二、名词解释题

1. 耦合的概念P97：是对一个软件结构内不同模块之间互连程度的度量。

2. 软件工程的定义P5：①把系统的、规范的、可度量的途径应用于软件开发、运行和维护过程，也就是把工程应用于软件；②研究①中提到的途径。

3. 软件危机的概念P1-2：软件危机是指在计算机软件开发、使用与维护过程中遇到的一系列严重问题和难题。它包括两方面：1.如何开发软件，满足对软件日益增长的需求；2.如何维护数量不断增长的已有软件。

4. 软件过程：软件过程是为了开发出高质量的软件产品所需完成的一系列任务的框架，它规定了完成各项任务的工作步骤。

5. 软件维护：概括地说，软件维护就是在软件已经交付用户使用之后，为了改正软件中的错误或使软件满足新的需求而修改软件的过程。

1. 软件生命周期：

软件生命周期3个时期 P13：软件定义、软件开发、软件维护

软件定义阶段：确定开发过程的总任务；工程的可行性；实现工程目标应采用的策略；系统必须完成的功能；估计资源和成本。

软件开发阶段：总体设计，详细设计（前两个阶段又称系统设计），编码和单元测试，综合测试（后两个阶段又称系统实现）。

1. 基线：

简要定义：基线就是通过了正式复审的软件配置项

IEEE定义：是已经通过了正式复审的规格说明或中间产品，它可以作为进一步开发的基础，并且只有通过正式的变化控制过程才能改变它。基线就是通过了正式复审的软件配置项。建立基线的三大原因是：重现性、可追踪性和报告。

8. 内聚标志着一个模块内各个元素之间彼此结合的紧密程度，它是信息隐藏和局部化概念的自然扩展。

9. 回归测试P160：指重新执行已经做过的测试的某个子集，以保证由于调试或其他原因引起的变化，不会导致非预期的软件行为或额外错误的测试活动。

10. 面向对象：

1. 简答题
2. 瀑布模型，原型模型与螺旋模型分别适用于开发哪些软件？

瀑布模型：项目周期较短。需求是预知的，软件实现方法是成熟的；

原型模型：无完整的需求说明，只有一些基本要求。开发出的软件产品通常能满足用户的真实需求，开发过程是线性顺序过程。

螺旋模型：适用于庞大、复杂并具有高风险的系统。

1. 可行性研究包括哪几个方面？并简要说明各方面的内容

一般说来，至少应该从以下三个方面研究每种解法的可行性：

a.技术可行性。对要开发项目的功能、 性能和限制条件进行分析， 确定在现有的资源条件下，技术风险有多大，项目是否能实现，这些即为技术可行性研究的内容。这里的资源包括已有的或可以搞到的硬件、软件资源，现有技术人员的技术水平和已有的工作基础。

b.经济可行性。进行开发成本的估算以及了解取得效益的评估， 确定要开发的项目是否值得投资开发，这些即为经济可行性研究的内容对于大多数系统，一般衡量经济上是否合算，应考虑一个“底线”，经济可行性研究范围较广，包括成本—效益分析， 长期公司经营策略， 开发所需的成本和资源，潜在的市场前景。

c.操作可行性。有时还要研究社会可行性问题，研究要开发的项目是否存在任何侵犯、妨碍等责任问题。社会可行性所涉及的范围也比较广，它包括合同、责任、 侵权和其他一些技术人员常常不了解的陷阱等。

必要时还应该从法律、社会效益等更广泛的方面研究每种解法的可行性。

1. 为什么说软件测试是软件开发中不可缺少的重要一环，但不是软件质量保证的安全网？
2. 软件测试是软件开发中不可缺少的重要一环，原因是：  
   ・测试的工作量约占整个项目开发工作量的 40% 左右，几乎一半。如果是关系到人的生命安全的软件，测试的工作量还要成倍增加。  
   ・软件测试代表了需求分析、设计、编码的最终复审。  
   ② 软件测试不是软件质量保证的安全网，因为软件测试只能发现错误，不能保证没有错误。
3. 为什么会产生软件危机？软件危机的表现是什么？

软件危机是指在计算机软件开发、使用与维护过程中遇到的一系列严重问题和难题。它包括两方面：如何开发软件，已满足对软件日益增长的需求；如何维护数量不断增长的已有软件。

