**软件工程题目**

# 1. 什么是三视点模型？每一个模型的特点和作用？

三视点模型，即对象模型化技术（OMT），它要求把分析时收集的信息建立在三个模型当中，分别是描述系统数据结构的对象模型，描述系统控制结构的动态模型和描述系统功能的功能模型。

**对象模型：**是表示静态的、结构化的系统的数据性质，它是对模拟客观世界实体的对象以及对象彼此之间关系的映射，描述了系统的静态结构。如**类图**。

**动态模型：**是表示瞬时的、行为化的系统的控制性质，它规定了对象模型中的对象的合法变化序列，一旦建立起对象模型之后，就需要考察对象的动态行为。如**状态转换图，泳道图，活动图，协作图，顺序图**等等。

**功能模型：**表示变化的系统的功能性质，它指明了系统应该做什么，因此更直接地反应了用户对目标系统的需求，功能模型通常由一组数据流图组成，其有助于软件开发人员更深入地理解问题域，改进和完善自己的设计。如**数据流图，用例图。**

# 2. 比较面向对象方法和面向过程方法的不同。

（1）面向过程方法：是一种以过程为中心的编程思想，也可称之为“面向记录”的编程思想，他们不支持丰富的面向对象特性（比如继承、多态），并且它们不准许混合持久化状态和域逻辑，就是分析出解决问题所需的步骤，然后用函数一步步实现这些步骤，使用时一个个调用即可。

（2）面向对象方法：其出发点和基本原则，是尽可能模拟人类习惯的思维方式，使开发软件的方法与过程尽可能接近人类认识世界解决问题的方法与过程，也就是使描述问题的问题域与实现解法的求解域在结构上尽可能一致。其要点是1.认为客观的世界是由各种对象组成，任何事物都是对象，复杂事物可由简单对象以某种方式组合而成。2把所有对象划分成各种对象类，每个对象类都定义了一组数据和一组方法3按照子类和父类关系，把若干个对象类组成一个层次结构的系统。4对象彼此之间仅能通过传递消息互相联系。

**方法不同：**  
面向对象以数据为中心，为了数据来组织功能；  
面向过程是以功能为中心，为了功能来组织数据。  
**过程不同：**  
面向对象经典的是喷泉模型  
面向过程经典的是瀑布模型  
**工具不同：**

面向对象程序设计方法可以利用框架产品（如MFC，Microsoft Foundation Classes）进行编程。 面向对象可利用框架 面向对象和面向过程的根本差别，在于封装之后，面向对象提供了面向过程不具备的各种特性，最主要的，就是继承和多态。

# 3. 模块独立的重要性/模块化的意义。

模块化就是把程序划分成独立命名且可独立访问的模块，每个模块完成一个子功能，把这些模块集中起来构成一个整体，可以完成指定的功能满足用户需求。

**模块独立的重要性**：

第一，具有独立的模块的软件比较容易开发出来。这是由于能够分割功能而且接口可以简化，当许多人分工合作开发同一个软件时，这个优点尤其重要。

第二，独立的模块化比较容易测试和维护。这是因为相对来说，修改设计和程序需要的工作量比较小，错误传播范围小，需要扩充功能时能够插入模块。

总之，模块独立是好设计的关键，而设计又是决定软件质量的关键环节。

**模块化意义：**

把问题分解成多个小的独立、互相作用的组件，来处理复杂大型的软件，降低软件复杂性；

提高系统的稳定性；

有效阻隔错误的传播；

便于修理维护系统。

# 4. 什么是模块，模块、构件、类的关系。

**模块：**

模块是数据说明、可执行语句等程序对象的集合。模块可以被单独命名，而且可通过名字来访问，模块是构成程序的基本构件，并具有接口、功能、逻辑和状态属性。

**关系：**

在软件体系结构中，模块是可组合、分解和更换的单元，构件是面向软件体系结构的可复用软件模块，可被用来构造其他软件，它可以是被封装的对象类、类树、一些功能模块、软件框架等。在纯面向对象的设计中，类、封装、继承三者缺一不可，但对构件没有继承性，只要实现封装即可，类对软件的复用是通过继承实现的，构件对软件的复用不仅可以通过继承还可以通过组装时的引用来实现。

