

03173 《软件开发工具》

考前资料

考试题型及分值分布

类别	题型	题量	分值/题	总分
选择题	单项选择题	20 题	1 分	20 分
非选择题	填空题	20 题	1 分	20 分
	简答题	6 题	5 分	30 分
	论述题	1 题	10 分	10 分
	应用题	1 题	20 分	20 分

注：本资料题型根据考试大纲或历年真题进行预测，仅供复习参考

第一章 绪论

★★考点 1：软件开发工具的由来（单选、填空、简答）

(1) 软件开发工具的范围：在高级程序设计语言（第三代语言）的基础上，为提高软件开发的质量和效率，从规划、分析、设计、测试、文档和管理等各方面，对软件开发提供各种不同程度的帮助的一类新型的软件

(2) 首先，它本身就是一种软件，通俗地说，它是开发软件用的软件。其次，它是在第三代语言的基础上发展起来的。第三代语言在这里指已经形成的软件开发技术。

★考点 2：软件开发工具的演变过程（单选、填空）

(1) 第一步是汇编语言，即第二代语言的出现。针对难以记忆的、无意义的、二进制的字符串，用在英语中具有一定意义的单词来代替它，这就是所谓“助记忆码”，或汇编码。用汇编码编写的程序称为汇编程序，而把这些汇编指令转化为机器指令的程序则称为汇编系统。

(2) 当电子计算机诞生时，人们面对的是只能执行机器指令的硬件设备，即所谓“裸机”

(3) 用通用软件帮助软件开发有三个主要的弱点。①是有许多工作是通用软件所无法完成的；②是用通用软件完成某些工作，只能表现出其表面的形式，而不能反映其逻辑内涵；③是用通用软件来帮助人们完成软件开发工作时，常常遇到难以保持一致性的困难。

(4) 软件开发工具发展的主要特点：第一个特点是面向网络，不仅基于网络，而且用于网络。另一个特点是开源软件的兴起和运用。

考点 3：软件开发工具的特性（单选）

(1) 硬件与软件的区别

①软件的实质：软件这个名词是有了计算机之后才产生的，而硬件则是自古就有的。

②硬件提供的是信息存储与处理的基础。软件一端连着计算机硬件，向硬件提供可以执行的机器指令，另一端面向用户，接受用户提出的要求和算法。从这个意义上说，软件是用户与硬件之间的桥梁。

(2) 20 世纪 90 年代，软件开发进入了大量应用软件开发工具的阶段，进一步扩大了软件开发的范围。

考点 4：软件开发工作的发展变化（填空）

软件开发工具及新的软件开发方法的发展主要表现在四个方面：自动化程度的提高；软件开发过程进一步向用户方面延伸；把软件开发工作延伸到项目及版本管理；吸收了许多管理科学的内容与方法。

卡内基·梅隆大学的软件工程研究所提出的能力成熟度模型，作为衡量软件开发项目组的标准。（填空）

★★考点 5：软件开发的过程（单选、简答）

(1) 软件开发工作的起点是初始要求的提出。软件开发工作首要的任务是需求分析。

(2) 第二阶段是总体设计。它的任务是根据软件功能说明书的要求，完成软件的总体设计，这包括整个软件的结构设计，公用的数据文件或数据库的设计，各部分的连接方式及信息交换的标准等主要内容。

结构设计是把软件划分成若干个模块，指定每个模块的功能要求，以及它们之间的相互关系（调用关系、参数传递关系或信息共享关系）。总体设计的成果是系统的总体设计文件及各个模块的设计任务书。总体设计文件应包括结构图、模块清单、公用数据结构。

(3) 程序的编写与文档的编写是两件并行的工作，统称之为实现阶段。

(4) 第四个阶段是测试或调试阶段。包括模块的调试与整个软件的联调两个部分。

★考点 6：软件开发工具的功能要求（填空）

软件开发工具应提供的各类支持工作：

(1) 认识与描述客观系统。主要用于需求分析阶段；(2) 存储及管理开发过程中的信息；

(3) 代码的编写或生成；(4) 文档的编制或生成；

(5) 软件项目的管理，项目管理包括进度管理、资源与费用管理、质量管理三个基本内容。

★★考点 7：软件开发工具的性能（单选、简答）

(1) 表达能力或描述能力； (2) 保持信息一致性的能力； (3) 使用的方便程度；

(4) 工具的可靠程度； (5) 对硬件和软件环境的要求。

★考点 8：软件开发工具的类型（单选）

(1) 按工作阶段划分

把软件开发工具分为三类：计划工具、分析工具、设计工具。

(2) 按集成程度划分

集成化的软件开发工具也常常被称为软件工作环境。

(3) 按与硬件、软件的关系划分

有的软件开发工具依赖于特定的计算机或特定的软件，有的软件开发工具则是独立于硬件与其他软件的，与工具自身的情况有关。

数据字典管理系统属于分析工具。（单选）

帮助人们绘制数据流程图的工具 FLOW 属于分析工具。（单选）

为项目主管人员服务的软件开发工具是计划工具。（单选）

第二章 软件开发过程及其组织

★考点 1：软件开发的基本问题（填空）

(1) 从表面上看，是程序总有错，总不符合要求，关键在于两个转换没有顺利地通过：首先是从用户的理解到程序员的理解，其次是从程序员的理解到程序的实现。

(2) 在软件开发过程中，程序员必须花相当的时间尽可能地了解与熟悉应用领域，并且正确地理解用户工作中的信息需求和信息流程。人们逐渐认识到程序员和用户之间的沟通与交流的重要性，把软件的概念从单纯的程序扩展到程序加文档。