软件危机的典型表现：

(1) 对软件开发成本和进度的估计常常很不准确。常常出现实际成本比估算成本高出一个数量级、实际进度比计划进度拖延几个月甚至几年的现象。而为了赶进度和节约成本所采取的一些权宜之计又往往损害了软件产品的质量。这些都降低了开发商的信誉，引起用户不满。

(2) 用户对已完成的软件不满意的现象时有发生。

(3) 软件产品的质量往往是靠不住的。

(4) 软件常常是不可维护的。

(5) 软件通常没有适当的文档资料。文档资料不全或不合格，必将给软件开发和维护工作带来许多难以想象的困难和难以解决的问题。

(6) 软件成本、软件维护费在计算机系统总成本中所占比例逐年上升。

(7) 开发生产率提高的速度远跟不上计算机应用普及的需求。

5.软件生命周期可以分为哪几个阶段，每个阶段的基本任务又是什么？

①软件生命周期由软件定义、软件开发、和运行维护 3 个时期组成。

②软件定义时期的主要任务是解决 “做什么” 的问题，即明确工程的总目标和可行性；导出实现工程目标应使用的策略及系统必须完成的功能；估计完成工程需要的资源和成本；制定工程进度表。该时期的工作也就是常说的系统分析，它通常又分为 3 个阶段：问题定义、可行性研究和需求分析。

软件开发时期的主要任务是解决 “如何做” 的问题，即具体设计和实现在前一个时期定义的软件，通常由概要设计、详细设计、编码和测试 4 个阶段完成。

软件运行维护时期的主要任务是使软件持久地满足用户的需要，通常由 4 类维护活动：改正性维护，也就是诊断和改正正在使用过程中发现的软件错误；适应性维护，即修改软件以适应环境的变化；完善性维护，即根据用户的要求改进或扩充软件，使它更完善；预防性维护，即修改软件为将来的维护活动预先做准备。

6.模块独立的程度可以由内聚和耦合来进行度量，请简述耦合和内聚两个度量指标的意

义，并说出模块开发过程中对于耦合和内聚程度应该遵循的一般设计原则是什么

①耦合的概念：是对一个软件结构内不同模块之间互连程度的度量

②内聚标志着一个模块内各个元素之间彼此结合的紧密程度，它是信息隐藏和局部化概念的自然扩展。

③（耦合设计原则：尽量使用数据耦合，少用控制耦合和特征耦合，限制公共环境耦合的范围，完全不用内容耦合。）

④（内聚设计原则：设计时力争做到高内聚，并且能够辨认出低内聚的模块，有能力通过修改设计提高模块的内聚程度并且降低模块间的耦合程度，从而获得较高的模块独立性）

7.模块的扇入和扇出分别是什么意思 ？一般情况下扇出太大或扇出太小意味着什么，应当如何处理？

扇出：是一个模块直接控制的模块数目。

扇入：表明有多少个上级模块直接调用它。扇入越大，共享该模块的上级模块的数目越多。

扇出过大意味着模块过分复杂，需要控制和协调过多的下级模块；扇出过小（例如总是1）也不好

扇出太大一般是因为缺乏中间层次，应该适当增加中间层次的控制模块。扇出太小时可以把下级模块进一步分解成若干个功能模块，或者合并到它的上级模块中去。

8.请从软件项目管理者的角度论述如何从软件开发的过程保障软件的质量

答：①在每个阶段结束前的技术审查和管理复查中，应该着重对可维护性进行复审，应该对将来要改进的部分和可能要改的部分加以注意指明

②应该讨论软件的可移植性问题，考虑可能影响软件维护的系统界面。

③在设计和编码过程中应该尽量使用可重用的软件构件，在软件正式交付使用之前，程序中可能需要做预防性维护的部分。

④在完成每项维护工作之后，都应该对软件维护本身仔细认真地复审。

1. 软件的可维护性与哪些因素有关？ 重修预理移

（1）、可理解性

           （2）、可预测性

           （3）、可修改性

           （4）、可移植性

           （5）、可重用性

1. 选择题

①用面向对象的方法开发软件，通常要建立三种模型，这三种模型分别是对象模型（静态结构）、动态模型（交互次序）、功能模型（数据变换）