

第一章

1. 软件是计算机程序、方法、规则、相关的（ ）以及在计算机上运行程序时所必需的数据。

答案:

(1) 文档资料

2. 按照软件产品的作用可分成基础软件、（ ）、应用软件和嵌入式软件 。

答案:

(1) 支撑软件

3. 软件工程是指应用（ ）、（ ）以及（ ）等原理，以工程化的原则和方法来解决软件问题的工程。其目的是提高软件的生产率、提高质量、降低成本、满足用户需求。

答案:

(1) 计算机科学

(2) 数学

(3) 管理科学

二. 简答题（共 10 题, 77.2 分）

1. 简述 IEEE 为软件下的定义。

答案:

计算机程序、方法、规则、相关的文档资料以及在计算机上运行程序时所必需的数据。

2. 简述“软件=程序+数据+文档”这个定义中的程序、数据和文档的概念。

答案:

程序和数据是软件运行的基本要素，而文档则是延续软件运行寿命的保障。

程序：按功能和性能需求事先设计的可执行的指令序列

数据：是程序能正常操纵信息的数据结构

文档：与程序开发、维护和使用有关的图文材料

3. 简述软件产品的特点有哪些？

答案:

1) 软件产品自身特点

（1）软件产品是一种抽象的逻辑产品。没有明显的物理加工过程，是一种把人的知识和技术、环境信息高度密集在一起的产品。因此工作进度、效率、质量等指标的观测性低，造成工作进展情况难以衡量、质量难以评价、开发过程难以管理和控制。

（2）功能的实现与硬件和支持环境密切相关，不能单独存在。

（3）可大量复制，知识产权保护困难。

（4）不会磨损，但会随时间和环境变化而退化。

（5）维护困难，费用越来越高。

2) 业界环境特点

（1）开发者与用户交流困难

（2）构件和复用技术不够不理想

（3）人员流动性大，质量和工期难以保障

4. 简述按照软件产品的作用，可将软件分成哪些类型？

答案：

按照软件产品的作用分成基础软件、支撑软件、应用软件和嵌入式软件。

1. 基础软件：用于保证计算机本身运行，并为应用软件的运行提供基础环境的软件；基础软件又包括系统软件、数据库、中间件等软件。

①系统软件：操作系统、语言处理系统、通用办公软件

②数据库：如 Oracle、SQL Server、DB2、Informix、MySQL 等国外数据库软件，国产的数据库软件有东软 OpenBASE、金仓 KingbaseES 等。

③中间件：为处于自己上层的应用软件提供运行与开发的环境，帮助用户开发和集成应用软件。最突出的特点是其网络通信功能。中间件即是管理应用软件的运行容器。

2. 支撑软件：在系统软件和应用软件之间，为应用软件设计、开发、测试、评估、运行检测等提供辅助功能的软件。支撑软件包括

① 编程工具：微软的 Visual Studio、IBM 提供的开源集成开发环境 Eclipse 等；

② 建模工具：如 IBM Rational Rose 软件建模；

③ 版本控制、软件测试；

3. 应用软件：解决特定应用领域问题或解决具体问题而开发的软件；应用软件可分为桌面软件和企业（业务）级软件。

1) 桌面软件：个人机上部署的应用软件。

2) 企业级软件：企业级软件是指那些为商业组织、大型企业、特定业务领域、行政管理、事业部门而创建并部署的软件。

4. 嵌入式软件：就是嵌入在硬件中的操作系统和开发工具软件及应用程序；应用及其广泛。例如手机、移动设备、数码相机、GPS、智能家电。

5. 什么是软件危机？产生软件危机的原因有哪些？软件危机的表现有哪些？

答案：

软件危机泛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。

出现软件危机的原因主要有：

（1）软件本身独有的特点确实给开发和维护带来了困难

（2）软件开发和维护过程不规范

（3）缺乏软件开发的经验和有关软件开发数据的积累

（4）轻视，人们在制定计划时总会有一些天马行空的想法和要求，轻视是一个最大的错误。

（5）忽视软件需求分析的重要性、忽视软件的可理解性、文档不完备、轻视软件的可维护性、过分强调编码技巧等等方面。

（6）用户与开发人员的沟通问题，从而技术人员获取的需求不够全面。

软件危机的表现：

（1）软件开发进度难以预测，开发成本难以控制，导致超预算、超时；

（2）产品功能难以满足用户需求；

（3）软件产品质量无法保证；

（4）软件缺少适当的文档资料，维护困难；

（5）软件成本超过硬件成本；

（6）软件开发生产率提高的速度远远跟不上计算机应用迅速普及深入的趋势；

6. NATO(北大西洋公约组织)会议上给出的软件工程定义。

答案:

软件工程就是为了经济地获得可靠的且能在实际机器上有效地运行的软件，而建立和使用完善的工程原理。

7. IEEE 给出的软件工程的定义:

答案:

软件工程是:

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。

8. 软件工程的定义。

答案:

软件工程是指应用计算机科学、数学以及管理科学等原理，以工程化的原则和方法来解决软件问题的工程。其目的是提高软件的生产率、提高质量、降低成本、满足用户需求。

9. 简述软件工程的三个要素有哪些?

答案:

软件工程的三个要素是过程、方法 和 工具。其中的过程是软件开发经历的一系列有组织的活动，从而能够合理地及时开发出计算机软件。方法是开发软件的方法，它涵盖了项目计划、需求分析、系统设计、程序实现、测试与维护等一系列的开发活动所要使用的方法。软件开发方法经历了从面向结构、面向对象、面向组件到面向服务的发展过程。

工具是指软件开发过程中需要的工具，工具为过程和方法提供了自动的或半自动的支持。这些软件工具被集成起来，建立起一个软件开发支持系统，称之为计算机辅助软件工程（CASE，Computer Aided Software Engineering）。CASE 集成了软件、硬件和一个存放开发过程信息的软件工程数据库，形成了一个软件工程环境。

10. 简述软件工程的基本原理。

答案:

1. 用分阶段的生命周期计划严格管理;
2. 坚持进行阶段评审;
3. 实行严格的产品控制;
4. 采纳现代程序设计技术;
5. 结果应能清楚地审查;
6. 开发小组的人员应少而精;
7. 承认不断改进软件工程实践的必要性。

第二章

一. 填空题（共 4 题, 14.0 分）

1. 软件计划时期是软件项目的（ ）阶段。该阶段的主要活动包括（ ）和（ ）；其具体活动内容是确定软件的（ ）、（ ）、（ ）和（ ）；

答案:

- (1) 初始
- (2) 问题定义
- (3) 可行性研究
- (4) 开发目的
- (5) 范围
- (6) 约束
- (7) 技术方案

2. 软件开发时期的主要活动包括需求分析、（ ）、（ ）和实现；其活动内容包括逐步细化软件模型、经过（ ）和（ ），最终得到可运行的软件。该阶段的目标是构建高质量的软件。

答案:

- (1) 详细设计
- (2) 详细设计
- (3) 编码
- (4) 测试

3. 软件运行时期的主要活动是（ ）和（ ）；其活动内容包括修改和进化软件；该阶段的工作目标是延长软件的寿命。

答案:

- (1) 运行；
- (2) 软件维护

4. 任何一个软件过程都可以看成是由一系列（ ）或过程组成的有序序列。任何软件过程都应遵循软件工程的（ ），研究软件开发方法论，规范软件开发的活动（ ）和活动（ ），提高软件质量。

答案:

- (1) 动作
- (2) 集合
- (3) 顺序

二. 简答题（共 24 题, 86.0 分）

1. 简述软件生存周期的概念。软件生存周期划分成哪三个阶段？

答案:

软件生存周期是指软件产品从定义到开发、使用和维护，直到最终被弃用的时期，任何一个软件都有其特定的生存周期。

软件生存周期通常被划分成计划、开发和运行三个时期。

2. 简述软件开发过程中，参与的主要人员有哪些？他们的职责是什么？

答案：

客户：投资并通过使用软件获得利益的人员，客户通常是指企业法人。

用户或用户代表：以某种方式使用目标软件的人员。参与开发的通常是选择的用户代表。

分析员：与用户交流进行需求获取和需求分析的人员。

设计员：负责软件设计的人员。

程序员：负责编写程序代码的人员。

测试员：负责软件测试的人员。

3. 简述软件生存周期包含了哪七个阶段。

答案：

软件生存周期的七个阶段包括问题定义、可行性研究、分析、概要设计、详细设计、实现和软件维护；

4. 简述软件过程(Software Procedure)的定义。

答案：

软件过程是为获得软件产品，在软件工具支持下由软件工程师完成的一系列软件工程活动。

5. 简述软件过程的分类有哪几种？

答案：

软件过程被分成基本过程、支持过程和组织过程。

6. 简述什么是基本软件过程？基本过程中包含了那些软件过程？

答案：

软件过程中，软件开发的主干过程被称为基本软件过程，这些过程是软件开发的实质性过程。

这些过程包括获取过程、供应过程、开发过程、运作过程、维护过程和管理过程等。

7. 简述什么是瀑布模型？瀑布模型的主要特点是什么？瀑布模型有哪些局限？

答案：

1) 瀑布模型

严格按照生命周期的顺序进行软件开发的模型叫做瀑布模型，其活动的内容和顺序就是问题定义、可行性研究、需求分析、概要设计、详细设计、编码、测试和维护。由于整个开发过程的活动是顺序的，因此也称线性顺序模型。

2) 瀑布模型的特点

①各项活动之间具有严格的顺序性和依赖性；

②推迟实现的观点；充分做好前期的分析和设计工作，将能够设计与实现分离，尽可能推迟物理实现，减少返工。

③质量保证的观点；每个阶段必须完成规定的、完整、准确的合格文档；前一阶段的输出文档就是后一阶段的输入文档。

3) 瀑布模型的局限

瀑布模型是一种理想化的文档驱动模型，阶段之间依赖书面的、正确的规格说明。对于瀑布模型来说：

需求分析是项目成败的关键；不适合需求模糊的系统；很难适应需求变化；

8. 简述快速原型的概念？使用快速原型可以解决什么问题？

答案：

快速原型是快速建立起来的可以在计算机上运行的程序，它所能完成的功能往往是最终产品能完成的功能的一个子集。

使用快速原型可以解决需求模糊问题，快速原型法可以解决瀑布模型的局限性问题。

9. 简述快速原型模型的特点。

答案：

- ① 使用快速原型，可以及早向用户展示系统要实现的界面及功能，增强用户的合作信心；
- ② 可以直观地表达，容易交流，消除理解上的歧义；
- ③ 可以把对需求的修改集中在前期的原型确认上，较大程度减少后期实施过程中的返工。
- ④ 可以加快开发进度；

10. 简述增量模型的概念，给出对每个增量进行开发的步骤有哪些？使用增量模型开发软件的必要条件是什么？

答案：

将系统或分成若干个子集，先开发出一个基本的子集，再按同样的开发步骤增加子集, 如此递增下去直至满足全部系统需求。其中每个增量子集可按快速原型法进行。

开发每个增量的步骤有分析、设计、编码、测试和交付。

使用增量模型开发软件的必要条件是软件的各个子集是逐渐并入已有的系统中，加入新的子集不能破坏已构造好的部分，这需要软件具备开放式的体系结构。

11. 增量模型的特点有哪些？

答案：

增量模型的特点有：

无须等待获取完整需求就可开始后续过程，尽快见到成果，增强双方合作信心；

分步开发，降低复杂性和难度，减少技术风险，并可并行开发；

边开发边投入，可及早发现问题，减少投资风险；

各个子集是逐渐并入已有的系统中，加入子集不能破坏已构造好的部分，这需要软件具备开放式的体系结构；

适用于需求不完整的软件开发，指的是需求逐渐摸清、逐步完善，并非随意改变，需求改变过大会导致项目整体性失控。

12. 简述螺旋模型的概念，螺旋模型的演进过程。

答案：

螺旋模型是一种融合了瀑布模型、快速原型模型和增量模型的演进模型，并引入风险分析机制。特别适合大型复杂的系统开发。

螺旋模型的演进过程如下：

螺旋模型将软件开发过程划分成问题定义和可行性研究、需求分析、总体设计、和详细设计编码测试等四个阶段，并把每个阶段分成制定计划、风险分析、实施工程和客户评估等四个小的阶段。其中

制定计划用于确定该阶段的工作目标，解决方案和限制条件（约束）。

风险分析的任务是评估该阶段的方案，识别和消除方案的风险，必要时可以编制对应的原型，可确定、修改或终止项目；
实施工程实施该阶段的具体任务。
客户评估的任务是评价前一阶段的开发工作。

13. 简述螺旋模型的特点有哪些？

答案：

螺旋模型的特点有：

- ①多种模型结合的一种演进模型，融合了瀑布模型、快速原型和增量模型的特点，融进了循环往复、迭代演进的思想；
- ② 增加风险分析，一旦风险成立，原方案应终止、修订，力求风险可控，
- ③ 客户参与每个阶段的开发, 每个阶段的成果需客户确认，避免错误的积累。

14. 简述统一过程的核心工作流有哪些？

答案：

统一过程定义了九个核心工作流，其中包含六个核心过程工作流(Core Process Workflows)和三个核心支持工作流(Core Supporting Workflows)。

其核心过程工作流(Core Process Workflows)

- ①商业建模(Business Modeling) 商业建模是业务层面的活动，目的是弄清项目边界和约束，做出计划。
- ②需求(Requirements) 需求的目标是描述系统应该做什么，并使开发人员和用户就这一描述达成共识。
- ③分析和设计(Analysis & Design) 分析和设计将需求转化成计算机可以实现的模型，是将客观世界虚拟到计算机世界的逐步细化过程。
- ④ 实现(Implementation) 实现是用计算机可以理解的语言将设计模型组织成可执行的文件、数据，俗称编程。
- ⑤ 测试(Test) 测试是发现软件中的错误，在实验环境下验证所有的需求是否被正确的实现。
- ⑥ 部署(Deployment) 部署是将软件分发给最终用户，安装在真实的环境下，由用户操作运行。

核心支持工作流(Core Supporting Workflows)是对核心过程工作流的配套支持和管理，保障核心过程工作流顺畅、高效运行。包括：

- ①配置和变更管理(Configuration & Change Management) 配置管理(Configuration Management, CM)是通过技术或行政手段对软件产品及其开发过程和生命周期进行控制、规范的一系列措施。目的是记录软件产品的演化过程，确保在软件生命周期中各个阶段都能得到精确的产品配置。
- ②项目管理(Project Management) 对核心过程工作流进行资源配置、评估监控、风险控制、计划调整等管理工作，做到投入产出的效益最大化。
- ③环境(Environment) 环境是向软件开发组织提供软件开发环境，包括人员、设备、过程和工具，以及各种规范、指导手册和保障措施。

15. 简述敏捷开发的基本概念。

答案：

敏捷开发以用户的需求进化为核心，采用迭代、循序渐进的方法进行软件开发。在敏捷开发中，软件项目在构建初期被切分成多个子项目，各个子项目的成果都经过测试，具备可视、可集成和可运

行使用的特征。换言之，就是把一个大项目分为多个相互联系，但也可独立运行的小项目，并分别完成，在此过程中软件一直处于可使用状态。

16. 简述敏捷开发的观点和价值观。

答案：

敏捷开发的观点：

1. 个体和交互 胜过过程和工具
2. 可以工作的软件胜过面面俱到的文档
3. 客户合作胜过合同谈判
4. 响应变化胜过遵循计划

敏捷开发的价值观：沟通、简单、反馈、勇气，谦逊。

- 1) 沟通：团队内部人员之间沟通、团队和项目（stakeholder）涉众之间的沟通。
- 2) 简单：建模成为简化软件和软件（开发）过程的关键。
- 3) 反馈：通过图表交流想法，快速获得反馈，并按照建议行事。
- 4) 勇气：当决策证明不合适时，需要做出放弃或重构（refactor）的决策。
- 5) 谦逊：应该拥有谦逊美德，认识到自己并不是无所不知的。假设每一个项目参与者都有相同的价值，都应该被尊重。