考点 2：大型软件开发中的困难（单选）

(1) 一致性的保持成为十分困难的问题；

(2) 测试的困难大大增加；

(3) 工作进度难以控制；

(4) 文档与代码的协调十分困难。文档包括功能说明书、总体设计、各模块的设计任务书、每个部分的研制报告、测试报告、直到最终的验收报告和给使用者的用户手册。

(5) 版本更新带来的困难。

大型软件开发项目管理不同于一般工程项目管理的特点是有版本更新问题。（单选）

考点 3：困难产生的原因

(1) 来自大系统的复杂性。人类对于大量元素组成的复杂系统，至今尚没有理想的描述与控制方法。

(2) 许多具有主动性的个人之间的组织与协调带来大量的困难。当组织许多人共同完成一项大的任务时，分工和协调就成为关键因素。

(3) 各个应用领域之间的差别导致困难的加重。软件作为应用领域与硬件之间的桥梁，一端是固定的，另一端却是变化无穷的。

(4) 时间的因素，变化的因素给软件开发工作带来许多困难。

★★★考点 4：软件开发方法的发展（单选、填空、简答）

(1) 从开发的方法而言，就先后提出了结构化程序设计，软件工程方法，面向对象的程序设计方法，即插即用的程序设计方法等。

(2) 大型软件开发过程中涉及了用户、项目负责人（或软件架构师）、程序员和硬件四个角色。

(3) 用户承担的工作为四项：提出需求、验收、使用、要求修改。

(4) 项目负责人在开发过程中起着十分关键的作用。

★★★考点 5：结构化程序设计方法（单选、填空、简答）

(1) 程序的结构分解成三种基本模块：处理单元，循环机制，二分决策机制。问题的关键就在于合理地划分模块，并保证模块符合“结构良好”的要求。

(2) 模块的划分应当符合以下三条基本的要求。

- ①模块的功能在逻辑上尽可能地单一化、明确化，最好做到一一对应；
- ②模块之间的联系及互相影响尽可能地少；
- ③模块的规模应当足够小，以便使它本身的调试易于进行。

(3) 结构化程序设计的方法主要是为程序员服务的。

在软件系统的模块结构图中，一个模块的右上角有一个小圆圈，该记号表示选择调用。(单选)

模块之间的联系及互相影响应尽可能少，即应当尽量避免逻辑耦合，而仅限于数据耦合。(填空)

结构化程序设计的关键在于合理地划分模块，并保证其符合“结构良好”的要求。(填空)

★★考点 6：软件工程方法（单选、论述）

(1) 由于软件危机的发生，要想大幅度地，从根本上提高软件开发的效率和质量，就应当吸取人们在各种传统产业中的成功经验，从组织和管理的角度加强力量，使软件生产从程序员的个人劳动提高成为有组织的、可控制的工程，这就产生了软件工程。

(2) 应用软件的开发过程

- ①需求分析；②分析与设计阶段；③编程阶段；④测试阶段；⑤使用及维护阶段。

(3) 在标准化方面，人们主要从两个方面去探索，一方面是表达方式的标准化；另一方面是对工作的质量及检查制订标准，基于上述工作过程的模型，设置一系列检查点，定期进行一定的检查。

★★★考点 7：面向对象的程序设计方法（单选、填空、简答）

(1) 面向对象的程序设计的基本思想

①客观世界的任何事物都是对象，它们都有一些静态属性（相应于数据结构），也都是一些相关的操作（相应于程序模块）。封装性：作为一个整体，对象对外不必公开这些属性与操作。

②对象之间有抽象与具体，群体与个体，整体与部分等几种关系。

③遗传性：抽象的、较大的对象所具有的性质，包括静态属性和动态操作，自然地成为它的子类的性质，不必加以说明或规定。

④对象之间可以互送消息。这消息可以是传送一个参数，也可以是使这个对象开始某个操作。

(2) 在认识的无穷的、阶梯式的层次结构中，原则上每一层都可以作为观察的基点，认识的方法既不是笼统地自顶向下，也不是笼统地自底向上，而是在于合理地选择层次。

(3) 面向对象的程序设计之所以能产生巨大的影响，其根本原因在于它提供了认识框架。也正因为这样，认识框架迅速地散布到程序设计语言的范围之外，以至出现了面向对象的系统分析，面向对象的系统设计，面向对象的数据库管理系统等。

考点 8：即插即用的程序设计方法（填空）

(1) 即插即用的程序设计是近年来软件界讨论的又一个热门话题，与它紧密相关的还有组件程序设计，组合文件，宏观程序设计等。

(2) 即插即用的程序设计是在面向对象的程序设计方法的基础上提出来的。

考点 9：面向开源软件的程序设计方法

使用开源代码并不一定能够减轻工作量。实际上，这对于软件开发人员，特别是项目负责人或软件系统的架构师来说，要求更高、更难。至少在目前阶段，特别是在有关键使命的软件的开发中，利用开源软件，很难达到功能和效率的要求所以需要谨慎从事。

