑战 3点
2. 分析建模的三个模型：数据模型、功能模型、行为模型
3. ER图，状态转换图（选择判断）
4. 验证软件需求：一致性、完整性、现实性、有效性

**通过需求分析，除了创建分析模型之外，还需要完成什么作为这一阶段的工作结果？通常应从哪些方面去验证复审？**

软件需求规格说明书。

通常应从一致性、完整性、现实性和有效性4个方面去验证复审。

**第五章**

1. **耦合度（从低到高）：非直接耦合，数据耦合，标志耦合；控制耦合；外部耦合，公共耦合；内容耦合**
2. **内聚性（从低到高）：偶然内聚、逻辑内聚、时间内聚、过程内聚、通信内聚、顺序内聚、功能内聚**
3. 常用的软件结构设计**启发规则**：改进软件结构提高模块独立性；模块规模应该适中；**深度、宽度、扇出和扇入都应适当；模块的作用域应该在控制域之内；**力争降低模块接口的复杂程度；设计单入口单出口的模块；模块功能应该可以预测

**第六章**

1. **盒图、PAD图、判定树的基本控制结构（顺序、选择、循环）**
2. **环形复杂度计算**
3. 人机界面设计的设计问题：系统响应时间、用户帮助设施、出错信息处理、命令交互。
4. 过程设计工具：图形工具、表格工具、语言工具
5. PDL 是用正文形式表示数据和处理过程的设计工具。
6. Jackson 方法是一种典型的面向数据结构的分析与设计方法。
7. 程序复杂性主要指模块内程序的复杂性。
8. McCabe 度量法，又称环路复杂性度量，是一种基于程序控制流的复杂性度量方法。
9. Halstead 方法，工具程序中运算符和操作数的总数来度量程序的复杂程度

**第七章**

1. 程序设计语言分为以下几类：机器语言、汇编语言、高级语言
2. **编码风格包括：程序内部的文档、数据说明、语句结构、输入／输出方法和效率几个部分。力图从编码原则的角度提高程序的可读性，改善程序质量。**

**例：**

1. 程序段二有注释，注释起到了提示的作用，有利于程序阅读者对程序段全貌的理解;
2. 程序段二采用了缩进格式，清晰的显示了程序中逻辑判断的结构，即视觉组织好；
3. 程序段二中，变量选取了含义鲜明的名字，对于帮助阅读者理解程序很重要。
4. **软件测试基础：**

**目标：发现程序中的错误**

**准则：所有测试追溯到用户需求；在测试前制定出测试计划；80%的错误是由20%的模块引起的；从小规模到大规模；不可能进行穷举；有独立的第三方进行测试**

**方法：黑盒白盒**

**步骤：模块测试、子系统测试、系统测试、验收测试、平行运行**

1. 单元测试、集成测试、确认测试
2. 白盒逻辑覆盖，黑盒等价划分和边界值分析
3. 调试过程和途径

**第八章**

1. **软件维护的定义：在软件产品发布后，为了改正错误或满足新的需求，适应新环境对软件产品进行的修改就是所谓的软件维护**。
2. **软件维护的类型有：改正性维护、适应性维护、完善性维护、预防性维护，其中完善性维护占比最大。**
3. **软件文档，分类：用户文档、系统文档**

**用户文档：描述系统功能和使用方法**

**系统文档：描述系统分析、设计、实现和测试等方面的内容**

**应满足的要求：**

(1)必须描述如何使用这个系统，没有这种描述即使是最简单的系统也无法使用；

(2)必须描述怎样安装和管理这个系统；

(3)必须描述系统需求和设计；

(4)必须描述系统的实现和测试，以便使系统成为可维护的。

**第九-十二章**

1. 面向对象方法学的要点：

客观世界是由各种对象构成的

所有对象划分成各种对象类

按照子类和父类的关系，可以把若干个对象组成一个层次结构的系统

对象彼此之间能通过传递消息互相联系

1. **面向对象方法学的优点：与人类习惯的思维方法一致、稳定性好、可重用性好、较易开发大型软件产品、可维护性高**
2. **面向对象的基本概念：对象、类、封装、继承**
3. **面向对象分析建模的三个模型及关系：对象模型、动态模型和功能模型，关系p228 （8条太多了）**
4. **主要的面向对象方法：coad、booch、Jacobson、OMT、UML**