17. 敏捷开发方法遵循一些共同原则。

答案：

敏捷开发方法遵循的共同原则：

- 1) 持续交付，事件间隔越短越好。可工作的软件是首要的进度度量标准。
- 2) 欢迎改变需求。
- 3) 业务人员和开发人员在一起工作。
- 4) 对团队开发人员充分信任。在团队内部提倡面对面的交谈。
- 5) 保持一个长期的、恒定的开发速度。
- 6) 优秀的技能和好的设计会增强敏捷能力。
- 7) 最好的构架、需求和设计出于自组织团队。
- 8) 简单是最根本的。

或

- 1-最高目标是通过尽早和持续交付有价值的软件来满足客户；
- 2-欢迎对需求提出变更 - 即使在项目开发后期，要善于利用需求变更，帮助客户获得竞争优势；
- 3-要不断交付可用的软件，周期从几周到几个月不等，越短越好
- 4-项目过程中，业务人员与开发人员必须在一起
- 5-要善于激励项目人员，给他们以所需要的环境和支持，并相信他们能够完成任务
- 6-无论是团队内还是团队间，最有效的沟通方法是面对面的交谈
- 7-可用的软件是衡量进度的主要指标
- 8-敏捷过程提倡可持续的开发，项目方，开发人员和用户应该能够保持恒久稳定的进展速度
- 9-对技术的精益求精以及对设计的不断完善将提升敏捷性
- 10-要做到简洁，尽可能减少不必要的工作，这是一门艺术
- 11-最佳的架构，需求和设计出自于自组织的团队
- 12-团队要定期反省如何能够做到更有效，并相应调整团队的行为。

18. 简述极限编程（Extreme Programming）的概念。

答案:

极限编程（英语：Extreme programming，缩写为 XP），是一种软件工程方法学，是敏捷软件开发中应用最为广泛和最富有成效的几种方法学之一。极限编程和传统方法学的本质不同在于它更强调可适应性而不是可预测性。极限编程的支持者认为软件需求的不断变化是很自然的现象，是软件项目开发中不可避免的、也是应该欣然接受的现象；他们相信，和传统的在项目起始阶段定义好所有需求再费尽心思的控制变化的方法相比，有能力在项目周期的任何阶段去适应变化，将是更加现实更加有效的方法。

19. 简述极限编程模型。

答案:

极限编程 (XP) 是敏捷方法中一组轻量级的软件开发实践。其主要特征是适应环境变化和需求变化，让客户全面参与软件的开发设计，充分发挥开发人员的主动精神，降低软件项目风险，改善业务变化的反应能力，提高开发期间的生产力。其开发模型如下：

- 1) 聆听客户需求 客户加入到开发团队，参与日常开发与沟通会议。客户向程序员讲解用户需求。
- 2) 测试 增加一个测试；运行所有测试，发现不能用过的测试；
- 3) 编码 修改程序代码，再测试，直到所有测试通过；
- 4) 设计 重构代码，消除重复设计，优化设计结构。

20. 简述极限编程的 12 个核心实践

答案:

极限编程的 12 个核心实践如下：

- 1) 完整团队
- 2) 计划游戏 XP 的计划过程主要考虑软件发布计划和周期开发计划，软件发布计划。
- 3) 小版本
- 4) 现场客户
- 5) 隐喻
- 6) 持续集成
- 7) 代码集体所有
- 8) 编码标准
- 9) 测试驱动开发
- 10) 重构
- 11) 简单设计
- 12) 结对编程

21. 简述 XP 极限编程中的重构的概念。

答案:

在软件工程学中重构（Refactoring）就是在不改变软件现有功能的基础上，通过调整程序代码改善软件的质量、性能，使其程序的设计模式和架构更趋合理，提高软件的扩展性和维护性。

22. 简述软件开发方法有哪些？

答案:

软件开发方法经历了如下一些方法的变化过程。

- 1) 结构化开发方法

- 2) 面向对象的软件开发方法
- 3) 面向组件的软件开发方法
- 4) 面向服务的软件开发方法
- 5) 面向云计算的软件开发方法

23. 简述软件过程改进的概念。

答案:

软件过程改进是指在软件过程的实施中为了更有效地达到优化软件过程的目的所实施的改善或改变其软件过程的系列活动。过程改进的实施是在认知现有软件过程的基础之上, 利用过程运作和过程监控中所获得的反馈信息, 发现软件过程中存在的问题和缺陷, 提出改进的意见, 进而实现软件过程的改进和完善。过程改进的实施对象就是企业的软件过程, 也就是软件产品的生产制造过程, 当然也包括软件维护之类的辅助过程。

24. 简述软件能力成熟度模型, 并详细给出成熟度模型的五个等级。

答案:

软件能力成熟度模型是一种对软件组织在定义、实施、度量、控制和改善其软件过程的实践中各个发展阶段的描述形成的标准。其英文全称为 Capability Maturity Model, 英文缩写为 SW-CMM, 简称 CMM。它是对于软件组织在定义、实施、度量、控制和改善其软件过程的实践中各个发展阶段的描述。CMM 的核心是把软件开发视为一个过程, 并根据这一原则对软件开发和维护进行过程监控和研究, 以使其更加科学化、标准化、使企业能够更好地实现商业目标。

CMM 是一种用于评价软件承包能力并帮助其改善软件质量的方法, 侧重于软件开发过程的管理及工程能力的提高与评估。CMM 分为五个等级: 一级为初始级, 二级为可重复级, 三级为已定义级, 四级为已管理级, 五级为优化级。

CMM/CMMI 将软件过程的成熟度分为 5 个等级, 以下是 5 个等级的基本特征:

(1) 初始级 (initial)。工作无序, 项目进行过程中常放弃当初的计划。管理无章法, 缺乏健全的管理制度。开发项目成效不稳定, 项目成功主要依靠项目负责人的经验和能力, 他一但离去, 工作秩序面目全非。

(2) 可重复级 (Repeatable)。管理制度化, 建立了基本的管理制度和规程, 管理工作有章可循。初步实现标准化, 开发工作比较好地按标准实施。变更依法进行, 做到基线化, 稳定可跟踪, 新项目的计划和管理基于过去的实践经验, 具有重复以前成功项目的环境和条件。

(3) 已定义级 (Defined)。开发过程, 包括技术工作和管理工作, 均已实现标准化、文档化。建立了完善的培训制度和专家评审制度, 全部技术活动和管理活动均可控制, 对项目进行中的过程、岗位和职责均有共同的理解。

(4) 已管理级 (Managed)。产品和过程已建立了定量的质量目标。开发活动中的生产率和质量是可量度的。已建立过程数据库。已实现项目产品和过程的控制。可预测过程和产品质量趋势, 如预测偏差, 实现及时纠正。

(5) 优化级 (Optimizing)。可集中精力改进过程, 采用新技术、新方法。拥有防止出现缺陷、识别薄弱环节以及加以改进的手段。可取得过程有效性的统计数据, 并可据进行分析, 从而得出最佳方法。

第三章

一. 简答题（共 6 题, 100.0 分）

1. 软件计划时期的工作目标和工作内容是什么？

答案:

软件计划时期的工作目标是研究项目的可行性，研究如何合理地运用软件项目开发所需的资源、经费，掌握和控制开发进度，控制项目开发过程按计划进行。

软件计划时期的主要工作内容包括：确定项目实施范围、定义递交的工作成果、评估实施过程中的主要风险、制定项目实施的时间计划、成本和预算计划、人力资源计划等。

2. 简述问题定义的目的和工作内容。

答案:

问题定义的目的是弄清要计算机解决的根本问题所在(要解决的问题是什么？)，确定新系统的作用域，以及项目所需的资源、工期和经费。

问题定义的主要任务是编写项目报告提交审查，作为可行性分析的依据。工作内容包括

1 分析员要进行系统初步调查，确定项目的性质、目标、规模。调查的内容包括系统目标、现行系统、其它系统、各种解决方案、数据元素、结构及算法。

2 撰写问题定义报告。

3 在组织内部正式成文“系统开发任务书”，是项目立项的第一个基础性文件。

3. 简述可行性分析的概念。

答案:

可行性分析的目的是确定项目是否必要和可行。这个阶段的任务是研究问题的范围，探索这个问题是否值得去解，在特定的限制条件下，问题能否得到解决。

4. 简述可行性分析的基本过程。

答案:

可行性分析的基本过程如下：

(1) 进一步分析和澄清问题的定义，在澄清问题的基础上，导出系统的逻辑模型（系统流程图 或 业务模型）

(2) 研究问题的解决方案 从系统逻辑模型中，选择问题的若干种主要解法，研究每一种解法的可行性，为以后的行动提出建议；

(3) 给出结论性意见 如果问题没有可行的解，建议停止系统开发；如果问题有可行的解，应该推荐一个较好的解决方案，并为项目制定一个初步的计划。

5. 简述可行性分析的内容。

答案:

可行性分析的内容包括：技术可行性、经济可行性和社会可行性(法律可行性)等三个方面；

1) 技术可行性 分析利用现有的技术能否实现，能否解决系统中的技术难题，所开发的系统能否达到所要求的功能和性能，系统对技术人员的要求，现有的技术人员能否胜任，开发所需要的软件与硬件能否如期得到等；

(1) 开发风险分析 在给定的约束条件下，判断能否设计和实现所提出的功能和性能。

(2) 资源分析 是否具备开发项目的相关人员、已有的技术和经验、设备、环境、培训等能力。

(3) 技术分析 现有技术能否实现新系统，技术难点、建议采用技术的先进性。

2) 经济可行性 分析项目的成本与收益，包括短期效益与长远利益。要作出投资的估算和系统投入运行后可能获得的经济效益或可节约的费用估算。

① 一次性开发费用的估算：主要考虑软硬件设备，人员工资，培训，材料消耗；

②年运行费用的估算：主要考虑维护人员，软硬件升级，必备物资和材料；

③效益估算：承制方—开发、维护成本；对投资方就是软件运行的经济效益和社会效益；

2) 法律可行性 考虑的是市场、政策与法律方面的问题，法律可行性分析的主要内容包括社会政治、侵权和用户接受等。

①社会政治：一定要考虑这样几件万不可违背的大事：国家法律，国家政策，国家统一，民族团结。

②侵权：考虑是否有侵犯知识产权、商标权、肖像权或宗教信仰；供需双方对软件产品的成果分享及转让权利。

③用户接受：考虑用户现行管理制度、人员素质、操作方式是否可行。

6. 教材中介绍了哪些可行性分析的工具？

答案：

本课程介绍了系统流程图和业务流程图两种模型？

(1) 系统流程图(System Flow Diagram, SFD) 采用图形符号描述当前系统的工作流程的一种图。SFD 表示物理部件和信息流动，SFD 是可行性分析当中描述系统物理模型的常用方法，SFD 是 IT 人员与领域人员都能理解的表达方式，借助 SFD 图使供需双方达到对问题的一致理解。

(2) 业务流程图(Transaction Flow Diagram, TFD) 是一种描述系统内各部门、人员之间业务关系、作业顺序和管理信息流向的图表，利用它可以帮助分析人员描述企业的业流程，找出业务流程中的不合理流向，以改进企业的业务过程。它是一种物理模型。

第四章

一.简答题（共 20 题,100.0 分）

1. 简述软件需求的概念。

答案:

软件需求是

- (1) 用户解决问题或达到目标所需条件或权能 (Capability)。
- (2) 系统或系统部件要满足合同、标准、规范或其它正式规定文档所需具有的条件或权能。
- (3) 一种反映上面 (1) 或 (2) 所述条件或权能的文档说明。它包括功能性需求及非功能性需求，非功能性需求是对软件的设计和实现提出的限制，比如性能要求，质量标准，或者设计限制。

2. 简述需求分析的概念。

答案:

需求分析是开发人员经过深入细致的调研和分析，准确理解用户对项目的功能、性能和可靠性等方面的具体要求，将用户的非形式的需求表述转化为完整的需求定义，从而确定系统（的功能）必须做什么的过程。

3. 需求分析的目的是什么？

答案:

需求分析的目的是：弄清用户对系统的细节要求，完整、准确、清晰、具体地回答目标系统（“做什么”）应具有哪些功能。通过细致的调查分析，准确地理解用户提出的软件功能、性能等应用问题及其环境的要求。将用户的非形式的需求陈述转化成完整的需求定义，再由需求定义转换成相应需求规格说明。

4. 简述需求分析的必要性及参与角色。

答案:

需求分析的必要性

- 1) 用户与开发者的知识领域不同，产生歧义；
- 2) 软件开发失败 50% 是需求不合理，早期错误易放大。
- 3) 需求阶段的错误会引起错误的放大，后期发现错误要花费等大的代价修改。

需求分析的参与角色是双方的高层技术人员及相关业务人员，开发方参与的人员包括分析师、设计师和架构师，客户方参与的人员包括部门负责人、领域专家和系统用户。

5. 系统分析员的职责有哪些？

答案:

系统分析员的职责包括：

- 1) . 负责需求分析、系统设计，提出解决方案、并编写需求分析说明书和系统设计文档
- 2) . 负责管理、组织需求和系统设计评审

- 3) .负责分析、评估系统体系结构、功能、性能和效益等
- 4) .负责需求变更管理、需求的生命周期管理
- 5) .全方位管理监控项目的开发流程、及时对系统进行调整
- 6) .验收研发部门提交的产品是否满足业务需求
- 7) .配合收集和分析产品市场信息,进行产品北京调查,收集整理产品相关资料
- 8) .配合定义产品功能框架、并承担相应模块的需求规约
- 9) .参与制定项目计划
- 10) .参与系统软件、硬件平台的安装搭建与实施工作
- 11) .参与编制用户文档
- 12) .参与技术咨询、技术支持和培训

6. 软件设计师岗位职责有哪些?

答案:

软件设计师是软件蓝图的绘画者,软件设计师的岗位职责如下:

1. 负责软件的需求分析和软件系统设计;
2. 指导程序员的开发编程工作;
3. 负责软件测试工作,记录测试结果;
4. 对软件进行修改和完善;
5. 编写软件程序设计规格说明书等文档;

7. 软件架构师的职责是什么?

答案:

软件架构师是软件行业中一种新兴职业,工作职责是在一个软件项目开发过程中,将客户的需求转换为规范的开发计划及文本,并制定这个项目的总体架构,指导整个开发团队完成这个计划。主导系统全局分析设计和实施、负责软件构架和关键技术决策的人员。

8. 简述需求工程的概念。

答案:

需求工程是指应用已有技术、方法进行需求分析,确定客户需求,帮助分析人员理解问题并定义目标系统的所有外部特征的一门学科。它通过合适的工具和记号系统地描述待开发系统及其行为特征和相关约束,形成需求文档,并对用户不断变化的需求演进给予支持。

需求工程可分为系统需求工程和软件需求工程。软件需求工程是一门分析并记录软件需求的学科,它把系统需求分解成一些主要的子系统和任务,把这些子系统或任务分配给软件,并通过一系列重复的分析、设计、比较研究、原型开发过程把这些系统需求转换成软件的需求描述和一些性能参数。

需求工程是一个不断反复的需求定义、文档记录、需求演进的过程,并最终在验证的基础上冻结需求。

80 年代,Herb Krasner 定义了需求工程的五阶段生命周期:需求定义和分析、需求决策、形成需求规格、需求实现与验证、需求演进管理。近来,Matthias Jarke 和 Klaus Pohl 提出了三阶段周期的说法:获取、表示和验证。

9. 简述需求工程中包含了那些过程？

答案：

需求工程将需求定义成需求开发和需求管理两个过程。

其中需求开发过程又分成问题获取、需求分析、规格说明和需求验证等四个过程。

①需求获取(Gathering detailed requirements) 需求是指用户对目标软件系统在功能、性能、行为、设计约束等方面的期望。需求获取则是从用户那里得到需求要求。

②需求分析(The analysis phase: Understanding what the customer wants) 使用各种图形符号和文字建立、修改和完善系统模型的过程。使用户和开发者间建立共同语言基础，消除对用户需求在理解上的歧义。