★★考点 10：好软件的特征（单选、简答）

- (1) 正确地实现所要求的功能，准确地给出预定的输出结果。
- (2) 用户界面友好，符合实际用户的使用习惯与知识能力。
- (3) 具有足够的速度（而不是越快越好），能在符合用户要求的时间限度内，给出所要求的处理结果。
- (4) 具有足够的可靠性，能够在各种干扰下保持正常的工作。
- (5) 程序易读，结构良好，文档齐全，从而保证系统易于修改。

★考点 11：一个好的项目组应当具备的条件（简答）

- (1) 有严格的、成文的工作规范和文档标准，而且应当为全体成员所熟知，并且切实得到遵守。
- (2) 人员之间有严格的分工，除了程序员之外，必须有专门的秘书。
- (3) 每个项目都要事先制定详细的时间表，并且得到严格执行。
- (4) 工作进度难以控制。

第三章 软件开发工具的理论基础

★★考点 1：软件开发过程中涉及的信息（单选）

- (1) 有关系统环境、现状及需求的信息。这类信息由用户提出，由分析人员采集。
- (2) 有关软件的功能设计与物理设计的各种信息。这类信息是由分析人员根据需求分析而形成的。
- (3) 软件成果本身，包括程序与文档。由程序员根据设计方案，依据某种计算机语言编制出来的。
- (4) 用户对系统的各种变更要求，以及系统的各种变更的记录。这类信息是用户在使用过程中逐步形成的，由维护人员收集、整理，并且实施必要的修改。

前三类信息是在一个开发周期内而言的，而第四类信息是跨开发周期的。

★考点 2：信息管理工作（简答）

用到的理论和方法很多，其中比较重要的有以下六项。

- (1) 认知科学中关于概念模式的概念与方法。
- (2) 数据库技术的理论与方法。
- (3) 编译技术的有关方法。
- (4) 关于人机界面的理论与方法。
- (5) 管理科学中关于项目管理与版本管理的理论与方法。
- (6) 系统科学与系统工程中的有关理论与方法。

考点 3：概念模式及其作用（填空）

(1) 概念模式，也译为概念模型，指人们在认识事物过程中，对于某一事物或某一系统形成的、抽象的、一般化的框架。

(2) 概念模型的重要性

- ①是人们认识客观世界的一种方法、一种工具。
- ②是人们在长期的实践活动中逐渐形成的。
- ③在认识中的作用十分重要，同时作为交流与表达工具也是十分重要的。
- ④该意义决定了它的相对性，即面对各种不同的用途，必然会有不同形式的概念模式。

★考点 4：框图（单选、填空）

框图是人们在编写软件时最早使用的一种概念模式，用来描述程序执行的逻辑过程的。它把程序的基本步骤归纳为处理、判断、输入输出、起始或终结等几个基本功能，并用不同的记号加以表示。用箭头表示控制或执行的顺序。

考点 5：结构图（单选）

结构图用以表示大型软件的层次结构，即模块结构，它的基本观点是，程序的调用方式有三种：顺序调用、选择调用、循环调用。结构图的好处是能够体现层次观点，由粗到细、自顶向下地描述程序。

★★考点 6: 数据流程图 (填空、简答)

数据流程图面对的是一个系统的信息流程。它更多地用于描述某一业务处理系统的信息来源、存储、处理、去向的全面情况。它的基本思想是把信息流看做一个组织或系统运作的线索, 力图简明扼要地勾画出全局。数据流程图的基本元素是外部实体 (即系统以外的信息来源或去向)、数据处理与数据存储。用箭头表明信息在它们之间的流动状况。

考点 7: 实体关系图 (填空)

实体关系图 (简称 E-R 图) 是一种用于描述静态数据结构的概念模式。以实体、关系、属性三个基本概念概括数据的基本结构。

考点 8: 数据字典

数据字典也是一种描述数据内容的概念模式。用表格的形式列出数据的基本属性以及相互关系,

★★考点 9: 时序网络 (单选、填空)

时序网络是一种较为特殊的概念模式。它主要描述系统状态及其转换方式, 因此常常用于一些实时控制方面的软件的功能描述。它的基本概念是状态与转换。状态指系统在运转中某一特定的形态或工作方式, 而转换指状态之间在一定条件下的相互变化。

考点 10: 数学与逻辑模型 (单选)

- (1) 数学与逻辑模型的最大优点是严格、准确、可计算
- (2) 作为逻辑模型的表达方式, 决策树和决策表也是经常使用的

考点 11: 计算机模拟模型

计算机模拟是一种强有力的概念模式。它利用计算机大量、高速处理信息的能力, 在计算机内设置一定的环境 (如资源条件等), 又以程序来实现客观系统中的某些规律或规则, 在二者基础上, 计算机就可以高速运行, 以便人们观察与预测客观系统的状况。

★★考点 12: 信息库 (填空、论述)

(1) 信息库无非就是数据库。数据库是在文件系统基础上发展起来的一种数据管理技术, 它的重点在于合理地组织数据, 体现出数据之间的逻辑关系, 并尽可能地帮助用户检索, 减少冗余。