**综合题**

**软件过程模型**

**瀑布模型**

* 阶段间具有顺序性和依赖性。
* 推迟实现的观点。
* 质量保证观点。（每个阶段必须完成规定的文档；每个阶段结束前完成文档审查，及早改正错误。）

优势是规范的、文档驱动的方法；这种模型的问题是，最终交付的产品可能不是用户真正需要的。

**快速原型模型（原型模型）**

一般用于最终系统的早期用户评价，开发工期短，质量有保证。

主要特点：

* 使用这种软件过程开发出的软件产品通常能满足用户的真实的需求
* 软件产品的开发过程基本上是线性顺序过程

**增量模型**

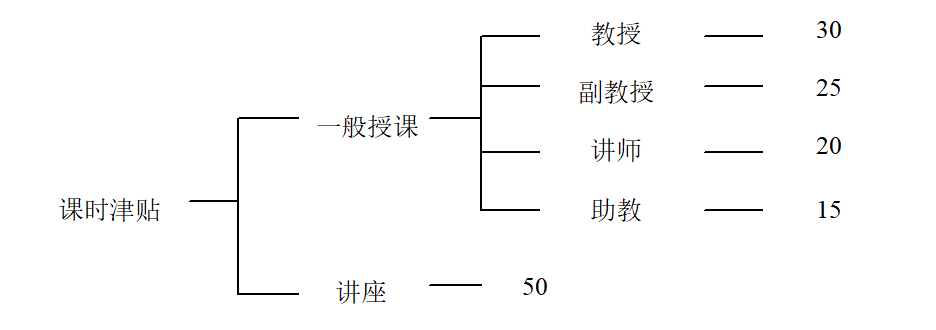
* 能在较短时间内向用户提交可完成部分的工作的产品；
* 逐步增加产品功能，从而使用户有较充裕的时间学习和适应新产品，减少一个全新的软件给客户带来的冲击。

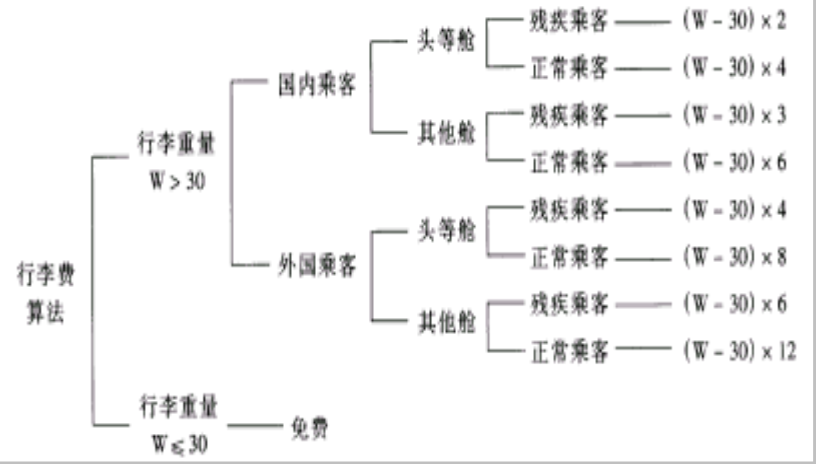
增量模型具有能在软件开发的早期阶段使投资获得明显回报和易于维护的优点，开发出的软件具有较好的可扩充性。但是，要求软件工程师具有较高的技术水平，能够设计出开放的软件体系结构。