③需求规格说明(Capture the right user requirements with these best practices for writing software specifications) 需求规格说明书是需求分析阶段的最终成果，也是需求分析阶段复审的依据；是领域专家、软件分析师、软件设计师共同交流的途径和媒介；是交付给用户文档的一部份；

④需求验证(Determine user requirements now to avoid problems later) 也称为需求评审。是根据需求说明书，由分析师、设计师、用户共同对需求文档进行评审，主要检查需求的正确性、一致性、完整性和无二义性，并对需求文档进行最终确认的过程。

需求工程中的另一个过程是需求管理过程，需求管理过程则被分成需求确认、需求跟踪和需求变更控制。

① 需求确认：经双方共同开发的需求，需要共同评审和确认，达成共识后要做出书面承诺，使需求文档具有商业合同效力。一旦确认，任何修改都要处于控制之下。

② 需求跟踪：建立需求文档与后续的工作成果之间的对应关系，确保产品依据需求文档进行开发和修改；保证工作成果与需求选项的一致性，从而确保后续成果符合用户需求。

需求跟踪又具有正向和逆向两种形式：

正向跟踪：从需求规格说明到后续工作成果的跟踪。

逆向跟踪：从后续工作成果到需求选项的跟踪。

③ 需求变更控制 变更控制的目的是并不是控制变更的发生，而是对变更进行管理，确保变更有序进行。对于软件开发项目来说，发生变更的环节比较多，因此变更控制显得格外重要。项目中引起变更的因素有两个，一个是来自外部的变更要求，如客户要求修改工作范围和需求等；另一个是来自于开发团队内部的变更要求，如为解决测试中发现的一些错误而修改源码甚至设计。比较而言，难处理的是来自外部的需求变更，因为 IT 项目需求变更的概率大，变更引发的工作量也大，特别是项目后期的需求变更。

1) 目的是防止需求变更失去控制而导致项目混乱。

2) 拒绝不合实际的变更，减少变更带来的风险，防止变更范围扩大、蔓延，杜绝随意的变更

3) 变更控制的流程是：申请、审批、实施和重新确认。

10. 需求的层次有哪些？

答案：

需求的层次有业务需求、用户需求、功能需求和非功能性需求等。

1. 业务需求(business requirement)是反映企业/组织对软件系统的高层次目标要求,即软件系统的建设目标。业务需求通常是“问题定义”或“可行性研究”阶段获取的内容；在需求规格说明书中

反映在项目背景、系统目标或任务概述的描述中。获取的主要对象是客户方的高管、专家、部门负责人。

2. 用户需求(user requirement) 用来描述用户使用产品必须要完成的任务；用户需求可以用业务领域的术语描述，采用开发者与用户都能理解的语言和图形表达。用户需求是经过调查、归纳后双方认同的结果。获取的主要对象是部门负责人、软件的操作者或称终端用户。

3. 功能需求(functional requirement)定义了开发人员必须实现的软件功能，结果在需求规格说明书中；

功能需求用软件行业术语表达，通常是需求建模的结果即目标系统的逻辑模型，如结构化的功能模型、数据模型、行为模型，面向对象的类模型等。

4. 特性(feature)是指一些非功能需求，是满足业务需求的性能要求。如界面的交互性、数据的安全性、数据的事务性、用户的并发性、响应的快速性、操作的实时性、错误与异常的恢复性、软件的容错性等等。

项目的失败或拖延一般不是在功能上，而恰恰倒是在性能要求上，因为这些性能与软件的体系结构有关，与构成系统的网路与硬件环境等底层技术有关，往往超越一般开发人员的技術能力。

11. 简述需求获取的一般方法有哪些？

答案：

1. 需求获取的一般方法有：

①采访、观察、座谈、先前版本测试等。必要时采用快速原型法。

②从用户观点出发，收集用户原始资料，数据、工作方式、工作流程、使用要求，深入到部门、车间、班组，做好原始纪录；

③根据对问题及环境的理解与开发经验，改正用户需求的模糊、歧义和不一致的要求，排除用户的不合理要求，挖掘用户尚未提出但具有价值的潜在需求，使用户需求逐步精确化、一致化和完全化；

④需求获取需要往复进行、逐步深化。

⑤需求获取的内容：写进“需求规格说明书”，确认。

12. 需求获取的策略有哪些？

答案：

需求获取的策略有：

① 循序渐进策略；

② 优先级策略：给需求划分优先级，级别高的先进行重点的调研，这有助于识别出重大的风险，并为制定迭代计划提供指导；

③ 不要陷入技术问题：需求未明确，应回避对技术问题讨论。

④ 挖掘用户的潜在需求。需求的概念是模糊的、笼统的，而且尺度难以把握，预测潜在需求。

⑤ 区分不必要的需求：客户对有些需求提不出来，自然也会提出一些不必要的需求。

13. 需求建模过程

答案：

需求建模过程：

(1) 获得当前系统的物理模型

- (2) 抽象出当前系统的逻辑模型
- (3) 建立目标系统的逻辑模型

14. 需求规格说明的作用有哪些？

答案：

SRS(Software Requirement Specification)是需求分析的最终结果，是软件项目的一个关键性文档，主要用来描述待开发系统所要实现的功能和目标，清楚地阐述一个软件系统必须提供的功能性需求和非功能性需求以及所要考虑的限制条件。

需求规格说明是

- (1) 用户、分析人员和设计人员之间进行理解和交流的手段；
- (2) 开发者与用户之间的技术合同，是制定目标系统测试和项目验收计划的依据；
- (3) 设计和编码的基础，指导着整个系统的开发过程，如果对评审过的 SRS 进行任何改动都需要进行变更控制；
- (4) 制定初步测试计划的基础。
- (5) 编写用户手册的基础。

15. 需求规格说明的质量要求有哪些？

答案：

软件项目的需求规格说明要达到明确(Clear)、完整(Complete)、一致(Consistent)、可测试(Testable)、可跟踪(Traceability)、可修改(Modifiability)等标准。

16. 需求验证的标准有哪些？

答案：

需求验证的标准有正确性、完整性、一致性、可修改性、可跟踪性和可验证性等。

- (1) 正确性：对系统功能、行为、性能等的描述必须与用户的期望相吻合，代表了用户的真正需求。
- (2) 完整性：需求规格说明应该包括软件要完成的全部任务，不能遗漏任何必要的需求信息，注重用户的任务而不是系统的功能将有助于你避免不完整性
- (3) 一致性：需求规格说明对各种需求的描述不能存在矛盾，如术语使用冲突、功能和行为特性方面的矛盾以及时序上的不一致等。
- (4) 可修改性：格式和组织方式应保证后续的修改能够比较容易和协调一致。我们可以使用软件工具，或者使用目录表、索引和相互参照列表等方法使软件需求规格说明更容易修改。
- (5) 可跟踪性：可跟踪性意味着每项需求都能与其对应的来源、设计、源代码和测试用例联系起来。
- (6) 可验证性：描述的需求都可以运用一些可行的手段对其进行验证和确认。

17. 需求评审的方法有哪些？

答案：

按照评审人员可划分成用户评审和同行评审，按照评审方式可分为正式与非正式评审，编写设计初始测试用例和初始用户手册等辅助文档也可以用于需求评审。

18. 简述需求变更的概念，它的危害有哪些？

答案：

需求变更是指在软件开发过程中对需求的修改。需求变更的危害包括：增加开发时间、影响项目进度、降低系统质量、严重的会导致项目的失败。

19. 简述需求变更的概念。

答案：

需求变更管理是软件维护的一项任务，也是配置管理的重要内容，是现代软件工程的重要体现。

需求变更管理的目的是在用户和开发人员之间建立对需求的共同理解，维护需求与软件工作成果的一致性，并控制需求的变更。

20. 需求变更管理的主要活动有哪些？

答案：

需求变更管理的主要活动有

- ① 需求确认：经双方共同开发的需求，需要共同评审和确认，达成共识后要做出书面承诺，使需求文档具有商业合同效力。一旦确认，任何修改都要处于控制之下。
- ② 需求跟踪：建立需求文档与后续的工作成果之间的对应关系，确保产品依据需求文档进行开发和修改；保证工作成果与需求选项的一致性，从而确保后续成果符合用户需求。
- ③ 需求变更控制；
 - 1) 目的是防止需求变更失去控制而导致项目混乱。
 - 2) 拒绝不合实际的变更，减少变更带来的风险，防止变更范围扩大、蔓延，杜绝随意的变更
 - 3) 变更控制的流程是：申请、审批、实施和重新确认。