(2) 数据仓库的基本思想是: 现实中的大量原始数据与实际的应用是有相当距离的, 为了能够切实为企业管理者提出对管理真正有用的信息, 就要对信息或数据进行多次的加工或提炼, 从而得出许多派生的或综合的数据。

★★考点 13: 信息库研究三大方面 (填空、简答)

(1) 信息库的内容

①所述软件的工作环境、功能需求、性能要求、有关的各种信息来源的状况、用户状况、硬件环境以及在该专业领域中的作用等外部信息;

②需求分析阶段中收集的有关用户的各种信息;

③逻辑设计阶段的各种调查材料和由此生成的各种文档;

- ④设计阶段的各种资料;
- ⑤编程阶段的所有成果;
- ⑥运行及使用情况的详细记录;
- ⑦维护及修改的情况;
- ⑧项目管理的有关信息、人员变更、资金投入、进度计划及实施情况。

(2) 信息库应当具备哪些管理功能。

(3) 如何保持一致性。主要的困难在于, 复杂的、大型的软件系统不是由一个程序模块组成的, 一个模块的更新往往带来其他模块的相应变更, 并不是一个模块更新后就能够自然而然地与其他模块一起构成一个新的版本。

考点 14: 人机界面的研究

(1) 对于用户, 界面就是系统本身,

(2) 人机界面的研究有两种角度: ①从软件开发的角度; ②归纳出人机界面的基本要素与基本操作

★★★考点 15: 用户界面的基本原则 (填空、论述)

(1) 用户界面的基本原则

- ①用户界面的主要功能是通信。关键要使人 与计算机之间能够准确地交流信息;
- ②用户界面必须始终一致;
- ③用户界面必须使用户随时掌握任务的进展状况;
- ④用户界面必须能提供帮助;
- ⑤宁可让程序多干, 不可让用户多干。

(2) 人机界面设计中经常遇到的一个困难的权衡: 灵活性和一致性的矛盾。

(3) AD/Cycle 包括三个方向的接口, 程序员接口、通信接口、用户接口。

(4) 人机交互的手段归结为八个方面: 键盘操作、屏幕滚动、菜单选择、帮助系统、鼠标操作、色彩应用、数据录入、信息显示。

★考点 16: 项目管理 (填空)

(1) 项目管理是指与固定的生产线上的日常生产管理不同的, 具有更大的变动性、时间性的管理任务。

(2) 项目管理四个特点

- ①子任务多, 关系复杂。
- ②任务不可重复, 形势不断变更。
- ③协调组织的任务十分突出, 资源浪费闲置的风险与合理地优化组合、提高效益的机会并存。
- ④信息处理工作的作用与意义更为突出。进度和质量对于软件开发来说, 是最关键的问题。

★★考点 17: 项目管理与版本管理的基本思想 (填空、简答)

(1) 项目管理的基本目标

- ①使产品 (或工程) 的质量得到有效的控制。

②保证整个系统按预定的进度完成。

③有效地利用各种资源，尽可能使资源的闲置与浪费减少。

④控制与降低成本。

(2) 为了达到这些要求，人们采用了工程计划网络方法、甘特图、检查点方法、排队论等

(3) 版本管理的核心是保持两个一致性。

(4) 版本管理的具体方法包括规定长远的版本更新计划，制订版本有关的信息范围及收集、管理方法，在项目组内明确分工，在程序编写、测试、文档缩写等工作中贯穿项目的思想及要求。

第四章 软件开发工具的技术要素

★考点 1：软件开发工具的基本功能（简答）

(1) 提供描述软件状况及其开发过程的概念模式，以协助软件开发人员认识软件工作的环境与要求，合理地组织与管理软件开发的工作过程。软件开发工具是引导人们建立正确的、有效的概念模式的一种手段。

(2) 提供存储和管理有关信息的机制与手段。

(3) 帮助使用者编制、生成及修改各种文档。

(4) 帮助使用者编写程序代码，即一般常说的代码生成。

(5) 对于历史信息进行跨生命周期的管理，把项目进度与版本更新的有关信息科学地管理起来。

★★考点 2：软件开发工具的一般结构（单选、填空、简答）

(1) 软件的基本组成部分就是代码和文档。

(2) 总控和人机界面。它是使用者和软件开发工具之间交流信息，实现所有的支持功能的桥梁。这一部分的有效实现对于软件开发工具来说，是它的实用性和灵活性的主要保证。一方面，人机界面面对着不同的用户类型。这些用户既有区别，又有共性。人机界面既要保持统一和一致，又要能够在正确区分和识别用户的基础上，针对不同用户的要求与习惯，提供有效的帮助。另一方面，总控面对着系统内部的各种功能与各种信息，它要保证各部分相互之间的协调一致，保证各部分之间信息有效地、准确地流通。

★★★考点 3：总控与人机界面（单选、填空、简答）

(1) 作为一个完整的、一体化的软件开发工具，总控和人机界面处于中心的位置。

(2) 在技术上的考虑可以归纳为三个要点：面向使用者、保证信息的准确传递、保证系统的开放性（或灵活性）。

(3) 软件开发过程的每一个周期分为五个阶段：需求分析阶段、分析设计阶段、编码阶段、测试阶段及维护阶段：

①需求分析阶段的任务是建立逻辑模型。具体地说，首先建立起软件所处领域或环境的模型，其次，建立软件所要处理的信息的静态模型，即数据模型。第三，建立信息流通的模型，即信息的来源、去向、存储及处理的逻辑过程。

②分析设计阶段的基本任务是完成系统的总体设计，这包括数据结构的详细设计、处理过程的详细设计、子系统或模块的划分以及它们之间相互联系的具体规定；

③编码阶段指具体地编写软件的阶段，这里是实际的程序代码的产生点。

④测试阶段的任务是对已经完成的各个模块或子系统进行测试、调整，以便最终形成完整的软件。

⑤维护阶段的任务是组织管理软件的日常运行，收集运行中的状态信息及出现的问题，并且及时地进行局部的修改与完善。

(4) 面向用户，最根本的是要立足于符合实际的应用领域，符合软件开发人员思路的概念模型，包括

静态模型和动态模型。

(5) 面向用户的原则

面向用户的原则具体体现在两方面，一方面是总控对各部分的调度与安排应当符合上述模式；另一方面，向用户提供的统一界面应当体现这个模式。

★考点 4：信息库及其管理（填空）

(1) 信息库是软件开发工具的基础。

(2) 信息库的技术考虑主要涉及四个问题：信息库的内容、信息库的组织方式、信息库的管理功能、历史信息的处理方法。

★★考点 5：信息库的内容（单选、简答）

(1) 关于软件应用的领域与环境的状况。

(2) 设计成果，包括逻辑设计与物理设计的成果。

(3) 运行状况的记录。

(4) 有关项目管理与版本管理的信息。

考点 6：信息库的管理功能

与一般数据库管理系统不同的两点

(1) 信息之间逻辑联系的识别与记录。

(2) 如何实现定量信息与文字信息的协调一致。

★★★考点 7：代码生成（单选、填空、简答）

(1) 代码生成器的基本任务是根据设计要求，自动地或者半自动地产生相应的某种语言的程序。

(2) 输出的代码有两种情况：某种高级程序设计语言的代码和某种机器（包括硬件和操作系统）环境下可运行的机器指令。

(3) 对话屏幕、输入屏幕、输出报表等类型的模块比较容易生成。

(4) 对于统计分析、数据加工、决策判断等类型的模块则比较难生成，因为它们的灵活程度较大。

(5) 生成代码时依据：首先是信息库里已有的有关资料。其次，代码生成器还要利用各种标准模块的框架和构件。第三方面的依据是使用者通过屏幕前的操作送入的信息。

(6) 文档生成包括文章、表格、图形三大类。

第五章 软件开发工具的使用与开发

★考点 1：购置与开发的权衡（简答）

(1) 专门作为商品生产出来的工具是以销售为目的软件开发工具，它在某些环节上可代替一些简单的、重复性的工作；对项目较大，周期较长的情况，还可以使工作的组织管理比较规范，信息的管理与检索比较方便，部分地做到信息与知识的重用和共享。但价格比较昂贵，特别是项目管理和系统分析工具也不易按实际需要剪裁与扩充。

(2) 为自己所用而开发的工具目标明确，符合需要，便于扩充。但规范化、商品化程度低，问题与故障多。

★★★考点 2：影响购置与开发权衡的因素（单选、简答）

(1) 准备从事的软件开发工作的性质与要求；

(2) 开发人员对支持工作与支持程度的实际需要；

①对于某些项目的开发、报表设计或屏幕设计的数量很大，又不太复杂，购买现成设计工具则是合理的。

②对于一些游戏或教学辅助软件，则不能用屏幕设计的现成工具，需要多媒体软件开发工具。

(3) 工作环境；

(4) 人员的因素。

★★考点 3：软件开发工具的选择与购置（单选、填空、简答）

(1) 明确目的与要求

①为哪个软件开发项目而使用工具；

②在哪个工作阶段使用工具；

③工具将供哪些人使用；

④工具将在怎样的软件和硬件环境下运行。

(2) 调查市场研究重点

①软件开发工具的功能；

②软件开发工具的性能；

③软件开发工具所使用或依据的开发方法或开发理论；

④软件开发工具的运行环境；

⑤软件开发工具的文档资料是否齐全；

⑥软件开发工具的服务、培训条件；

⑦价格。

考点 4：在软件开发工具的选择与购置工作中步骤与方法

(1) 明确购买软件开发工具目的与要求。

- (2) 明确购买软件开发工具的环境条件与制约条件。
- (3) 市场调查。
- (4) 对于可供选择的各种工具进行综合比较。
- (5) 进行测试和检验。
- (6) 正式签约购置。
- (7) 安装与试用。

★★★考点 5：组织管理工作（单选、填空、简答）

- (1) 严格使用制度。
- (2) 记录使用的详细过程。
- (3) 培训使用人员。
- (4) 经常进行审计与评价工作。

所谓审计是指对一个系统的运行状况及效率进行检测与评价，以便进一步用好或改进这个系统。（填空）

审计的基础就是前面所述的日常记录的信息，没有日常信息的积累，审计工作就无法进行。（单选、填空）

审计的范围应当包括工具使用的环境、人员、工作负担、工作效果、存在问题、改进方向等许多方面。（简答）

★★★考点 6：软件开发工具的开发（单选、简答）

自行开发软件开发工具须注意的要点

- (1) 从实际出发，设定现实的、有限的目标。
- (2) 自行开发工具一定要坚持短小实用，逐步积累，避免期望过高，贪大求全。
- (3) 注意文档的齐全与资料的积累。

第六章 软件开发工具的现状与发展

考点 1: 软件开发工具的发展现状

(1) 国外发展状况

- ①软件开发工具的兴起是在 20 世纪 80 年代中期。
- ②IBM 于 1989 年提出的 AD/Cycle——关于应用系统开发和 CASE 工具的总框架。

(2) 国内的发展状况

对我国的软件产业来说，当务之急应是大力普及软件工程及方法学的知识，提高软件开发工作的水平，这样才能使软件开发工具的应用具有实践的基础。

★★★考点 2: 软件开发工具的几个值得注意的发展方向（单选、填空、简答）

从目前的软件开发工具的发展来看，智能化、网络化、一体化、标准化是值得重视的几个方向。

(1) 智能化，是在软件开发工具的研究与使用中引用人工智能、神经网络等技术，使得软件开发工具对于不确定性的信息、模糊信息具有更强的处理能力。

(2) 网络的应用是计算机应用领域中的一个重要方向。

(3) 一体化对软件开发过程与相关信息认识深度。

(4) 标准化的问题是由软件部件、组合软件的想法引起的。软件开发工作分成两个部分：软件构件的开发（如硬件的芯片），用这些构件组成大型软件。

开发工具一体化的困难：一体化的趋势早在 20 世纪 80 年代后期已经十分明显。但是，在开始时人们显然低估了一体化的困难。许多教训告诉我们，困难不在于技术上的问题（如速度、容量等），而在于对软件开发过程与相关信息认识深度。只有对于软件开发中涉及的各种信息，以及在开发过程中它们的发生、变化、关系、一致性等有了完整与深刻的理解，才能真正实现软件开发工具的一体化。关于信息库的内容及处理功能的讨论，从理论上为实现一体化提供了基础。

为了使软件构件能够组成大型软件，软件构件必须实现标准化。（单选）

★★考点 3: 软件工具的发展轨迹（单选、填空）

(1) 40 年来，软件开发工具经历了三个时代：结构化时代、面对对象时代、互联网时代。

(2) 在软件开发工具发展的过程中，存在着多样性和趋同性并存的特点。

(3) 软件开发工具包括一个庞大的谱系，其中一端是从某些计算机语言，通过增添各种辅助功能发展出来的工具，另一端是抽象的概念模式出发设计出来的开发平台或开发环境。

作为一款著名软件工具，Google 网站工具包（GWT）出现于互联网时代。（单选）

作为一款著名软件工具，并行版本系统（CVS）出现于结构化时代和面向对象时代的交集。（单选）

★★考点 4: 软件开发工具（单选）

随着软件规模的增大，复杂度的提高，开发人员已经不满足于手头的开发工具作为一门语言的延伸存

在。新的开发工具不断诞生，不断向着更高的集成度和更强的智能化迈进。

关于 Java 语言的技术中，属于数据库支持级别的是 JDBC。（单选）

关于 Java 语言的技术中，属于网络和 Web 开发支持技术的是 Servlet。（单选）

★★考点 5: Visual Studio 和 Eclipse（单选、填空）

（1）Visual Studio 属于集成开发环境（IDE）。它支持多种编程语言，如 C++、C#、Visual Basic 等，并且可以通过插件扩展其功能。

（2）作为对软件开发环境的要求，开发人员希望可以在代码编写过程中能把精力集中到优化代码逻辑和算法上。

（3）在大中型项目中，由于项目复杂，开发周期长，交互性高，大多数开发团队倾向使用的开发工具是集成开发工具。

第七章 Eclipse 入门

★★考点 1: Eclipse 简介【填空】

（1）Eclipse 是一个开放源代码的、基于 Java 的可扩展集成应用程序开发环境。

（2）Eclipse 的体系结构：①运行时内核；②工作空间；③工作台；④其他插件。

工作台为 Eclipse 提供用户界面。（填空）

第八章 Eclipse 工作台

考点 1: Eclipse 工作台（填空）

工作空间以项目为单位组织文件和目录，它将各种资源组织成树形结构，项目位于树的根部，文件和文件夹位于树枝的位置。

★★考点 2: 透视图及视图介绍（单选、填空）

（1）通过 Eclipse 的菜单栏，用户可以对整个集成开发环境进行整体的操作。

（2）视图包括导航器视图又称为资源管理器、大纲视图、控制台视图等。

显示当前 Eclipse 集成环境中加载的所有项目和各个项目中的文件列表的视图导航器视图。（单选）

在 Eclipse 中，程序员想找到当前活动编辑器所打开文件的概要，如函数、变量等，他应该打开大纲视图。（单选）

Eclipse 工作台的任务视图用于显示程序代码中未完成的任务。（单选）

为了提醒自己未完成的编程工作，程序员可以在代码中添加一条以 TODO 开头的注释标记，这样就可以在 Eclipse 的任务视图添加一项任务。（填空）

考点 3：管理项目（单选）

在 Eclipse 集成开发环境中，每个小程序都是以项目为单位存在的，源代码、注释、配置文件、各种文件夹等都封装在项目里。

第九章 使用 Eclipse 进行 C/C++ 开发

★考点 1：新建 C/C++ 项目（单选、填空）

- (1) 所有程序都是以项目为单位，编写程序之前，首先需要新建 C/C++ 项目。
- (2) 编译 C 程序是指将.c 文件编译为可以运行的.exe 文件。
- (3) 编译成功后将在资源管理器中看到 Release 文件夹，Release 文件夹中包含了被编译的可执行文件。

为了在不指明完整路径的情况下，系统就能够认识 MinGW 的执行命令，需要设置系统环境变量。（填空）

★考点 2：Eclipse 的浏览功能（单选）

Eclipse 为 C/C++ 程序员提供了一系列工具以方便程序开发。其中，使用浏览功能可以从多个角度快速查看并定位到程序中的各个元素，包括代码的层次结构、调用关系、继承关系等。

(1) 在 C 语言中，include 称为文件包含命令，其意义是把尖括号或引号内指定的文件包含到本程序，成为本程序的一部分。被包含的文件通常是由系统提供的，其扩展名为.h。因此，也称为头文件或首部文件。

(2) 大型的程序中可能包含若干全局变量与局部变量，Eclipse 提供了从变量的任意引用位置直接定位到变量定义行的方法。

(3) 在一个符合面向对象思想的程序中，存在多种抽象、封装和继承特征结构，使得程序易于阅读和维护。Eclipse 提供了快速打开程序中已经在使用的特定数据类型及其相关继承结构的方法。

(4) 与变量的声明和引用类似，大型程序中，存在复杂的函数调用和参数传递过程。Eclipse 提供的浏览功能可以浏览函数的定义与调用层次关系并快速定位到需要的函数。

在 Eclipse 中，“打开变量声明”选项的快捷键是 F3。（单选）

Eclipse CDT（C/C++ Development Tools）的当前默认透视图是 C/C++。（单选）

★考点 3：Eclipse 的重构功能（填空、简答）

(1) 在 Eclipse 中，重命名变量位于菜单栏中的重构菜单项中。

(2) 使用 Eclipse 重构功能，用户可以在不影响程序行为的情况下进行系统范围内的代码更改。具体包括重命名变量、抽取方法、抽取常量。

(3) 将程序中出现多次的代码段抽取出来形成方法，可以大大减少代码的长度并增加程序可读性和易维护性。

第十章 调试程序

★考点 1: GDB 简介 (填空)

GDB 是 GNU 开源组织发布的一个强大的 UNIX(不仅限于 UNIX)下的 C/C++程序调试工具。它搭配 MinGW 使用, 可以使用户完成整个 C/C++程序的编译和运行工作。有了 GDB, 用户可以在调试本地的 C 程序时不需要第三方 C/C++工具。

★★考点 2: 设置行断点 (单选、填空)

而断点视图会清楚地显示每个断点的状态, 激活状态的断点会以蓝色的圆圈显示, 非激活状态的断点则是以白色透明圆圈显示。

(1) 添加行断点: 设置后 Eclipse 将在标记区域上的这一行显示一个蓝色圆圈图标。

(2) 激活和禁用行断点: 禁用后的行断点在 Eclipse 中以白色圆圈显示。

考点 3: 设置方法断点

Eclipse 对某个方法或函数设置断点, 在程序运行到该方法的第一行时自动暂停。

考点 4: 设置断点动作

Eclipse 规定了四种可用动作, 分别是声音动作、日志动作、重新启动程序动作和外部工具动作。

考点 5: 调试程序 (单选)

(1) 单步遍历程序包括: ①单步跳入; ②单步跳过; ③单步返回。

(2) 中断调试

使用 Eclipse 调试程序时, 用户按下<Ctrl+F2>键, 这时程序将执行的操作是中断调试 (单选)

第十一章 Eclipse CDT 开发常用功能

★考点 1：定制工具栏（单选、填空）

观察 Eclipse 工具栏发现，默认情况下工具栏区域被竖直的虚线分成若干部分。此工具栏处于解锁状态，用户可以随意地更改工具栏的状态。

为了防止工具栏被意外改动，定制工具栏后可以将其锁定。

考点 2：定制透视图（填空）

（1）打开的窗口上方可以看到四个标签，分别用来定制工具栏、菜单栏、命令组和快捷方式。

（2）Eclipse 本身自带的透视图无法删除。

考点 3：使用代码模板

（1）Eclipse 允许用户在注释、代码和文件三个维度设置代码模板。

（2）在打开的代码模板编辑器中为 C 程序文件定制模板需要注意的是，变量以\$开头，并用{}括起来

考点 4：将文件替换为历史版本

选择其中一个版本，即可在底部窗口中显示该版本与目前文件的对比结果，其中的差异将会以浅绿色标记显示。

第十二章 CVS 的安装及使用

考点 1: CVS 介绍

使用 CVS 可以对代码进行集中管理，记录代码所有的更改历史，提供协作开发的功能并支持多人同时修改代码文件。

- (1) 对程序修改进行有效的管理：只有管理员可以将程序冻结和解冻，被冻结的程序是不允许修改的。
- (2) 将开发环境与测试环境、运行环境进行有效的隔离

考点 2: 常用版本控制软件

CVS：它在全球中小型软件企业中得到了广泛使用。其最大的遗憾就是缺少相应的技术支持，许多问题的解决需要使用者寻找资料，甚至是读源代码。

★★★考点 3: CVS 的日常使用（单选、填空、应用）

- (1) 代码同步

如果本地文件包含了 CVS 上该文件的所有代码，并且有新的代码添加，开发人员确认以后认为可以提交该代码，这时应该执行 Commit 命令。（单选）

- (2) 工作备份

用户对资源的修改不是直接在 Server 端进行的，而是根据资源库的内容创建一个本地的工作备份，用户在工作备份中工作，工作完成后再将修改的内容提交到资源库

- (3) 标签

标签与版本是同义词。对某个时刻的快照赋予一个标识名称，标识名称就被称为标签。将来通过标签就可以获得模块在该时刻的快照

第十三章 Eclipse 插件的使用与开发

考点 1: 插件的定义

插件的功能多种多样，它们被用来满足用户的不同需求。除了运行时的内核外，Eclipse 都是由若干插件组成的。

★★考点 2: 插件的构成（单选、填空）

插件都是由一个插件清单文件和一些可选文件组成。

一个典型的插件文件夹中一般包括以下文件

- (1) plugin.xml，插件清单文件。
- (2) plugin.properties，插件的一般信息。
- (3) about.html，记录证书信息。
- (4) *.jar，插件需要的类文件。
- (5) lib，容纳第三方 JAR 包。
- (6) icons，容纳 icon 文件，通常是 GIF 格式。
- (7) 其他需要的文件。Eclipse 在首次启动时，会扫描并查找 plugins 目录下的已被定义的插件。

★★考点 3: PDE 基本操作（填空）

- (1) 清单编辑器

①概述页面

此页面由“一般信息”和“执行环境”两个主要部分组成，它们定义了重要的插件属性

②依赖性页面

PDE 的重要任务是要准确地反映运行时的类路径，以保证插件开发运行的准确性。

- (2) 插件模版

PDE 为平台的常用扩展点提供了复杂性和功能均有所不同的模板。

考点 4: 常用插件介绍与使用

EMF 插件：EMF 是 Eclipse 的建模框架。

第十四章 常用建模工具

★★考点 1：面向对象方法基础（单选、填空）

（一）面向对象方法的出现和发展

（1）面向对象概念认为客观世界的任何事物都是“对象”。或者说对象是客观世界的抽象。

（2）在程序设计中，对象表达为被描述事物的数据和对数据的处理的统一整体，也称为封装。把对象按相同(相似)特征区分称为类

（二）面向对象的一些概念

（1）属性：属性是以静态的数据组成，用以描述类和对象所固有的特征，是类和对象的性质，并以此来区分不同的类和对象。

（2）抽象：抽象是事物或现象的简括描述，突出事物或现象之间的共性

（3）封装：封装即信息隐藏。对象就实现了很好的封装，它向外提供了一组数据结构和一组操作，而把内部的细节隐藏起来。

（4）继承：继承指对象继承它所在类的结构。操作和约束也指一个类（子类）继承另一个类（父类）的结构。继承体现了父类和子类之间的共享机制。

（5）消息与多态：在面向对象方法中，完成一件事情的方法就是向有关对象发送消息。

（三）面向对象方法与传统方法

面向对象分析法是一种全新的、不同于面向过程的分析方法，它更侧重于建模而非分析流程。

★考点 2：组件思想（单选、填空）

组件是一个可重用的软件构件，一个预先构建的封装的代码模块，它能够与其他组件或硬编码一道很快地生成定制的应用程序。组件的目标是粗粒度的复用，它的核心是接口。

★考点 3：UML 与系统开发（简答）

UML 把系统开发分成五个阶段：需求分析、分析、设计、编程和测试。

（1）需求分析：UML 用 UstCase 模型来获取客户需求。

（2）分析：析阶段涉及基本抽象(识别类和对象)和问题领域中的机制。

（3）设计：在设计阶段，分析的结果要从技术上得到解决方案。

（4）编程：在编程阶段，要把从设计阶段获得的类变换成面向对象编程语言的程序代码。

（5）测试：系统建成后需要经过单元测试、集成测试、系统测试和验收测试。

★★考点 4：需求分析（填空、简答）

（1）用例图

用例和用例之间存在着三种关系，分别是扩展关系、包含关系和泛化关系。

（2）活动图

活动图在本质上就是流程图，它很好地描述了系统的活动、判定点、先后顺序和分支等。

考点 5：系统分析与设计

(1) 顺序图是强调消息时间顺序的交互图。

(2) 协作图是强调参加交互的各对象的组织。组件图通常包含三种元素，即组件、接口和依赖关系。

计数器组件和连接数据库组件是这个系统中被重用次数很多的两个组件。

(3) 类图是根据系统中的类以及各个类之间的关系描述系统的静态视图。类由三部分组成，分别是类名称、类属性和操作。类图是生成代码的核心要图。