**快速原型并不快，需要快的应该选增量模型，瀑布模型比快速原型都快**

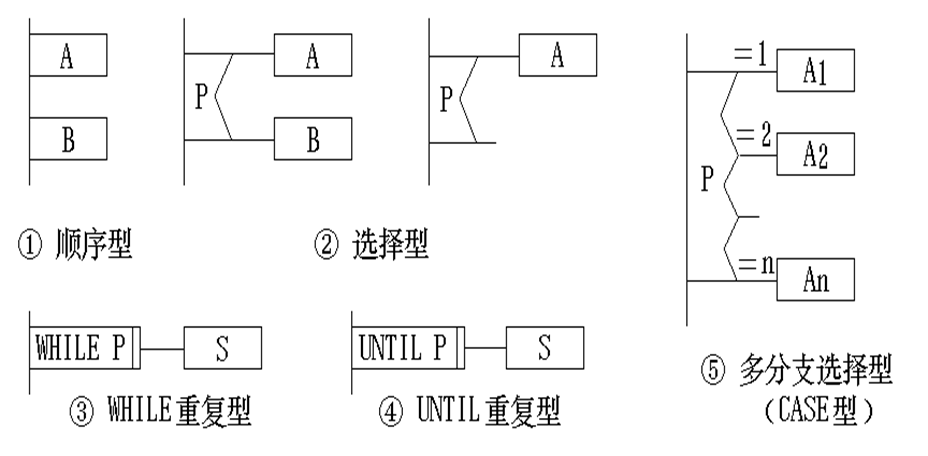
**判定树**

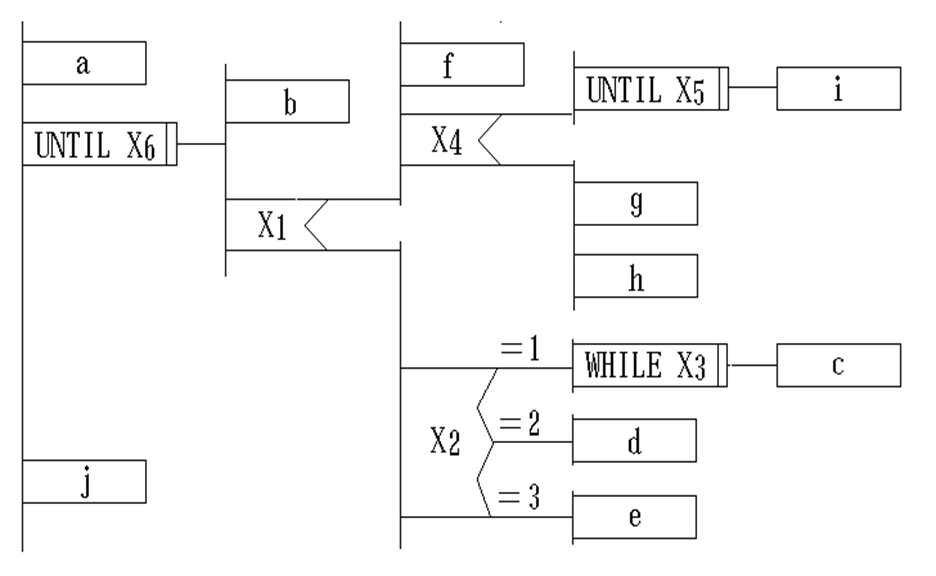
**判定树是判定表的变种**，也能清晰地表示复杂的条件组合与应做的动作之间的对应关系。其优点在于它的形式简单，易于掌握和使用，是一种比较常用的系统分析和设计的工具。

[](http://www.leonlist.top/images/%E8%BD%AF%E4%BB%B6%E5%B7%A5%E7%A8%8B%E5%9F%BA%E7%A1%80%E5%A4%8D%E4%B9%A0/image-20210107143158315.png)

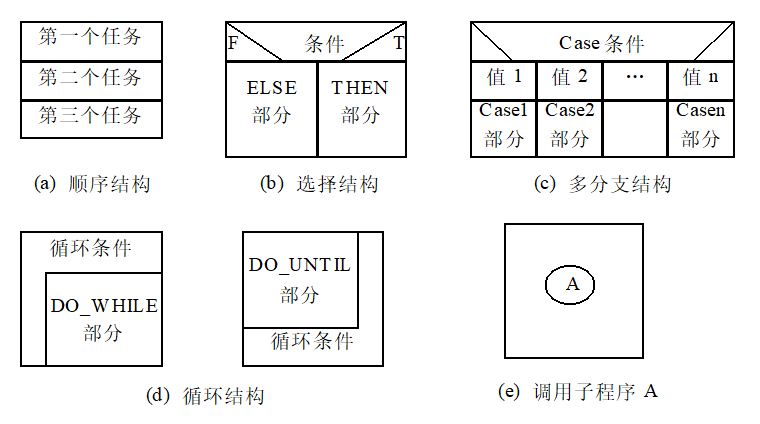
[](http://www.leonlist.top/images/%E8%BD%AF%E4%BB%B6%E5%B7%A5%E7%A8%8B%E5%9F%BA%E7%A1%80%E5%A4%8D%E4%B9%A0/image-20210107143208914.png)