第五章

一. 简答题（共 12 题, 79.2 分）

1. 简述抽象与逐步求精的概念。

答案：

抽象是一种思维方法，指导人们在问题的某个层次上考虑问题，而忽略那些低层次的细节；同时抽象也是一种对某个（或某些）客观事物的概括性描述。例如学生就是对所有学生的概括性描述。
求精是降解抽象的层次，就是向具体化的过渡过程。逐步求精是自顶向下、逐层分解、分而治之来解决复杂问题的思想方法。求精的每一步都是用更为详细的描述替代上一层次的抽象描述；细化的本质是分解，表现就是分层。在整个设计过程中用模块组成系统的层次结构，采用模块化层层分解。

2. 按照软件的生命周期，人们对软件的抽象层次有哪些？

答案：

问题定义：软件定义

可行性分析：可行性分析报告

需求分析：业务需求、用户需求、功能需求、非功能需求

设计：概要设计、详细设计

实现：程序代码、软件系统

维护：更新的软件系统

3. 软件开发过程中，设计阶段的输入和输出是什么？

答案：

设计阶段的输入是需求规格说明，输出是软件的设计文档，包括软件的概要设计和详细设计。

4. 软件设计的总体任务有哪些？

答案：

软件设计的总体任务包括系统的结构设计和详细设计。系统结构设计 (architectural design) 也叫总体 (general) 设计或概要 (preliminary) 设计，又分为体系结构设计和详细设计两部分。

体系结构设计是支撑和管理软件运行的环境设计。由于现代的软件是处在操作系统、网络、各种服务器共同搭建的环境下运行，并且具有并发、安全、事务等多方面的管理，是软件设计优先考虑的问题。领域问题结构设计是满足需求的软件功能设计，是软件设计的核心任务。其内容是将领域问题的分析模型细化成软件结构模型，也就是划分软件的模块结构及确定模块之间的关系。

系统的过程设计 (procedural design) 也叫详细 (detail) 设计。

又分为过程和数据结构、接口和数据库设计三个部分。

过程和数据结构设计是对模块内部的过程和数据结构进行设计。也就是对模块内进行算法分析和程序设计。

接口设计包括人机交互界面的设计和与外部系统之间的接口设计。数据设计主要是对数据库的逻辑设计和物理设计。

概要设计是根据需求确定软件和数据的总体框架；详细设计是在概要设计的基础上完成模块的算法和数据结构设计。

5. 简述模块的概念，模块有哪些外特性和内部特性？

答案：

模块是完成特定功能的程序实体，可以单独命名且可以通过名字来访问。如汇编语言中的子程序，Pascal 语言中的过程，C 语言中的函数和宏等都是模块，面向对象方法学中的类也是模块。模块是构成程序的基本构件。

模块具有输入和输出(参数传递)、功能、内部数据结构(局部变量)和程序代码四个特性。其中输入、输出和功能是模块的外部特性；内部数据结构和程序代码是模块的内部特性。软件结构设计仅需要了解该模块的外部特性。

6. 简述模块化的概念，简述模块化设计的优点？

答案：

模块化是指解决一个复杂问题时自顶向下逐层分解成若干模块的过程。每个模块完成一个特定的子功能，所有模块按系统结构组装起来，完成整个系统所要求的功能。

模块化设计是将一个大型软件划分为一个个较小的、相对独立但又相关的模块。模块化设计的优点：

- 1 将问题简化，划分了模块,使每一个模块,完成单一功能；
- 2 可以独立地进行模块的编码测试，便于软件开发工作的组织；
- 3 把每一个模块要解决的问题局限在有限的范围，减少出错机会；
- 4 便于纠错；
- 5 便于对特定的模块进行优化处理；
- 6 一个模块多次重复使用,利于提高软件产品的利用率；
- 7 程序易于理解；
- ⑧ 利于估计工作量和开发成本。

7. 简述信息隐藏原理的概念。

答案：

信息隐藏原理：模块应该设计得使其所含信息(过程和数据)对于那些不需要这些信息的模块不可访问；每个模块只完成一个相对独立的特定功能；模块之间仅仅交换那些为完成系统功能必须交换的信息，即模块应该独立。显然，模块独立的概念是模块化、抽象、信息隐藏和局部化等诸多概念的直接结果。

信息隐藏不仅支持模块的并行开发，而且还可减少测试和后期维护的工作量。因为测试和维护阶段不可避免地要修改设计和代码，模块对大多数数据和过程处理细节的隐藏可以减少错误向外传播。此外，整个系统欲扩充功能亦只需“插入”新模块，原有的多数模块无须改动。

8. 简述模块独立性的概念，模块独立性的衡量标准有哪些？

答案：

1. 模块独立的含义是指每个模块都要完成独立的功能，符合信息隐藏和信息局部化原则，模块间关联和依赖程度应尽可能地小。

2. 衡量软件独立性的两个度量标准是模块的内聚度和耦合度。内聚度是一个模块内部各个元素间（语句和程序段）彼此的紧密程度的度量。耦合度是指软件结构中各模块间相互联系紧密程度的一种度量。

9. 简述模块的内聚度(cohesion)概念，简述模块内聚度的七个等级。

答案：

模块的内聚度(cohesion)是标志一个模块内部各成分彼此结合的紧密程度。内聚度按其高低程度可分为七个等级，高内聚度模块独立性强，设计时应尽可能提高模块内聚度。

模块内聚度的七个等级：

- 1 偶然内聚：如果一个模块的各成分之间毫无关系，则称为偶然内聚。这样的模块可以完成一组任务，但这些任务之间的关系松散，实际上没有什么联系。这些成分存放在一个模块内是偶然的。
- 2 逻辑内聚：几个逻辑上相关的功能被放在同一模块中，则称为逻辑内聚。尽管逻辑内聚比偶然内聚合理一些，但逻辑内聚的模块各成分在功能上并无关系，即使局部功能的修改有时也会影响全局，因此这类模块的修改也比较困难。
- 3 时间内聚：模块内部的各个功能在同一时间段内执行。即这些功能放在同一个模块中仅仅是因为与时间相关的因素。
- 4 过程内聚：模块内部的各个功能是相关的，且必须按特定的顺序执行。所有的这些功能由同一个控制流连接在一起。
- 5 通讯内聚：如果一个模块的所有成分都操作同一数据集或生成同一数据集，则称为通信内聚。
- 6 顺序内聚：如果一个模块的各个成分和同一个功能密切相关，而且一个成分的输出作为另一个成分的输入，则称为顺序内聚。
- 7 功能内聚：模块的所有成分对于完成单一的功能都是必须的，则称为功能内聚。

10. 模块的耦合度(coupling)的概念，简述耦合度的七个等级。

答案：

模块的耦合度(coupling)是对软件结构中模块间关联程度的一种度量，耦合的强弱取决于模块间接口的复杂性、进入或调用模块的位置以及通过接口界面传送数据的多少等。在设计软件时应追求尽可能松散耦合的系统。

模块间的耦合程度直接影响系统的可理解性、可测试性、可靠性和可维护性。

耦合度的七个等级：

- (1) 内容耦合。当一个模块直接修改或操作另一个模块的数据, 或者直接转入另一个模块时, 就发生了内容耦合。此时, 被修改的模块完全依赖于修改它的模块。
- (2) 公共耦合。两个以上的模块共同引用一个全局数据项就称为公共耦合。
- (3) 外部耦合。若一组模块都访问同一全局数据项, 则称为外部耦合。
- (4) 控制耦合。一个模块在界面上传递一个信号（如开关值、标志量等）控制另一个模块, 接收信号的模块的动作根据信号值进行调整, 称为控制耦合。
- (5) 标记耦合。模块间通过参数传递复杂的内部数据结构, 称为标记耦合。此数据结构的变化将使相关的模块发生变化。
- (6) 数据耦合。模块间通过参数传递基本类型的数据, 称为数据耦合。
- (7) 非直接耦合。模块间没有信息传递时, 属于非直接耦合。