# 5. 什么是分析模型和设计模型？他们之间的关系。

**分析模型：**包含用例分析的结果、工件：分析类的实例，是设计模型的一个抽象，包含分析类和任何与之相关的工件。

**设计模型：**是一个描述用例实现的对象模型，它可作为对实施模型及其源代码的抽象。设计模型可以或多或少地接近实施模型，接近程度取决于如何将它的类、包和子系统映射到实施模型中的构件、包和子系统。

**关系：**通常，分析模型中的分析类可直接演进为设计模型中的元素，某些分析类变为设计类，其他分析类变为设计子系统，一个分析类可以成为设计模型中的单个类、某个类的一部分或一个聚合关系类，分析的目标是确定一种从所需行为到系统中建模元素的初步映射，设计的目标是将此初步映射转换成可实施的模型元素集，结构，模型元素从分析阶段演进到设计阶段后，细节和精度都得到精化。

# 6. 计算机系统结构的变化和软件工程发展的关系。

计算机系统结构经历了串行计算时代和并行计算时代，并行计算是在串行计算的基础上发展起来的，并行计算将一项大规模的计算任务交由一组相同处理单元共同完成，在此期间，各处理单元相互通信与协作，从而获得更高的效率。体系结构的发展是每个计算时代到来的重要标志，目前的计算机系统结构仍是冯若依曼结构，由运算器、控制器、存储器及输入输出设备五部分组成，但随着技术发展，计算机系统结构正朝着数据流结构、归约系统结构智能系统结构方向发展。计算机系统结构的不断变化使得软件开发一直变化，由此产生了软件危机，迫使人们不得不研究、改变软件开发的技术手段和管理方法，从此软件开发进入了软件工程时代，此阶段的特点是：硬件已向巨型化、微型化、网络化和智能化四个方向发展，数据库技术已成熟并广泛应用，第三代、第四代语言出现。

# 7.计算机语言的发展有哪些特点？和软件工程发展的关系？

计算机语言是人与计算机之间传递信息的媒介，其发展经历了从**机器语言、汇编语言到高级语言**的历程，机器语言由一组0、1数字，按一定的规则排列组成，若要计算机执行一项简单的任务，需要编写大量的这种指令，非常繁琐，复杂；汇编语言由比较容易识别、记忆的助记符替代特定的二进制串，但这些助记符号计算机无法识别，需要一个专门的程序将其翻译成机器语言，其可移植性与机器语言一样不好；高级语言与自然语言和数学表达式相当接近，通用性好，其大大提高了程序编写的效率和程序的可读性，尤其是面向对象语言的出现，使得软件开发领域掀起了巨大的变革，这也极大促进了软件工程的发展，面向对象方法学是软件工程方法学中非常重要的内容，面向对象方法在概念和表示方法上的一致性，保证了在各项开发活动之间的平滑过渡，软件工程中面向对象方法特有的继承性和多态性，进一步提高了面向对象软件的可重用性。

# 8.需求模型到设计模型的转换关系

需求模型是软件定义时期的最后一个阶段，它的基本任务是准确的回答“系统必须做什么？”这个问题，在该阶段结束之前，系统分析员应该写出软件需求规格说明书，以书面形式准确地描述软件需求。经过需求分析阶段的工作，系统必须“做什么”已经清楚了，设计模型阶段进而出现，该阶段主要由两部分组成：系统设计阶段，确定系统的具体实现方案；结构设计阶段，确定软件结构；其首先寻找实现目标系统的各种不同的方案，需求分析阶段得到的数据流图是设想各种可能方案的基础。设计模型阶段的目的就是通过需求模型阶段的软件规格说明书来确定如何完成预定的任务，即应该确定系统的物理配置方案，并且进而确定组成系统的每个程序的结构。

# 9.层次系统结构和基于消息的层次系统结构有什么区别？

层次系统组织成一个层次结构，每一层为上层服务，并作为下层客户。这种风格支持基于可增加抽象层的设计。这样，允许将一个复杂问题分解成一个增量步骤序列的实现。每一层最多影响两层。在一些层次系统中，除了一些精心挑选的输出函数外，内部层次只对相邻层可见，这样的系统中，构建在一些层实现了虚拟机（在另一些层次系统中层是部分不透明的），连接件通过决定层间如何交互的来定义，拓扑约束包括相邻层间的交互约束。

JB/ HMB风格基于层次消息总线、支持构件的分布和并发,构件之间通过消息总线进行通讯消息总线是系统的连接件,负责消息的分派、传递和过滤,以及处理结果的返回;各个构件挂接在消息总线上,向总线登记感兴趣的消息类型;构件根据需要发出消息,由消息总线负责将该消息分派到系统中所有对此消息感兴趣的构件,消息是构件之间通讯的唯一方式;构件接收到消息后,根据自身状态对消息进行响应,并通过总线返回处理结果.由于构件通过总线进行连,并不要求各个构件具有相同的地址空间或局限在一台机器上。

# 10、计算100个以内的有效数（规定值域之内）的平均值；有效数的总和；有效数的个数。

**程序：**

1: i =1; total.input=total.valid=0; sum=0;

2,3: DO WHILE value[i] < >-999 AND total.input<100

4: total.input=total.input +1;

5,6 IF value[i] >= minimum AND value[i]<= maximum

7: THEN total.valid=total.valid +1; sum= sum+ value[i];

8: ENDIF

i=i+1;

9: ENDDO

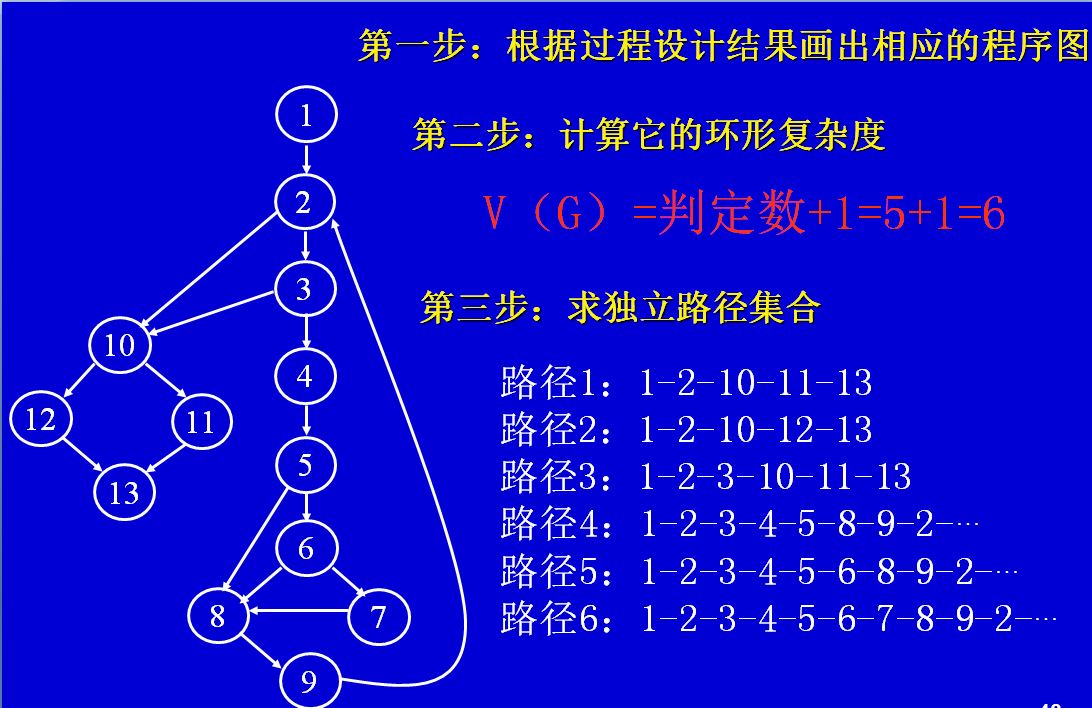
10: IF total.valid >0

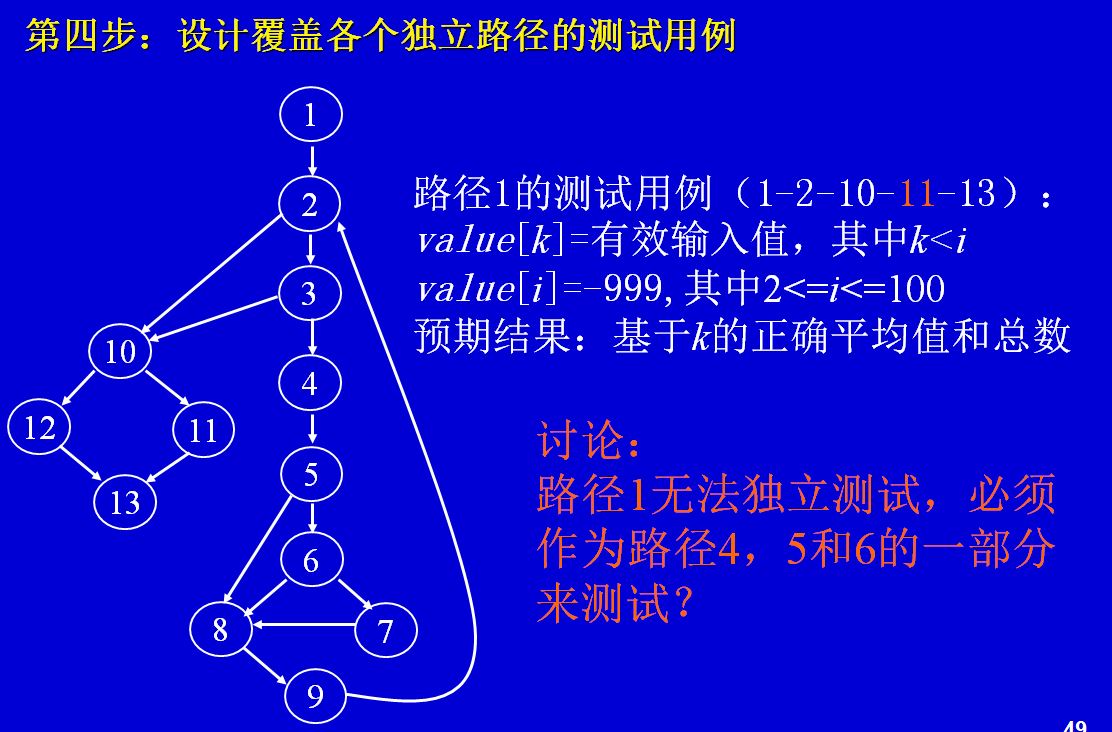
11: THEN average=sum / total.valid ;

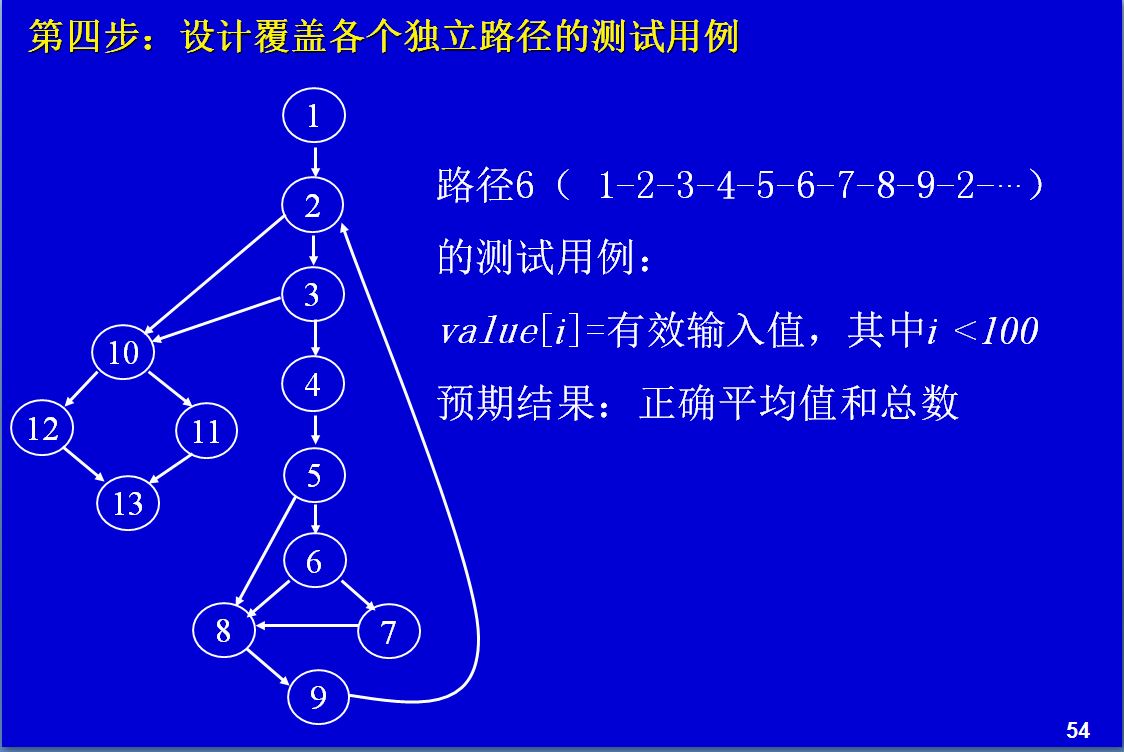
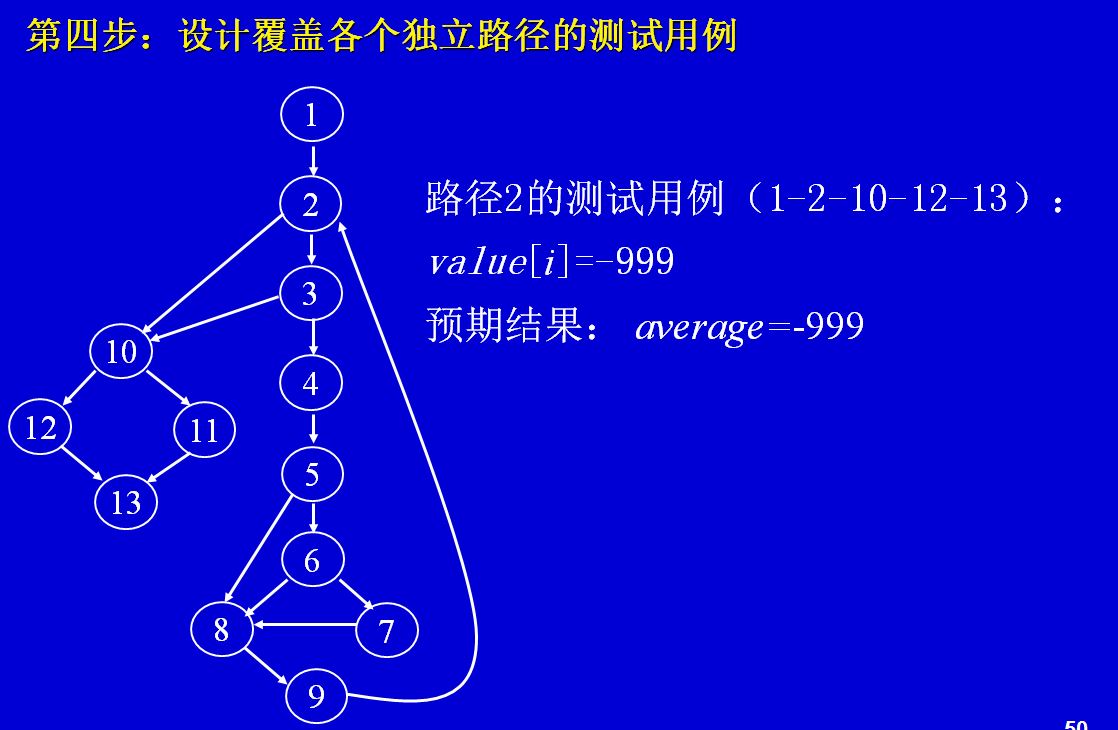
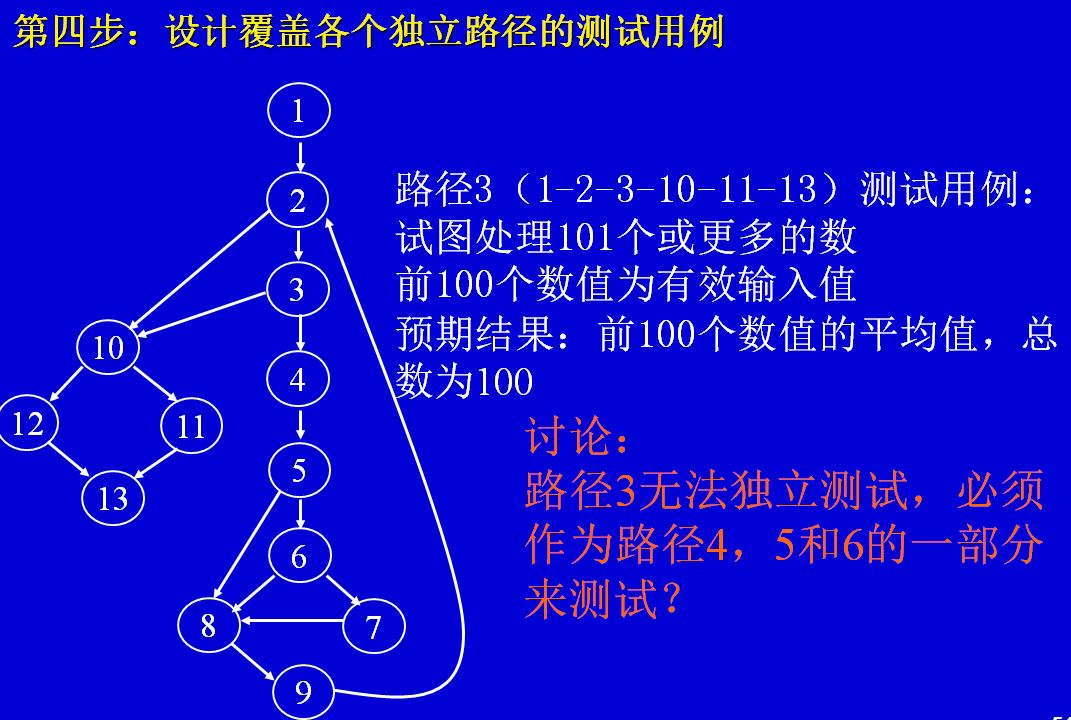
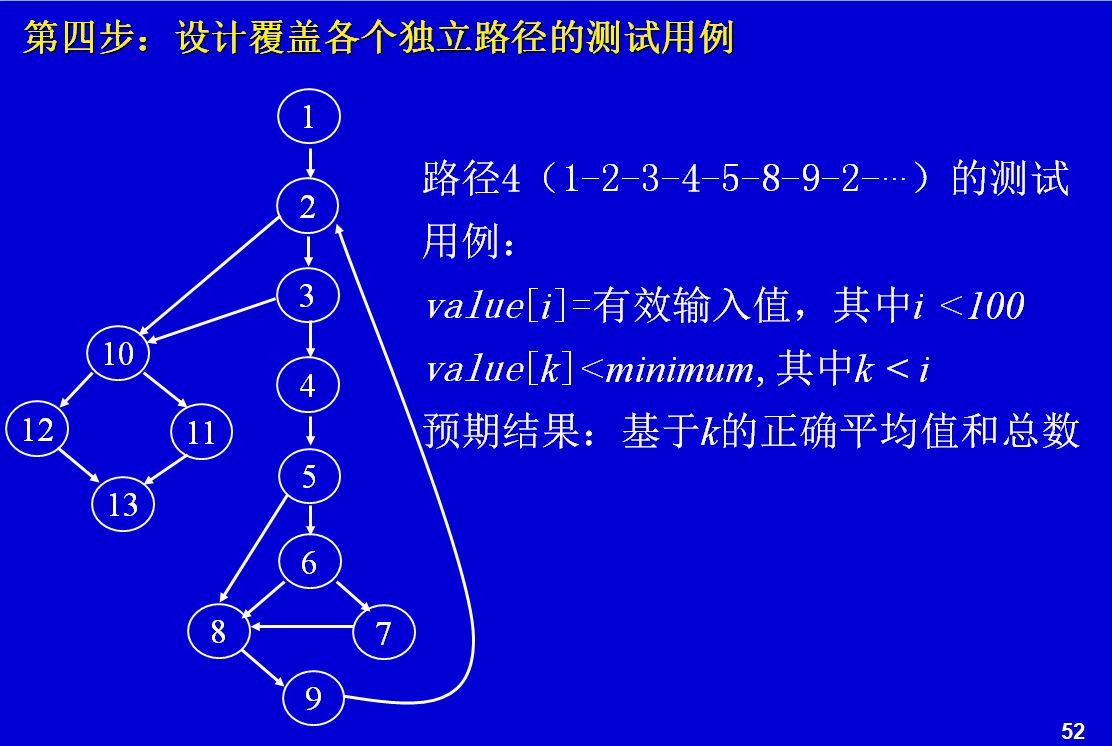
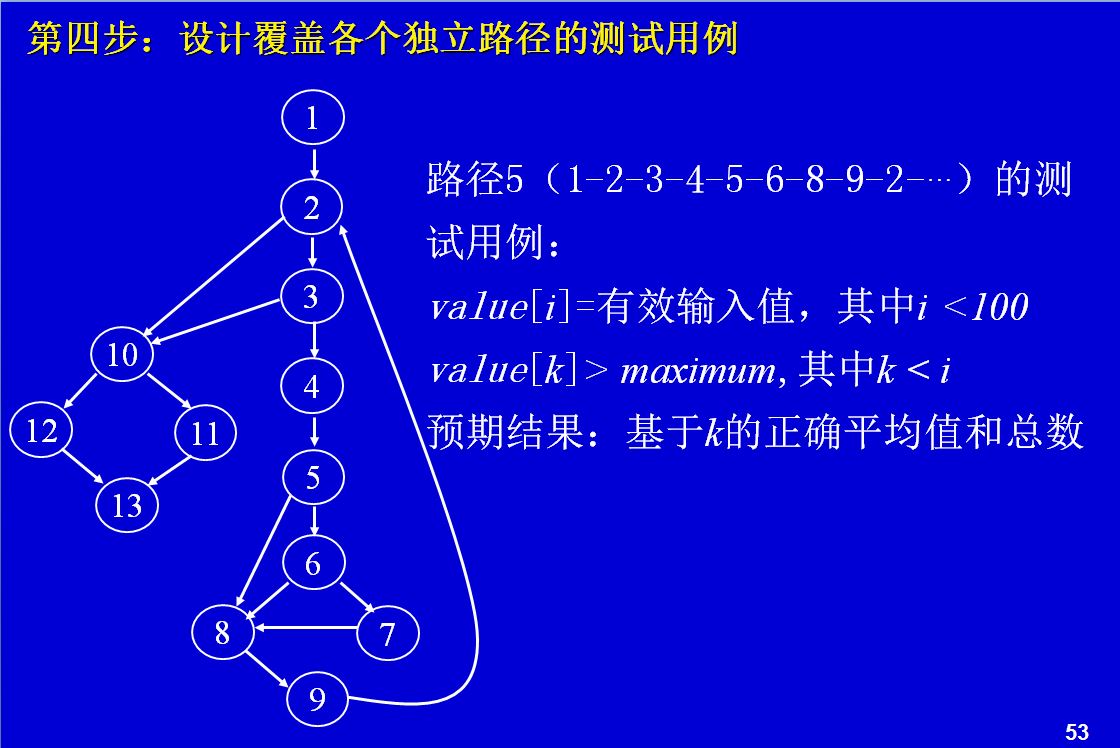
12: ELSE average=-999

13: ENDIF

END average







# 11、软件测试阶段的划分；不同阶段关联的文档。

注意：（本题第一个答案来自网上，但是上课老师不是按照这样的思路说的，我自己整理的第二个答案来自书本，但是关联的相关文档没有找全。）

**答案一：**

　（1）测试需求分析阶段主要工作是获得测试项目的测试需求（测试规格）。

　　输出产物：《可测试性需求说明书》和《测试规格》

　（2）测试计划阶段。

　　以测试需求为基础，分析产品的总体测试策略。

　　输出产物：《产品总体测试策略》

　　（3）测试方案设计阶段。

　　本阶段主要是以测试规格为基础获得特性测试方案，对于有自动化测试的项目，进行自动化测试的分析，获得测试策略。

　　输出产物：《产品或者版本总体测试方案》

　　（4）测试用例实现阶段。

　　本阶段主要是完成各个特性的测试用例的编写和自动化脚本的编写。

　　输出产物：《产品自动化测试用例》和《手工执行测试用例》

　　（5）测试执行阶段。

　　本阶段是根据测试策略开展测试执行和回归测试。

　　输出产品：《产品或版本测试报告》和《缺陷分析报告》

　　（6）评估与关闭阶段。

　　只对前面的各个阶段的执行情况，完成对测试项目的关闭，同时提供完整的度量数据和项目总结报告。

**答案二：**

1）单元测试：源程序清单，详细设计说明书

2）集成测试：详细设计文档

3）确认测试：需求规格说明书，软件计划，用户文档，开发文档，支持环境，源程序文本

4）系统测试：用户手册