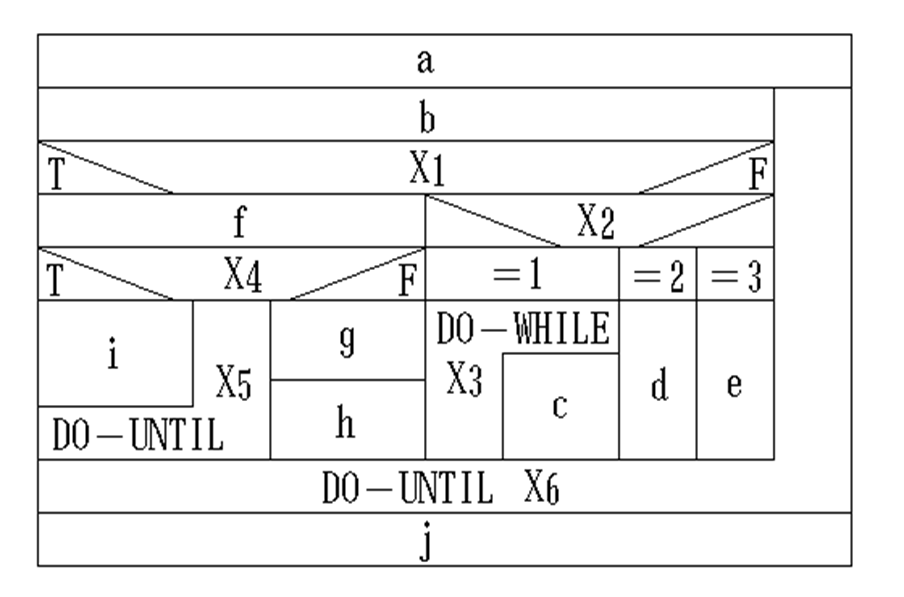
**PAD图**

[](http://www.leonlist.top/images/%E8%BD%AF%E4%BB%B6%E5%B7%A5%E7%A8%8B%E5%9F%BA%E7%A1%80%E5%A4%8D%E4%B9%A0/image-20210107143257142.png)

[](http://www.leonlist.top/images/%E8%BD%AF%E4%BB%B6%E5%B7%A5%E7%A8%8B%E5%9F%BA%E7%A1%80%E5%A4%8D%E4%B9%A0/image-20210107143310386.png)

**盒图（N-S图）**

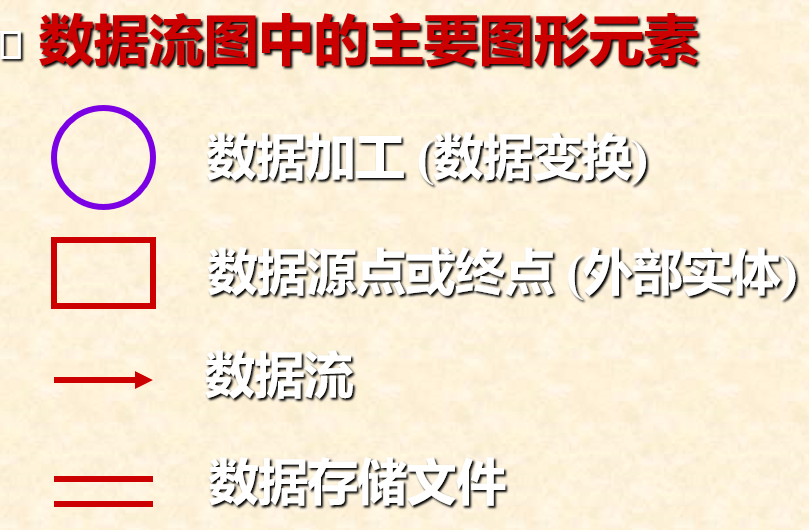
[](http://www.leonlist.top/images/%E8%BD%AF%E4%BB%B6%E5%B7%A5%E7%A8%8B%E5%9F%BA%E7%A1%80%E5%A4%8D%E4%B9%A0/image-20210107143405206.png)

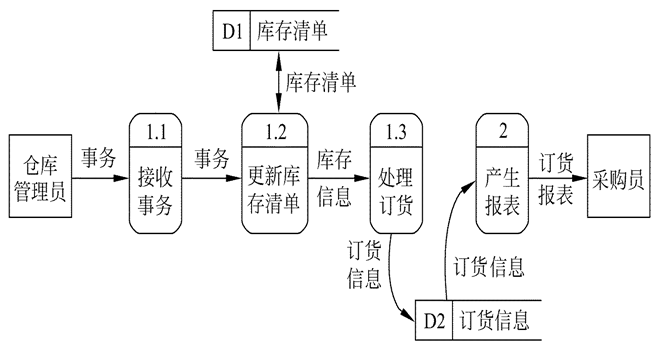
[](http://www.leonlist.top/images/%E8%BD%AF%E4%BB%B6%E5%B7%A5%E7%A8%8B%E5%9F%BA%E7%A1%80%E5%A4%8D%E4%B9%A0/image-20210107143422404.png)

**DFD图（数据流图）**

DFD图，不要细化，知道功能就行，符号就四种，见书本p41

所有的图符都要有名字，数据流，原点，加工，存储都要有名字

[](http://www.leonlist.top/images/%E8%BD%AF%E4%BB%B6%E5%B7%A5%E7%A8%8B%E5%9F%BA%E7%A1%80%E5%A4%8D%E4%B9%A0/image-20210107143747777.png)

[](http://www.leonlist.top/images/%E8%BD%AF%E4%BB%B6%E5%B7%A5%E7%A8%8B%E5%9F%BA%E7%A1%80%E5%A4%8D%E4%B9%A0/image-20210107143828443.png)

**设计测试用例**

详见书本P162

**白盒**

1. **语句覆盖**

使程序中每个语句至少执行一次，语句覆盖是最弱的逻辑覆盖

1. **判定覆盖**

每个语句执行一次，且每个判定的真假分支都至少执行一次。判定覆盖仍是弱的逻辑覆盖。

1. **条件覆盖**

每个语句执行一次，且使每个判定的每个条件的可能取值至少执行一次。

条件覆盖不一定包含判定覆盖，判定覆盖也不一定包含条件覆盖

1. **判定/条件覆盖**

选取足够多的测试用例，使判断中的每个条件的所有可能取值至少执行一次，同时每个判断本身的所有可能判断结果至少执行一次。

能同时满足判定/条件两种覆盖标准取值

（不能多写，最好两组。要写预期的路径和结果）

**黑盒**

1. 等价类划分
2. 边界值分析

看例子与作业，先划分等价类表，再编号，防止遗漏，尽可能完成所有等价类覆盖。每个无效等价类，单独设计测试类。  
测试样例：测试数据+预期结果，覆盖了哪个类最好写一下。边界值要作为测试用例，对于测试样例，需要取最接近边界值的测试样例。

详见书本P172

**总结**

* 简答题和综合题一定要多背多记
* 作业，测试要多看
* 可以多刷题
* 选择、判断随缘：)

**上学期软件班期末考**

部分题型

**选择题**

1. 小组成员完全平等，享有充分民主，通过协商做出技术决策。这种组织方式是：

**民主制程序员组**

传统程序员组

主程序员组

现代程序员组

1. 下列属于面向对象开发方法的是：

Jackson

**Booch**

HIPO

warnier

1. 软件开发技术一般不包括：

软件开发方法学

软件工程环境

软件工具

**软件质量度量**

1. 软件测试方法中的（ ）属于静态测试方法

黑盒法  
逻辑覆盖  
错误推测  
**人工检测**

1. Jackson方法是一种面向（ ）的设计方法

数据流

**数据结构**

对象

控制

1. 软件需求规格说明的内容不包括（ ）

**算法的详细描述**

软件主要功能

软件的性能

用户界面及运行环境

1. 软件生命周期中所花费用最多的阶段是：

软件编译

软件测试

**软件维护**

详细设计

1. 面向对象方法学的主要优点不包括：

稳定性好

可重用性好

与人类习惯的思维方法一致

**运行效率**

1. 经济可行性研究的范围包括：

资源有效性

管理制度

**效益分析**

开发风险

**多选题**

1. 画分层DFD图的基本原则有：

**数据守恒原则**  
分解的可靠性原则  
**子、父图平衡的原则**  
**数据流封闭的原则**

1. 软件测试主要包括（ ）方面的差错和纠错

**软件计划**

**软件设计**

**软件编码**

**软件价格**

1. 模块独立性准则由以下定性指标衡量：

**耦合**

信息屏蔽

**内聚**

分解度

1. 以下属于软件测试的原则有：

**程序最好由编写该程序的程序员自己来测试**

**所有测试标准都是建立在用户需求之上**

**注意测试中的群集现象（2-8原则）**

测试是证明程序有错的一个过程

1. 可重用的软构件应具有的特点包括：

**具有高度可塑性**

功能简单

**接口清晰、简明、可靠**

**模块独立性强**

1. 下面哪些（ ）可以作为模块

**子程序**

**宏**

**函数**

**过程**

1. 在人机界面的设计过程中，需要考虑下列（ ）问题

**系统响应时间**

输入输出数据

**出错信息处理**

**用户帮助**

1. 软件危机的主要表现是：

软件成本太高  
**软件产品的质量低劣**  
软件开发人员明显不足  
**软件生产率低下**

1. 关于数据字典正确的描述是（ ）

**数据字典最重要的用途是作为分析阶段的工具**

**实现数据字典三种常见的方法有全人工过程、全自动过程、混合过程**

**数据字典应该容易更新和修改**

**可以采用卡片形式书写数据字典**

**判断题**

1. 软件项目管理先于任何技术活动之前开始，并且贯穿于软件的整个生命周期之中。

**对**

1. HIPO法既是需求分析方法，又是软件设计方法

**对**

1. 在一个设计得很好的系统中，模块的控制域是模块的作用域子集。

**错**，模块的作用域应该在控制域之内，如果模块的作用域不在控制域之内，则会增加模块间数据的传递量，使模块间出现控制耦合。（详见书本P100）

1. 在发现错误后，则应按照一定的技术去纠正它，纠错的关键是错误定位。

**对**

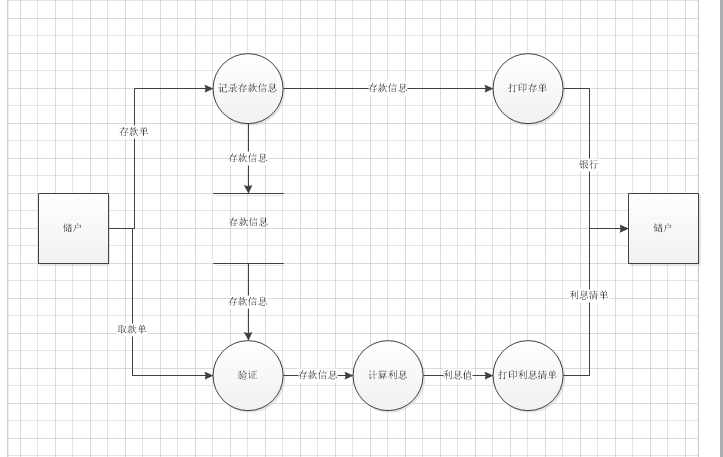
1. 面向对象的开发方法包括面向对象的分析、面向对象的设计和面向对象的实现。

**对**

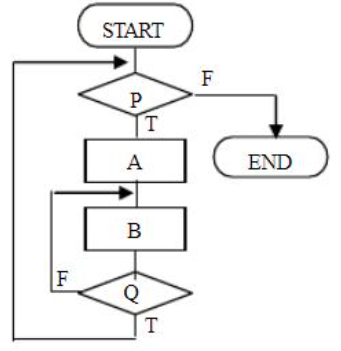
**综合题**

1. **为方便储户，某银行开发计算机存储系统。储户填写的存款单或取款单由业务员键入系统，如果是存款，系统记录存款人姓名、住址、存款类型、存款日期、利率等信息，并印出存款单给储户；如果是取款，系统计算利息并印出利息清单给储户。请画出此系统的数据流图。**

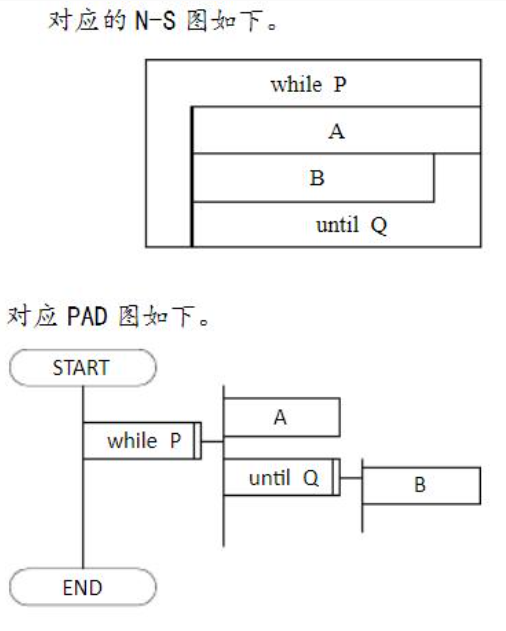
答：

[](http://www.leonlist.top/images/%E8%BD%AF%E4%BB%B6%E5%B7%A5%E7%A8%8B%E5%9F%BA%E7%A1%80%E5%A4%8D%E4%B9%A0/905692-20160418221059460-919259265.png)

1. **在程序流程图,试分别用N-S图和PAD表示之,并计算它们的McCabe复杂性度量。**

[](http://www.leonlist.top/images/%E8%BD%AF%E4%BB%B6%E5%B7%A5%E7%A8%8B%E5%9F%BA%E7%A1%80%E5%A4%8D%E4%B9%A0/image-20210106143857048.png)

答：

[](http://www.leonlist.top/images/%E8%BD%AF%E4%BB%B6%E5%B7%A5%E7%A8%8B%E5%9F%BA%E7%A1%80%E5%A4%8D%E4%B9%A0/image-20210106143919637.png)

McCabe 复杂性度量为 3

1. **假设自己被指派作为一个大型软件产品公司的项目负责人，工作是管理该公司已被广泛应用的字处理软件的新版本开发。由于市场竞争激烈，公司规定了严格的完成期限并且对外公布了。应选择哪种项目组结构？为什么？打算采用哪种软件过程模型？为什么？**
   1. 应该选择现代程序员组，因为小组成员都能对发现程序错误持积极、主动的态度。能更好的适应竞争。
   2. 对这个项目的一个重要要求是，严格按照已对外公布了的日期完成产品开发工作，因此，选择生命周期模型时应该着重考虑哪种模型有助于加快产品开发的进度。使用增量模型开发软件时可以并行完成开发工作，因此能够加快开发进度。这个项目是开发该公司已被广泛应用的字处理软件的新版本，从上述事实至少可以得到3点结论：第一，旧版本相当于一个原型，通过收集用户对旧版本的反映，较容易确定对新版本的需求，没必要再专门建立一个原型系统来分析用户的需求；第二，该公司的软件工程师对字处理软件很熟悉，有开发字处理软件的丰富经验，具有采用增量模型开发新版字处理软件所需要的技术水平；第三，该软件受到广大用户的喜爱，今后很可能还要开发更新的版本，因此，应该把该软件的体系结构设计成开发式的，以利于今后的改进和扩充。综上所述，采用增量模型来完成这个项目比较恰当。
2. **设计下列伪代码程序的语句覆盖和路径覆盖的测试用例：**

**Code**

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 | INPUT(A, B, C) IF A>5   THEN X=10  ELSE X=1 END IF IF B>10   THEN Y=20  ELSE Y=2 END IF IF C>15   THEN Z=30  ELSE Z=3 END IF PRINT(X, Y, Z) STOP |

答：  
语句覆盖：每条语句都要被覆盖到  
所以是A>5，B>10，C>15  
路径覆盖：每条路径都要被覆盖到  
A、B、C各有真和假这两条路径  
组合有2*2*2=8条路径  
A>5，B>10，C>15  
A>5，B>10，C<=15  
A>5，B<=10，C>15  
A>5，B<=10，C<=15  
A<=5，B>10，C>15  
A<=5，B>10，C<=15  
A<=5，B<=10，C>15  
A<=5，B<=10，C<=15