11. 模块优化的一些原则。

答案：

- 1 改进软件结构，提高模块独立性 在对初步模块进行合并、分解和移动的优化过程中力求提高模块的内聚，降低耦合。
- 2 模块大小要适中 过大的模块应分解以提高理解性和可维护性；过小的模块会使模块之间关系复杂，可合并到上级模块中。
- 3 软件结构图的深度、宽度、扇入和扇出要适当。
如扇出大，容易使结构扁平宽度过宽，若扇出小可能造成深度过深。
- 4 模块的作用域应该在控制域之内。
- 5 设计单入口单出口的模块。
- 6 降低模块接口的复杂程度，接口间的信息传递尽量要简单，为执行模块的功能所用。避免传递相互间没有联系的数据。

12. 简述详细设计的表达方式有哪些？

答案：

1. 伪代码(Pseudo code)
2. 程序流程图
3. 盒图(N-S) (Nassi 和 Shneiderman)
4. PAD 图(PAD-Problem Analysis Diagram)
5. 判定表
6. 判定树

二. 计算题（共 3 题, 20.8 分）

1.

已知冒泡排序的过程如下：

- 1) 比较相邻的元素，如果前一个比后一个大，就把它们两个调换位置。
- 2) 对每一对相邻元素作同样的工作，从开始第一对到结尾的最后一对。这步做完后，最后的元素会是最大的数。
- 3) 针对所有的元素重复以上的步骤，除了最后一个。
- 4) 持续每次对越来越少的元素重复上面的步骤，直到没有任何一对数字需要比较。

分别使用程序流程图、NS 图和 PAD 图描述冒泡排序算法。

答案：

设待排序的数据定义如下：

```
typedef struct{
    int key
    Data data
} Datatype;
Datatype A[1..n];
1 C 语言排序算法如下：
void bubble_sort(Datatype A[], int n)
//n 为数组 A 的元素个数
{
    //一定进行 N-1 轮比较
    for(int i=0; i<n-1; i++)
```

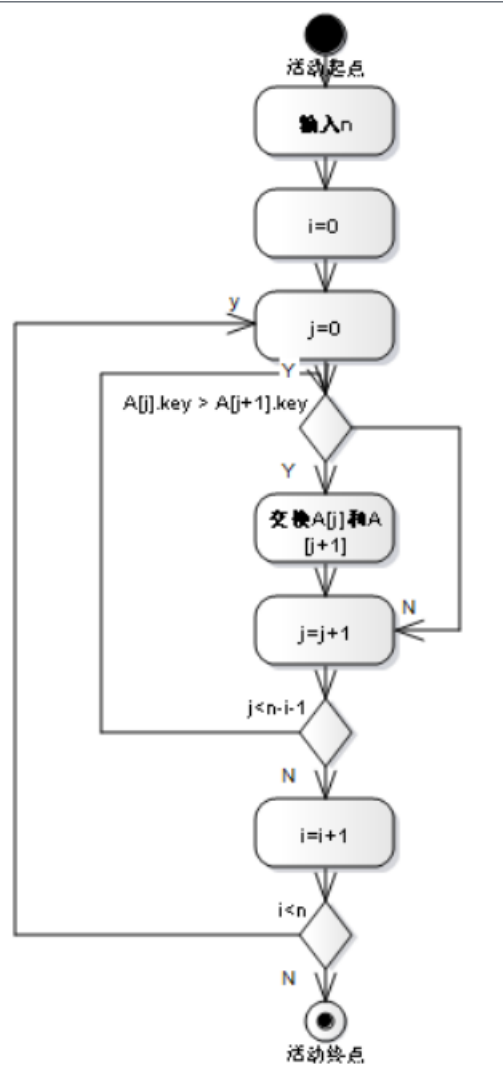


```

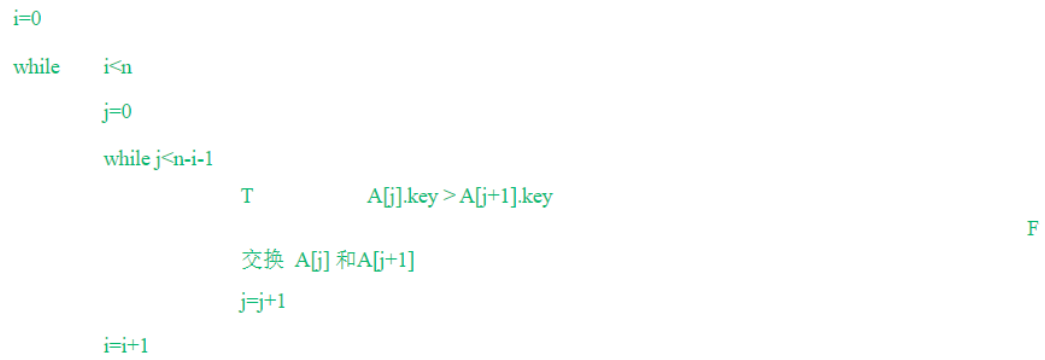
{
    //每一轮比较前 n-1-i 个，即已排序好的最后 i 个不用比较
    for(int j=0; j<n-1-i; j++)
    {
        if(A[j].key > A[j+1].key)
        {
            Datatype temp = D[j];
            A[j] = A[j+1];
            A[j+1]=temp;
        }
    }
}
}

```

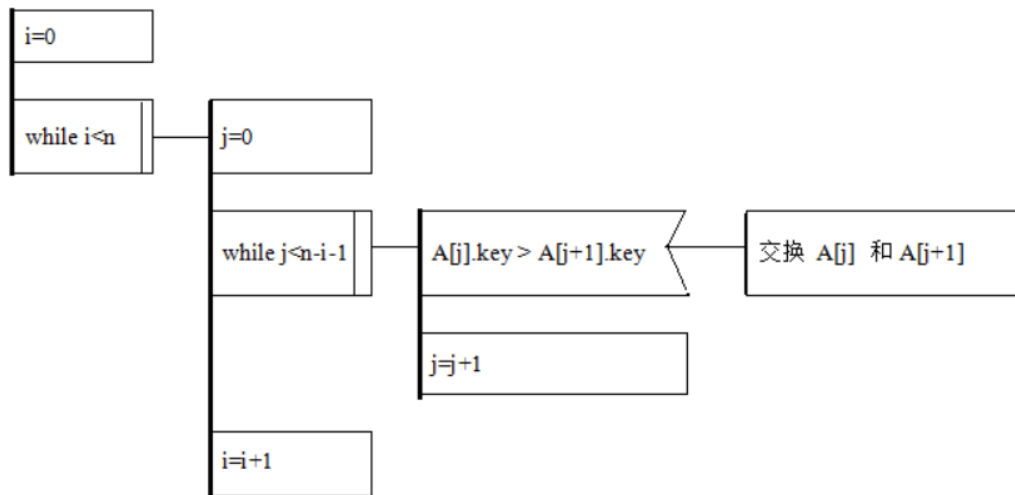
2 程序流程图如下



3 NS图



4. pad 图



3. 已知，某航空公司对满足不同条件的乘客规定了不同的行李托运费方法。

乘客可以免费托运重量不超过 30kg 的行李。当行李重量超过 30kg 时，对头等舱的国内乘客超重部分每公斤收费 4 元，对其他舱的国内乘客超重部分每公斤收费 6 元，对外国乘客超重部分每公斤收费比国内乘客多一倍，对残疾乘客超重部分每公斤收费比正常乘客少一半。假设 W 表示行李重量，请填写下列表格给出该问题的判定表表示；

行李托运计费方法决策表										
决策编号		1	2	3	4	5	6	7	8	9
条件	W≤30	T	F	F	F	F	F	F	F	F
	国内乘客		T	T	T	T	F	F	F	F
	头等仓乘客		T	F	T	F	T	F	T	F
	残疾乘客		F	F	T	T	F	F	T	T
决策	托运费	(W-30)*2								
		(W-30)*3								
		(W-30)*4								
		(W-30)*6								
		(W-30)*8								
		(W-30)*12								
		0								

答案：

行李托运计费方法决策表										
决策编号		1	2	3	4	5	6	7	8	9
条件	W≤30	T	F	F	F	F	F	F	F	F
	国内乘客		T	T	T	T	F	F	F	F
	头等仓乘客		T	F	T	F	T	F	T	F
	残疾乘客		F	F	T	T	F	F	T	T
决策	托运费	(W-30)*2			X					
		(W-30)*3				X				
		(W-30)*4		X					X	
		(W-30)*6			X					X
		(W-30)*8					X			
		(W-30)*12						X		
		0	X							

4. 已知，某航空公司规定的行李托运计费方法如下。

乘客可以免费托运重量不超过 30kg 的行李。当行李重量超过 30kg 时，对头等舱的国内乘客超重部分每公斤收费 4 元，对其他舱的国内乘客超重部分每公斤收费 6 元，对外国乘客超重部分每公斤收费比国内乘客多一倍，对残疾乘客超重部分每公斤收费比正常乘客少一半。假设 W 表示行李重量，请用判断树描述这个托运费计算方法。

答案：



第六章

一. 简答题（共 5 题, 62.5 分）

1. 简述结构化分析方法的步骤有哪些？

答案：

简述结构化分析方法的步骤有：

- 1、建立当前系统的“具体模型”。
- 2、抽象出当前系统的逻辑模型。
- 3、建立目标系统的逻辑模型。
- 4、为了对目标系统做完整的描述，还需要考虑人机界面和其他一些问题。

2. 简述数据流图的概念。

答案：

数据流图（Data Flow Diagram, DFD）是描述系统中数据流程的图形工具，它标识了一个系统的逻辑输入和逻辑输出，以及把逻辑输入转换为逻辑输出所需的加工处理。数据流图是对系统的一种抽象描述，结构化分析方法就是通过数据流图建立系统的抽象描述，再以这个抽象为基础进行分解得到一个更具体的描述，继续分解，知道得到一个比较满意的描述。并将这个最终描述作为系统的逻辑模型。

3. 简述数据流图的构成符号有哪些？

答案：

数据流图使用数据流、外部实体、加工和数据存储等四种符号。其中

- (1) 数据流 数据流是数据在系统内传播的路径，由一组成分固定的数据组成。由于数据流是流动中的数据，所以必须有来源、去向和流向，数据流应该用名词或名词短语命名。
- (2) 加工(又称为数据处理)。对数据流进行某些操作或变换。每个加工也要有名字，通常是动词短语，简明地描述完成什么加工。在分层的数据流图中，加工还应编号。
- (3) 数据存储(又称为文件)，指保存在某种介质上的数据，它可以是数据库文件或任何形式的数据组织。
- (4) 外部实体，是系统外的实体(包括人员、组织或其他软件系统)，统称外部实体。他们于本系统有数据交换关系。

4. 绘制数据流图的基本方法

答案：

绘制数据流图的基本方法是自顶向下、逐层分解的方法。即

- 1 先确定系统范围，画出顶层的 DFD 图。
- 2 逐层分解顶层 DFD 图，获得若干中间层 DFD 图，直到不能或不需要进一步分解为止。
- ③ 画出底层的 DFD 图。

5. 绘制数据流图的基本原则

答案：父图与子图的平衡、子图内的平衡、数据守恒原则、守恒加工原则

二. 论述题（共 1 题, 12.5 分）

1. 综合第四、五六三章的内容，总结一下结构化方法分析和设计软件的基本过程，以及这个过程中使用的技术和方法。

答案：

结构化方法分析和设计软件的基本过程，以及这个过程中使用的技术和方法如下。

1 结构化分析过程

- 1) 建立当前系统的物理模型，可使用系统流程图。
- 2) 抽象出当前系统的逻辑模型，使用数据流图和数据字典。
- 3) 建立目标系统的逻辑模型，使用数据流图和数据字典。
- 4) 考虑人机界面和其他一些问题，如系统需要满足的非功能属性。

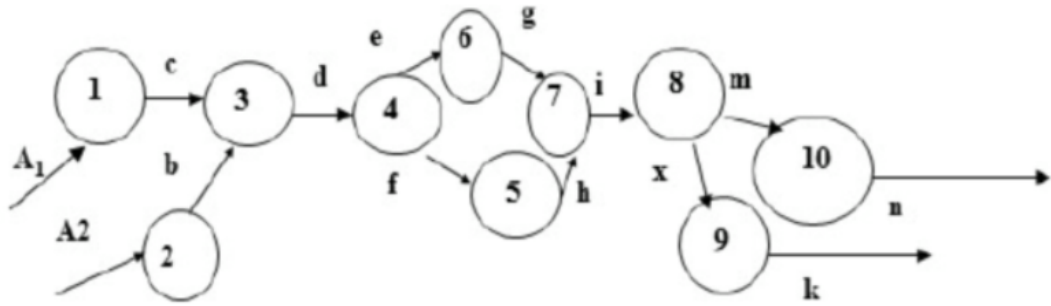
2 结构化设计过程

- 1) 概要设计 设计系统的结构，使用模块结构图表示，使用的主要技术是将数据流图转换成模块结构图。
- 2) 详细设计 依据模块结构图完成系统的每个模块的设计，包括输入输出设计、数据设计和处理逻辑设计。

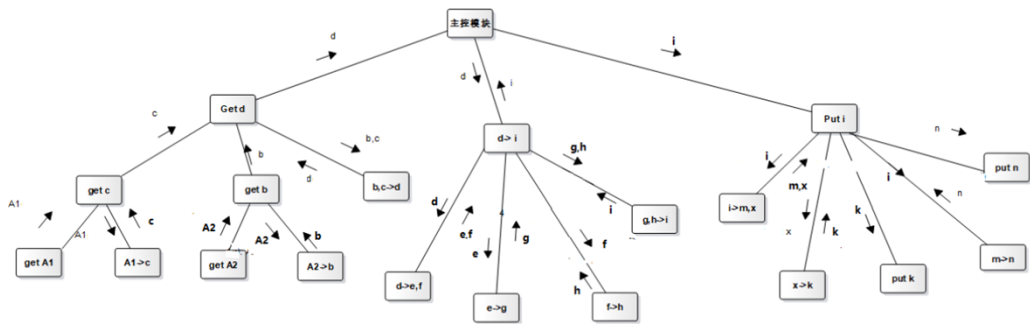
使用的工具包括人机界面模型、er 图模型和详细设计描述工具，例如伪代码、流程图、盒图、PAD 图、判断表和决策树等。

三. 计算题（共 2 题, 25.0 分）

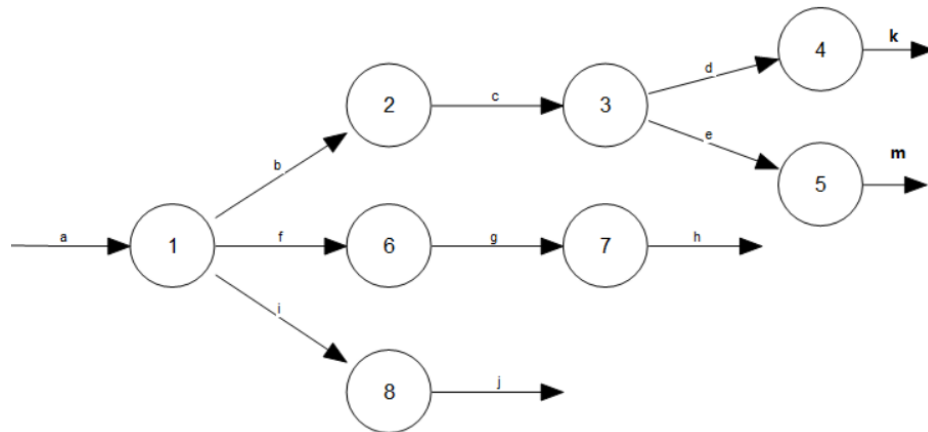
1. 已知一个数据流图如下，将下面的数据流图转换成模块结构图。



答案:



2. 已知一个数据流图如下，将下面的数据流图转换成模块结构图。



答案:

