

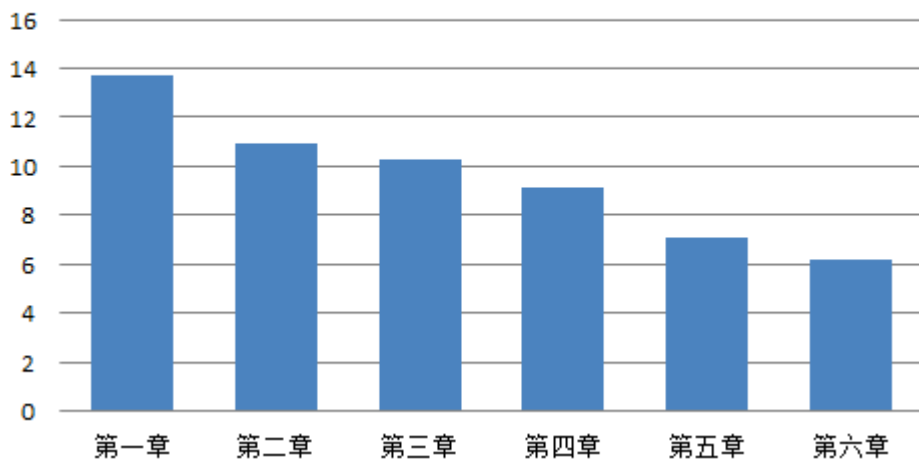
自考本科专业课
软件开发工具（课程代码:03173）
通关宝典（讲义）

软件开发工具

SOFTWARE DEVELOPMENT

软件开发工具考试内容与考核标准

考点分布图



通关宝典（又称全书讲义），是尚德老师们结合近年考试动态，精细研究官方教材和考试大纲，为学员们准备的一份备考利器。相比官方教材，内容精简了三分之二，保留了重要的考点、知识点，极大方便学员在最短时间掌握最核心内容的需要。

全书为以下板块：

- 考点频率图：展现全书各章节的重要程度，为大家集中精力解决主要矛盾指明方向；
- 章节思维导图：建立知识框架，才不会迷路方向，提纲挈领，快速把握章节内容；
- 课前故事：章节前方框内容（无须记忆），帮助大家轻松一刻，联系实际，举一反三；
- 正文部分：以知识点形式划分，方便碎片化时间学习，各个击破，准确把握学习进度。

所有的馥郁花香，都来自沉潜酝酿。

学习是一种信仰，为所有正在阅读本书的城市奋斗者喝彩，祝所有的尚德学子学有所获，期待我们举杯相庆的那一天。

目 录

目录

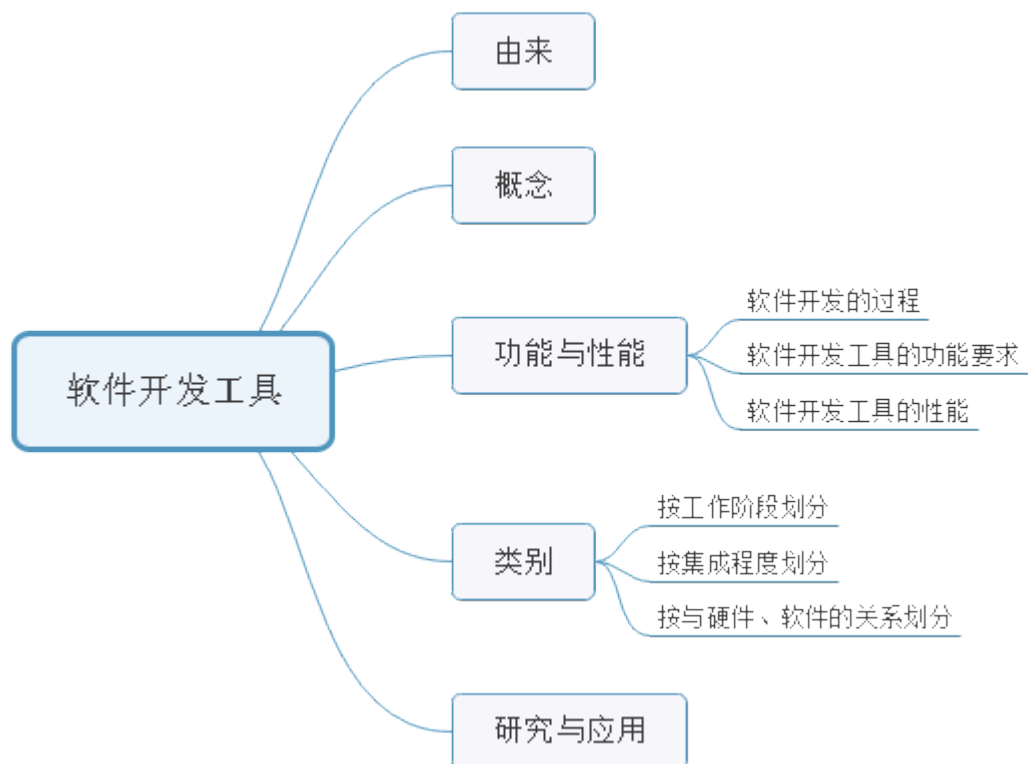
软件开发工具考试内容与考核标准	1
理论篇	5
第一章 绪论	5
模块一 软件开发工具的由来	6
模块二 软件开发工具的概念	7
模块三 软件开发工具的功能与性能	9
模块四 软件开发工具的分类	10
模块五 软件开发工具的研究与应用	10
第二章 软件开发过程及其组织	12
模块一 软件开发的困难	13
模块二 软件开发方法的发展	13
模块三 软件开发过程的管理	17
第三章 软件开发工具的理论基础	20
模块一 软件开发过程的信息需求	21
模块二 概念模式	22
模块三 信息库及其一致性	23
模块四 人机界面及其管理	24
模块五 项目管理与版本管理	25
第四章 软件开发工具的技术要素	28
模块一 基本功能与一般结构	29
模块二 总控与人机界面	29
模块三 信息库及其管理	31
模块四 文档生成与代码生成	32
模块五 项目管理与版本管理	33
第五章 软件开发工具的使用与开发	36
模块一 购置与开发的权衡	37

模块二 软件开发工具的选择与购置	37
模块三 软件开发工具的使用	38
模块四 软件开发工具的开发	39
第六章 软件开发工具的现状与发展	42
模块一 软件开发工具的发展现状	43
模块二 软件开发工具的发展方向与趋势	44
模块三 从计算机语言发展而来的工具	45
模块四 综合性的平台或开发环境	46

理论篇

第一章 绪论

思维导图



模块一 软件开发工具的由来

知识点一 软件开发工具的产生与演变

软件开发工具是在人类以往发展的许多技术的基础上，适应社会的实际需要，从无到有，从不完善到比较完善逐步发展起来的。

通用软件作为辅助工具的阶段	专用的软件开发工具的阶段。	集成的软件开发环境阶段。	二十一世纪的软件开发新阶段。
有许多工作是通用软件所无法完成的。通用软件完成某些工作也不能反映其逻辑内涵。常常遇到难于保持一致性的困难。	对软件的开发缺乏全面的、统一的支撑环境。	进入集成的软件开发环境阶段的标志是 IBM 在 1989 年提出的理论框架。	有两个鲜明的特点.第一是面向网络，不仅基于网络，而且用于网络。第二是开源软件的兴起和运用。

知识点二 使用通用软件辅助软件开发的优缺点

优点：与手工方式相比更方便、效果好、便于存储和修改，减少了许多工作量

缺点：

- (1) 许多工作通用软件无法完成
- (2) 通用软件完成某些工作，只能表现其表面形式，而不能反映其逻辑内涵
- (3) 常遇到难于保持一致性的困难

知识点三 软件开发工作的发展变化及其特点

软件开发工作的发展变化及其特点

最初阶段，用机器语言写成一系列机器指令，供硬件执行；
 第二阶段，用汇编语言开发软件，由汇编程序完成转换；
 第三阶段，用高级语言开发软件，由编译程序完成转换；
 第四阶段，在各种软件开发工具帮助下开发软件，由编译程序完成转换；
 新阶段，由软件构架师完成构架设计，程序员用软件开发工具完成程序开发，由编译程序完成转换。

知识点四 重要的时间点

20 世纪 60 年代初期，FORTRAN，ALGOL 和 COBOL 等高级程序设计语言的成熟与普及，标志着计算机真正走出了难以应用的困窘局面，开始向社会生活的各个领域全面渗透。

20 世纪 60 年代末期开始，人们对于软件工作的认识大大深入了。在认识到软件工作重要性的同时，也认识到了软件工作的困难性，这就是所谓“软件危机”问题。

20 世纪 80 年代以来，一些专门用于支持软件开发的软件开发工具陆续问世，从而进入了专用的软件开发工具的阶段。

注意点：

1. 汇编语言相对于机器语言：针对难以记忆的、无意义的、二进制的字符串，人们视图用在英语中具有一定意义的单词来代替它，这就是所谓的“助记忆码”，或汇编码。但汇编语言未能突破与机器指令一一对应的限制。
2. 高级程序设计语言相对于低级语言有哪些突破以及本身的不足：高级程序设计语言突破了与机器指令一一对应的限制。用尽可能接近自然语言的表达方式描述人们设想的处理过程，而把这种表达方式向机器指令的转化工作，交给专门的“工具”——编译系统去完成，高级设计语言实现了对机器的独立性，提高了程序的可移植性。不足之处为高级程序设计为过程化语言，需要安排执行的内容（与次序），需要考虑解决问题的细节。清楚如何解决问题，加重了用户的负担。
3. 操作系统的出现对计算机的应用产生的影响：操作系统的出现改善了人们应用计算机的条件。操作系统利用计算机本身迅速处理信息的优势，自动地完成系统初始化、文件管理、内存管理、作业管理、处理机管理等一系列工作，把计算机系统中的各种资源有效地、协调地管理起来，把原先由操作员担负的大部分职责接过来。

模块二 软件开发工具的概念

知识点一 概念

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、成文和管理各阶段，对软件开发提供各种不同程度帮助（支持）的一类广泛的软件。

知识点二 要点

- （1）它是在第三代程序设计语言基础的上，软件技术进一步发展的产物；
- （2）它的目的是在软件开发的全过程中给予人们各种不同方面、不同程度的支持或帮助；
- （3）它支持软件开发的全过程，而不是仅限于编码或其它特定的阶段。

知识点三 与软件开发工具有关的概念、术语

- 1、第四代语言（4GL）
- 2、CASE 工具（两种理解：1.计算机辅助软件工程 2.计算机辅助系统工程）
- 3、可视化程序设计
- 4、即插即用程序设计
- 5、组件程序设计
- 6、最终用户计算

注意点：

1. “软件实际上是人类知识与经验的结晶”的理解：硬件和软件缺一不可；有事先编好的指令（代码、软件、程序），硬件才能完成任务；这些指令就是人们在实践中形成的工作规范，可以对不同的数据反复使用，进一步提高了人类的能力。

2. 软件开发工具的提出与使用，是软件技术发展的一个新的阶段：区别于以往阶段的特点，它的发展要表现在四个方面。（1）自动化程度的提高，编程中的部分工作已由工具代替执行（2）将需求分析和架构设计包括在软件工作的范围之内，从而使软件开发过程进一步向用户方面延伸。顺应大量开源软件出现的客观环境，进一步显示出软件开发的继承性和开放性，进一步证明了软件是人类知识积累和传承的新的、有力的手段。（3）将软件开发工作延伸到项目及版本管理，从而超出了一次编程的局限，而扩展到了作为一个不断发展的客体生长完善的全过程。这也是软件研制从个体的、手工作坊的方式向科学的、有组织、有计划的方式转变的一个重要表现。（4）吸收了许多管理科学的内容与方法，将组织、管理等项目负责人的思想与方法放到了更重要的位置。人们越来越深刻地认识到，软件生产的成败更多地依赖于合理地组织与协调，而不是领导者或程序员个人的编程能力

3.20 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。

模块三 软件开发工具的功能与性能

知识点一 软件开发的过程

(1) 软件开发的过程

开发阶段	主要任务	成果
需求分析	从用户初始要求出发，经过大量的调查研究，抽象出应用领域中实际的信息需求，设计出在计算机系统内外的、合理的信息流程，并规定软件的功能与性能要求，最后形成严格的、明确的、可供实际开发使用的“软件功能说明书”。	软件功能说明书
总体设计	根据软件功能说明书的要求，完成软件的总体设计（包括整个软件的结构设计、公用数据库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）系统的总体设计文件（结构图、模块清单、公用数据结构）和各个模块的设计任务书	系统的总体设计文件（结构图、模块清单、公用数据结构）和各个模块的设计任务书
实现阶段	程序的编写与文档的编写	代码和文档
测试阶段	完成模块的调试与整个软件的联调	初步的软件产品

(2) 软件开发工具的功能和性能要求

功能：软件能做什么事。
性能：事情能做到什么程度。

- 1、功能要求：
- (1) 认识与描述客观系统

(2) 存储与管理开发过程中的信息

(3) 代码的编写与生成

(4) 文档的编制或生成

(5) 软件项目的管理
- 2、性能指标：
- (1) 表达能力或描述能力（首先考虑）

- (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) 对软、硬件环境的要求

模块四 软件开发工具的分类

分类		用途	举例	比较
按工作阶段	设计工具	用于实现阶段	代码生成器、4GL、测试工具	出现最早，数量最多，适用于个人开发
	分析工具	支持需求分析	数据字典管理系统、绘图工具	出现较晚，数量少些，适用于规模较大的软件开发
	计划工具	保存整个项目的宏观信息，为项目主管人员服务		出现最晚，数量最少，项目达到一定规模时才需要
按集成度	专用工具	面对某一工作阶段或某一工作任务		
	集成工具	面对软件开发全过程		
与软硬件关系	依赖软硬件	如：设计工具		
	独立软硬件	如：分析工具、计划工具		

模块五 软件开发工具的研究与应用

知识点一 对软件开发工具的研究来源三个方面

- (1) 软件开发工具的使用者
- (2) 一些厂家和研究单位
- (3) 软件技术和系统工程专家

知识点一 作为一般的使用者，学习软件开发工具的目的

了解软件开发工具的概念、理论基础、基本功能、发展现状与前景，以便在

实际工作中正确地选择与使用软件开发工具,在必要时能参加或组织软件开发工具的自行研制,从而达到提高软件工作水平与效率的目标。

本章题型分析

本章常考题型分为:单选题(常见分数占比为2.00%)填空题(1.80%)简答题(8.60%)

1.(201210 填空题)软件开发工作的起点是_____的提出。

答案:初始要求

解析:软件开发工作工具就是帮助人们开发软件的工具。起点是初始要求的提出即需求分析,第二个阶段为总体设计,第三阶段为实现阶段,第四阶段是测试或调试阶段。

2.(20610 单选题)为项目主管人员服务的软件开发工具是()
A.计划工具 B.编程工具 C.设计工具 D 软件工作环境

答案:A

解析:.计划工作是从更宏观的角度去看待软件开发。它不仅从项目的角度,帮助人们组织与实施项目,并且把有关进度、资源、质量、验收情况等信息有条

3.(201210 简答题)什么是专用的软件开发工具?它有什么优点和不足?

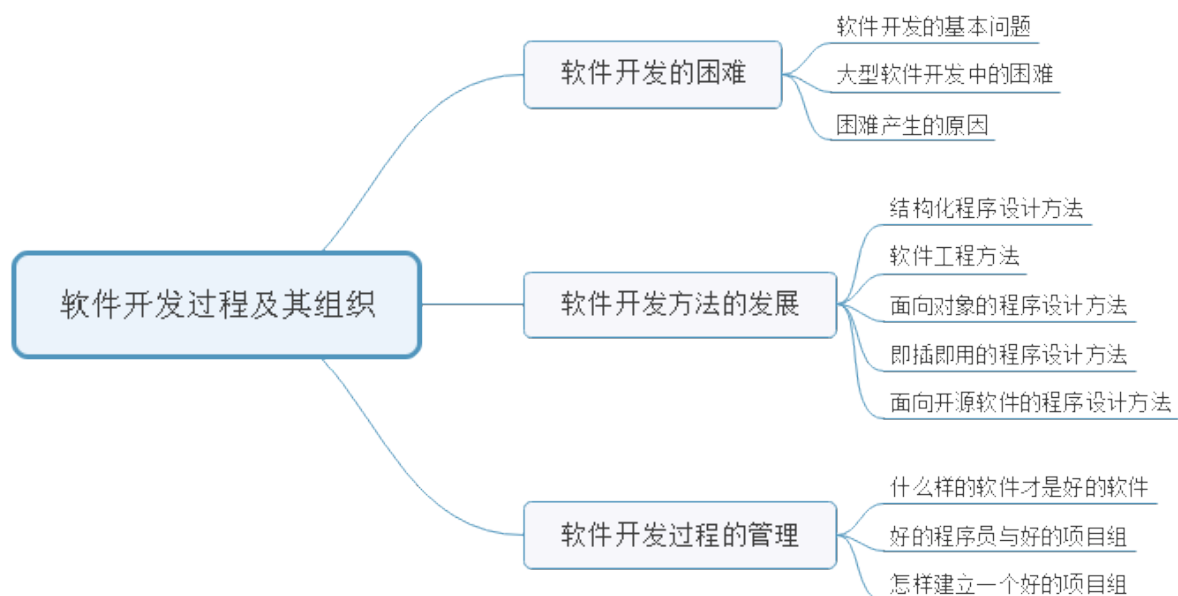
答案:专用的软件开发工具是面对某一工作阶段或某一工作任务的工具。优点是能提高软件开发的质量和效率。缺点是一致性的保持困难。对软件的开发缺乏全面的、统一的支撑环境。

解析:一些专门用于支持软件开发的软件开发工具陆续问世,从而进入了专用的软件开发工具的极端,但是,一批专用的软件开发工具的出现并未解决保持一致性的问题。问题的根源很明显,那就是对软件的开发缺乏全面的、统一的支撑环境。

本章为重点章节,作为全书的导言,概括地介绍软件开发工具的由来、概念、功能、类别与使用环境。其中软件开发工具的由来为重点掌握知识点。归纳总结的中注意点多为简单题需加强记忆。

第二章 软件开发过程及其组织

思维导图



模块一 软件开发的困难

知识点一 软件开发的基本问题

对单个程序员来说，做好软件的关键是做好两个转换。

第一个转换：从用户对软件功能的理解到程序员对软件功能的理解之间的转换。

采取的手段：利用尽可能标准化的方法编写程序设计任务书，用明确的语言或图形、表格，把程序要处理的信息的内容、格式、来源、去向、存储、处理清楚地表达出来，作为双方共同理解。

第二个转换：从程序员的理解到程序的实现。

采取的手段：提高程序员的知识水平和实际经验。



知识点二 大型软件开发中的困难

- (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
- (5) 版本更新带来的困难

知识点三 困难产生的原因

- (1) 大系统的复杂性；
- (2) 个人之间的组织与协调；
- (3) 各应用领域间的差别；
- (4) 时间、变化的因素。

模块二 软件开发方法的发展

知识点一 大型软件开发过程中的各种角色及其承担的主要任务

角 色	主 要 任 务
用户	提出需求、验收软件、使用软件、提出修改要求
程序员	编程、写文档、调试自己的程序、向项目负责人提交工作成果
项目负责人	分析用户需求形成设计方案、向程序员分配任务、验收程序员的工作成果、对整个软件联调、向用户交付使用、接收与分析用户的修改要求、向程序员分配修改任务、验收修改成果、向用户提交新的版本或修改的结果
硬件	运行、试运行

知识点二 软件开发的方法

（一）结构化程序设计方法

基本思想：把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，并保证模块的划分符合“结构良好”的要求：

- （1）模块的功能在逻辑上尽可能的单一化、明确化
- （2）模块间的联系及相互影响尽可能的少（应当尽量避免逻辑耦合，而仅限于数据耦合）
- （3）模块的规模应当足够小，以便其易于调试

模块划分的方法：自顶向下逐步分解，直到最底层的模块达到要求为止。

实施的原则：

- （1）限制甚至不用 GOTO 语句，绝对禁止超越模块边界的 GOTO 语句。
- （2）子程序尽可能做到只有一个入口、一个出口；
- （3）程序风格应尽量明确、清晰，包括适当增加注释，书写格式体现层次结构，变量名称的选用尽量具有逻辑意义等；
- （4）在程序编写的同时完成有关的文档编撰，不要拖延时间。

适用范围：主要是为程序员服务的

缺点：增大了程序的工作量，增加了编程中的麻烦与琐碎的工作，降低了程序的运行率。

优点：保证软件的可靠性、准确性、易于修改。

（二）软件工程方法

基本思想：借鉴传统工程的思想，从软件开发的经验中归纳出相应的工作步骤和

交流标准，大家都来遵守和执行，从而保证软件开发的进度与质量。

实施方法：

首先，对软件开发的工作过程进行规范。如 IBM 的 AD/Cycle 把软件开发过程划分成五个阶段：

- 第一阶段，需求分析。
- 第二阶段，分析与设计。
- 第三阶段，编程阶段。
- 第四阶段，测试阶段。
- 第五阶段，使用及维护。

在标准化方面：

- 1) 表达方式的标准化，如统一规格的数据流程图、数据字典、模块结构图等。
- 2) 对工作的质量及检查制定标准，在一定的時候作一定的检查。如著名的 ISO9001 标准。

适用范围：主要集中于加强项目管理者的工作

（三）面向对象的程序设计方法

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）和相关的操作（相应于程序模块）。作为一个整体，这些对象不必对外公开这些属性与操作。这就是“封装性”。

（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。这些

关系构成对象概念之间的网络结构。

(3) 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，不必加以重复说明或规定。这就是“遗传性”。

(4) 对象之间可以互送消息，这一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

面向对象程序设计产生巨大影响的根本原因在于它提供的“认识框架”，摒弃了持续许久的“自顶向下”和“自底向上”的争论，把对于复杂系统的认识归结为对一批对象及其关系的认识，通过合理地选择认识的层次，使系统的复杂性保持在可控制的范围内。

(四) 即插即用的程序设计方法

基本思想：用制造硬件的思路来生产软件。一部分人专门生产软件组件，一部分人设计整个软件结构，并且把软件组件插入这个结构，以便迅速完成大型软件的研制工作。

存在的困难：(1) 标准化的问题 (2) 软件部件的提供方式

(五) 面向开源软件的程序设计方法

基本思想：利用开源软件进行程序设计。

注意问题：

- (1) 对应用领域充分了解——判断能否用开源软件的主要依据；
- (2) 开源软件的质量，包括文档和代码的规范程度。

模块三 软件开发过程的管理

知识点一 什么样的软件才是好的

- (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和知识水平;
- (3) 具有足够的速度(不是越快越好), 能在符合用户要求的时间限度内, 给出所要求的结果;
- (4) 具有足够的可靠性, 能够在各种干扰下保持正常工作;
- (5) 程序易读, 结构良好, 文档齐全, 从而保证系统易于修改。

知识点二 好的程序员与好的项目组

由好的程序员组成的项目组, 并不会自然成为好项目组。成为一个好程序员与成为一个项目组的好成员并不是完全相同的。

(1) 就单个程序员来说, 主要的指标有:

- 1) 具有程序设计所需要的基本知识与技能, 如关于数理逻辑的知识与训练, 关于各类数据结构的概念及处理方法等;
- 2) 对本项目所在的那个领域有较深入的了解, 从而能够准确的理解用户的信息需求, 正确地把握有关的信息流程与信息处理原则;
- 3) 对于软件开发的技术环境比较熟悉, 包括硬件设备、软件环境和网络环境, 特别是对于所用的语言, 应当有足够的、实际运用的经验。

(2) 作为项目组的一员以下几点必须严格遵守:

- 1) 保证严格在本模块内操作, 绝不要使用可能干扰其它模块的命令或函数。
- 2) 严格按总体设计的要求和理解去传递参数值, 绝不要随意修改其内容或含义。
- 3) 在对公用的文件或数据库进行存取时, 必须完全地、准确地按统一规定的格式去操作, 绝不能擅自改变。
- 4) 在使用标识符时, 应按照统一的原则尽量使用易于看出逻辑含义的名称。特别是涉及公用数据及参数的时候。
- 5) 严格按照统一的要求编写文档, 在内容、格式、表达方式、符号使用上遵循项目组的统一规定。
- 6) 尽量保持程序风格的一致。如注释行的安排, 行首空格的使用等。

作为项目组的一员参加大型软件的开发, 必须具有高度的组织纪律性和团队精神。没有这样的精神, 好的程序员加在一起也成不了好的项目组。

知识点二 怎么建立一个好的项目组

- 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(负责文档的收集、审核及保管), 必须有专门的测试人员(负责测试和验收有关的部分), 特别是要有专职的、不陷入具体编程的项目负责人。
- 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, 每一项目完成之后都有完整的资料, 并得到妥善保存, 一但需要修改或更新版本时可以立即使用。

本章题型分析

本章常考题型分为: 单选题(常见分数占比为 3.80%) 填空题(3.36%) 简答题(3.36%)

1.(201604 单选题) 结构化程序设计的思想产生于 20 世纪的()。

A 60 年代初 B 60 年代末 C 70 年代初 D 70 年代末

答案: B

解析: 在软件开发的过程中, 产生了许多程序设计的方法, 例如结构化程序设计的思想产生于 20 世纪 60 年代末。

2. (201410 填空题) 人们在实践中认识到, “黑箱” 检验方法只能证明程序____, 而不能保证程序的正确性。

答案: 有错

解析: 黑盒测试也称功能测试或数据驱动测试, 它是在已知产品所应具有的功能, 通过测试来检测每个功能是否都能正常使用, 白盒测试也称结构测试或逻辑驱动测试按照程序内部的结构测试程序, 检验程序中的每条通路是否都有能按预定要求正确工作, 而不顾它的功能。

3. (201310 简答题) 简述一个好的项目组应当具备的条件

答案: 一个好的项目组至少应当具备以下几个条件

第一, 有严格的工作规范

第二, 人员之间有严格的分工, 必须有专门测试人员, 有专职的项目负责人

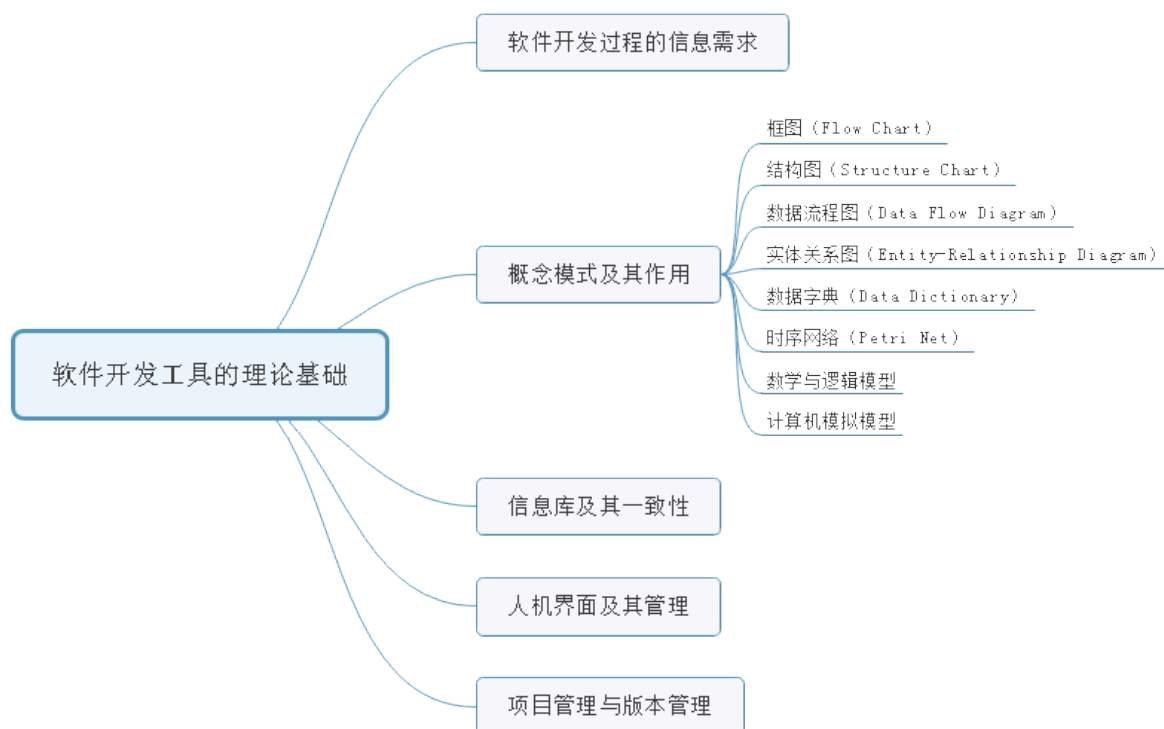
第三, 每个项目都要事先制定详细的时间表。每一项完成之后都有完整的资料

解析：从表面现象来看，取得好程序的关键是要有好程序员。然而，对于大型软件的设计来说，不但要每个程序员具有必要的水平和技能，还要能够组成好的项目组。好的项目组是在长期工作中逐渐形成的。需要具备以下条件，1. 有严格的工作规范 2. 人员之间有严格的分工，必须有专门测试人员，有专职的项目负责人 3. 每个项目都要事先制定详细的时间表。每一项完成之后都有完整的资料

本章内容是软件工程与程序设计方法学的基础知识。这些知识构成了软件开发工具的基础，其中软件开发方法的发展为重点掌握部分。

第三章 软件开发工具的理论基础

思维导图



模块一 软件开发过程的信息需求

知识点一 软件开发过程的信息流通情况

(1) **信息流通模型**——描述了软件开发过程中,各种信息在各种人员(用户、分析人员、程序员、维护人员)与计算机之间的流通状况。

(2) **信息流通模型中涉及四类信息:**

- 第一类,有关环境、现状及需求的信息。
- 第二类,有关软件的功能设计与物理设计的各种信息。
- 第三类,软件成果本身,包括程序和文档。
- 第四类,用户对系统的各种变更要求,以及系统的各种变更记录(是跨越开发周期的信息)。

知识点二 软件开发过程中涉及的信息管理工作

(1) 信息的存储

(2) 数据的转换与加工:

- ① 手工形式和计算机形式之间的相互转换;
- ② 计算机内部不同形式数据的转换。

(3) 人与人之间的信息交流:分析人员、程序员、用户和维护人员

软件开发工具正是为前面提到的四类信息的合理存储、正确转化和顺畅流通提供帮助。

知识点三 软件开发工具用到的理论和方法

第一,认知科学中关于概念模式的概念与方法。

第二,数据库技术的理论与方法。

第三,编译技术的有关方法。

第四,关于人机界面的理论与方法。

第五,管理科学中关于项目管理与版本管理的理论与方法。

第六,系统科学与系统工程中的有关理论与方法。

模块二 概念模式

知识点一 概念模式

（一）定义

指人们在认识事物过程中，对某一事物或某一系统形成的抽象的、一般化的概念框架。

（二）意义

既反映了客观现实，同时又具有主观性

（三）作用

（1）概念模式是人们认识客观世界的一种方法和工具。

- ◆ 概念模式是人们在长期实践活动中逐渐形成的。人们认识事物的过程就是概念模式不断修正、不断完善的过程。
- ◆ 概念模式总是在一定的理论、思想指导下形成的，同时不断的根据实际情况得到修正。

（2）概念模式是人们进行交流与表达的重要工具。

人们在互相配合、互相协调、共同完成某项任务的时候，需要统一认识、协同行动，就必须用某种公用的方式进行表述，概念模式正起到了这种作用

知识点二 软件开发中常用的概念模式

概念模式	作用	基本图例	特点
框图	描述程序执行的逻辑过程	处理、输入/输出、判断、 起始/终止、控制转向	适用于软件规模不太大的情况
结构图	表示大型软件的层次结构，即 模块的结构。它以模块调用关系为线索，从宏观描述软件的全貌。	模块；顺序调用、循环调用、选择调用；参数传递	体现层次观点，由粗到细、自顶向下描述程序，体现结构化程序设计思想。
数据流程图（DFD）	描述某一业务领域处理	外部实体，信息处理，	把信息流看作一个组

	系统的信息来源、存储、去向的全面情况。	信息存储, 信息流动情况	织或系统运作的线索, 力图简明扼要地勾画出全局。
实体关系图 (ER)	用于描述静态数据结构, 广泛应用于数据库的设计中。	实体、关系、属性	以实体、关系、属性三个基本概念概括数据的基本结构。
数据字典 (DD)	描述数据内容的概念模式	二维表	用表格的形式列出数据的基本属性以及相互关系。
时序网络	用于描述系统状态及其转换方式	状态、转换	用于一些实时控制方面的软件功能的描述
数学与逻辑模型	描述客观世界的状态与规律。	决策表, 决策树	严格、准确、可计算; 限制多, 适于描述单纯、基础的功能。
计算机模拟模型	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, 以程序来表现客观系统中的某种规律与规则, 计算机可以在设定好的环境、按定好的规则高速运行, 以便人们观察与预测客观系统的状况。		

模块三 信息库及其一致性

知识点一 信息库的概念

- 信息库 (Repository) 又译为: 中心库、总库、主库, 原意就是数据库, 但不同于 Database 或 Data Warehouse
- 是针对软件开发或信息系统开发中的大量信息管理工作提出来的。是软件开发全过程的信息管理总枢纽。它的特点是数据结构相当复杂, 而且会不断变化, 保持一致性的任务十分艰巨。

知识点二 对信息库的研究

主要集中与三个方面:

1) 信息库的内容

(1) 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——调查记录、原始数据、报表及单证的样本、绘制的各种图、系统说明书。

(4) 设计阶段的各种资料——数据库与数据文件格式、数据字典、程序模块的要求、总体结构、各种接口及参数的传递方式、设计方案。

(5) 编程阶段的所有成果——程序代码、框图、变量说明、测试情况、验收报告、使用说明。

(6) 运行及使用情况的详细记录。

(7) 维护及修改的情况，包括修改的目标、责任人、过程、时间、修改前后的代码与文档以及修改后的结果、原系统的备份。

(8) 项目管理的有关信息，版本信息。

2) 信息库应当具备的管理功能

一般数据库的基本功能：增、删、改操作

特殊功能：

(1) 把计算机内外的信息存储统一起来管理——规定复杂的内部结构以存放信息。

(2) 面对分析人员、程序员和维护人员等不同的对象，他们有各自的权限和使用目标——需要认真设计有关的界面以便使用。

3) 信息库如何保持一致性

(1) 信息库中的内容一般不删除，老信息加上时间标志移入历史信息中，并与新信息保持着历史的、逻辑的联系，在需要时可以随时调出来。

(2) 由多个程序模块组成的大型软件系统中，如何保证某个模块更新后与其它模块一起构成一个新的版本。

(3) 保证程序和使用手册之间的一致性。

(4) 对国际性的大软件公司来说，各种语言的资料之间的一致性同样需要认真地、细致的组织与安排。

如何保持信息库的一致性，是信息库研究中的核心问题。

模块四 人机界面及其管理

知识点一 人机界面设计的基本原则

(1) 用户界面的主要功能是通信。

(2) 界面必须始终一致。

(3) 界面必须使用户随时掌握任务的进展情况。

- (4) 界面必须能提供帮助。
- (5) 宁可让程序多干，不可让用户多干

知识点二 AD/Cycle 应用系统框架 SAA 中，人机交互的手段

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

模块五 项目管理与版本管理

知识点一 项目管理

- 1) 概念：指与固定的生产线上的日常生产活动不同的、具有更大变动性、时间性的一类管理任务。
- 2) 一般项目管理的特点
 - (1) 子任务多，关系复杂。
 - (2) 任务不可重复，形势不断变更。
 - (3) 组织协调的任务十分突出，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。
- 3) 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) 有效的利用各种资源，尽可能使资源的闲置与浪费减少。
 - (4) 控制与降低成本。

知识点二 版本管理

1) 版本管理的核心——保持两个一致性

- 从时间上说，要保证系统的逐步完善，以前达到的结果不丢失，需要时可以回溯，必要时可以重用；每一步新的进步都要切实与以前的工作一致，是改进而不是破坏或者降低系统的已有功能。
- 从系统各部分之间的关系上说，要求局部与整体保持一致，合理地判断每一次修改的必要性及影响范围，合理地组成每一个新版本，并通过命名、编号

等方法，使新的版本成为系统进步的漫长过程中的一个台阶，一个有目的、有方向的稳定的进步。

2) 版本管理的具体方法

- 规定长远的版本更新计划
- 制定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
- 在编程、测试、文档编写等工作中贯穿项目的思想及要求。

本章题型分析

本章常考题型分为：单选题（常见分数占比为 1.50%）填空题（1.90%）简答题（7.00%）

1. (201210 单选题) 不属于软件开发工具概念模式的是 ()

A 数据流程图 B 实体关系图 C 关键路线图 D 数据字典

答案：C

解析：软件开发工具的概念模式包括框图、结构图、数据流程图、实体关系图、数据字典、时序网络、数学与逻辑模型、计算机模拟模型

2.(201604 填空题) 信息库中应保存编程阶段的所有成果，包括程序代码、框图、变量说明、测试情况、_____和使用说明。

答案:验收报告

解析：编码阶段指具体地编写软件的阶段，是实际的程序代码的产生点。信息库中应保存编程阶段的所有成果，包括程序代码、框图、变量说明、测试情况、验收报告和使用说明。

3.(201310 简单题)简述用户界面设计的基本原则。

答案：

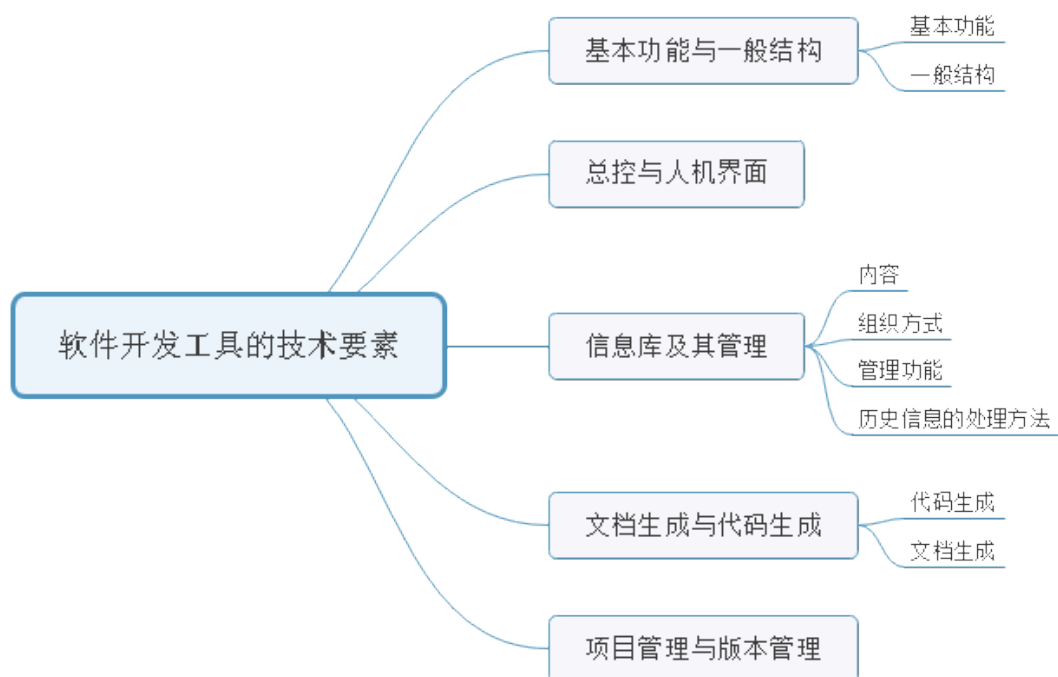
- (1) 用户界面的主要功能是通信
- (2) 用户界面必须始终一致
- (3) 用户界面必须使用户随时掌握任务的进展情况
- (4) 用户界面必须提供帮助
- (5) 宁可让程序多干，不可让用户多干

解析：对于用户来说，所关心得最主要的问题是自己能否能够有效地使用软件，而不是系统内部的算法或数据结构。人们已经公认，在当今的硬件与软件的环境下，如果一个软件没有很好的界面设计，就不能算是成功。关于用户界面的基本原则，以下几个方面：1.用户界面的主要功能是通信 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情况 4.用户界面必须提供帮助 5.宁可让程序多干，不可让用户多干

本章简略地介绍了软件开发工具的有关理论基础，包括概念模式、信息库、人机界面、项目管理等。其中概念模式及其作用为重点掌握部分。

第四章 软件开发工具的技术要素

思维导图



模块一 基本功能与一般结构

知识点一 基本功能

(1) 提供描述软件状况及其开发过程的概念模式, 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过程。

(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, 即代码生成。

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

知识点二 一般结构

(1) 总控和人机界面——中心位置, 使用者和工具间的桥梁, 工具实用性和灵活性的保证。

(2) 信息库和信息库管理模块——工具功能与作用的最基本依据。

(3) 文档生成和代码生成——最重要的两个信息出口

(4) 项目管理和版本管理——跨周期信息共享、知识重用、软件重用的关键问题。

模块二 总控与人机界面

知识点一 总控部分和人机界面重要性

无论是建立一体化的软件开发工具，还是把单项功能的工具集成起来，总控部分和人机界面都处于中心位置。

解决软件配置或集成问题的实质就是设立一个有效的总控部分，能够在各个具体工具之上实现信息的正确传递与转换，从而形成一个统一的、完整的支撑环境，并通过一个统一的、友好的人机界面与用户对话。

知识点二 总控部分及人机界面设计的原则

(1) 面向使用者（用户）的原则

(2) 保证各部分之间信息的准确传递

(3) 保证系统的开放性或灵活性——理想的系统结构应当是模块式的，易于剪裁的。

知识点三 IBM 的 AD/Cycle 框架

AD/Cycle 提供了开发过程模型的典型例子：

(1) 描述了一般的应用软件的结构 SAA——静态模型

构成	硬件	用户接口 CUA	程序员接口 CPI	通信接口 CCS
功能	平台	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	语言：过程化语言、非过程化语言、生成器 服务：数据库查询、对话显示、通信服务	面向对象的结构、数据流、应用服务、作业管理、网络管理、数据安全控制

(2) 提出了应用软件开发各个阶段的模型 AD/Cycle——动态模型

将软件开发过程的每个周期分为五个阶段：

- ① 需求分析，建立逻辑模型（企业模型，数据模型，信息流通模型）。
- ② 分析设计，完成系统的总体设计。
- ③ 编码，具体编写软件的阶段。
- ④ 测试，对已经完成的各个模块或子系统进行试算、调整，以最终形成完整的软件。
- ⑤ 运行维护，组织和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及时进行局部修改与完善。

模块三 信息库及其管理

知识点一 信息库的内容

- （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

知识点二 信息库的组织方式

组织方式	特点
集中存储	优点：充分利用已有的数据库技术，较容易保持一致性。 缺点：与模块化的要求相冲突。
分散存储	优点：易于变更和剪裁，一旦某一部分发生变更，不致影响整个系统。 缺点：每个模块都要自己进行一整套删改操作，各部分间的一致性无从保证。

逻辑上统一、物理上分散 (信息库管理模块)	通过信息库管理模块对分散的各数据库进行存取, 并进行一致性检查与维护, 使数据库与模块都实现了模块化, 保持了一致性和灵活性。
--------------------------	---

知识点三 信息库的组织方式

与一般数据库管理系统相同的功能: 录入更新、使用查询、一致性维护。

与一般数据库不同的管理功能:

- (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。

知识点四 信息库的组织方式

信息库管理中历史信息处理是一个重点和难点, 主要存在两个困难:

- (1) 历史信息的数据太大, 占用存储设备过多;
- (2) 历史信息格式不一致, 难以有效利用。

第一个问题的解决一般采用脱机备份的方法。

第二个问题至今尚无令人满意的解决办法, 加强标准化, 稳定数据结构不能完全解决这个问题。需要用一些智能化方法, 使得跨生命周期的信息得以相互转化、实现共享与重用。

模块四 文档生成与代码生成

知识点一 代码生成

- 1) **代码生成器**——根据设计要求, 自动或半自动地生产相应的某种语言程序(某种高级语言代码和某种机器环境下可运行的机器指令)。
- 2) **生成代码是依据三方面的资料**

- 信息库里已有的有关资料
- 各种标准模块的框架和构建件
- 使用者通过屏幕前的操作送入的信息

知识点二 文档生成

文档生成的功能与代码生成相比，数量更大，内容更复杂。所谓文档生成包括文章、表格、图形三大类。

其中，其中最容易生成的是表格，其次是图形，最难处理的是文章。

模块五 项目管理与版本管理

知识点一 项目数据库

针对项目管理在信息处理上的困难，提出以项目数据库为中心解决问题的思路。项目数据库是信息库的一部分，记录项目本身进展的各种有关信息，如项目的预期进度、实际进展、验收情况等。

知识点二 项目管理范围

(1) 研究与确定开发工作的方针与方法。例如，采用什么样的阶段划分方法，什么样的系统描述方法等。

(2) 开发任务的划分与分工。整个开发任务如何逐层分解为具体任务，这些任务委托给什么人（或哪个组）来做，这些任务之间的相互关系（顺序、制约等）。

(3) 资源情况。有多少人力、物力、设备、软件、资金，是否得到充分利用，是否需要增加投入。

(4) 人员情况。个人进度、技术水平如何，是否需要重新培训，是否需要调动

工作。

(5) 变更情况。需求、环境、人员、设备(包括硬件和软件)、技术的变更。

(6) 质量情况。检验得标准是什么,如何检验,由谁来检验,用什么数据检验。

知识点三 版本管理的主要内容

主要包括:各版本的编号,功能改变,模块组成,文档情况,推出时间,用户数量,用户反映,封存情况等

本章题型分析

本章常考题型分为:单选题(常见分数占比为2.40%)填空题(2.60%)简答题(4.30%)

1. (201310 填空题) 文档生成包括文章、_____和图形三大类。

答案:表格

解析:文档生成的功能与代码生成相比,数量更大、内容更复杂,包括三大类:文章、表格、图形。其中,最容易生成的是表格,其次是图形,最难处理的是文章

2. (201310 单选题)在信息库的管理方面,目前比较好的管理方式是

A 逻辑上、物理上都统一 B 逻辑上、物理上都分散

C 逻辑上统一、物理上分散 D 逻辑上分散、物理上统一

答案:C

解析:逻辑上统一,物理上分散,设立统一的信息库管理模块来进行管理。通过信息库管理模块来对分散的各个数据库进行存取,并进行一致性的检查与维护。这样一来,数据库与模块都实现了模块化,既保持了一致性,又保持了灵活性。

3. (201404 简答题) 简述信息库中保存的版本管理信息的内容。

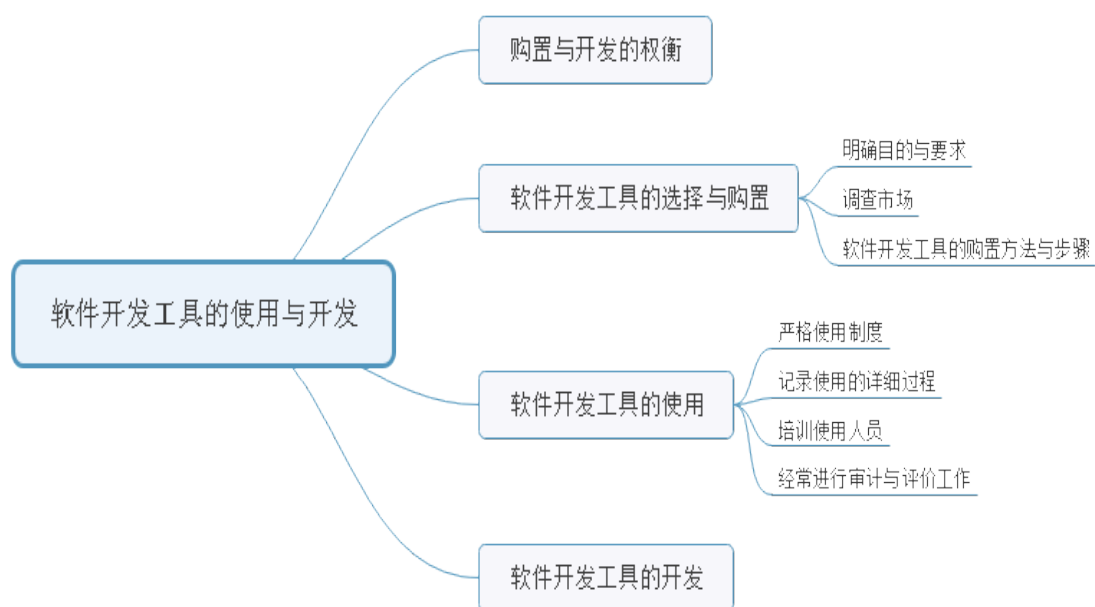
答案 :包括各个版本的备份、每个版本的推出日期、与以前版本相比的变更说明。

解析 :关于版本的信息,有的工具中把它作为项目数据库的一部分来处理。它的主要内容是包括各个版本的备份、每个版本的推出日期、与以前版本相比的变更说明。

本章综合了常见的软件开发工具的一般情况,介绍了软件开发工具中的各方面技术状况。其中总控和人机界面,信息库及其管理为重点掌握知识点。

第五章 软件开发工具的使用与开发

思维导图



模块一 购置与开发的权衡

知识点一 购置与开发的权衡的四要素

- (1) 准备开发的软件的性质与要求---权衡购置还是自行开发的最基本因素
- (2) 开发人员对支持工作与支持程度的实际需要
- (3) 工作环境(硬件配置、操作系统、数据库管理系统、网络通信等)
- (4) 人员的因素

许多软件技术人员手边都积累了一些自制的、专用的、规模不一的软件开发工具。

模块二 软件开发工具的选择与购置

知识点一 明确目的与要求

在选择软件开发工具之前,首先要清楚此次引入工具要达到怎样的目标,下面几点必须明确:

- (1) 为哪个软件开发项目而使用工具,为了一个项目还是为了一批项目而购置软件开发工具,为哪一种类型的项目而购置这些工具?
- (2) 在哪个工作阶段使用工具,是单在代码生成或文档生成中使用工具,还是准备长期开发、不断更新版本?
- (3) 工具将供哪些人使用,这些人以前用过工具没有,他们熟悉的语言、开发环境、表达图示、交流方式是什么?这些人对于使用工具的态度、期望、可塑性如何?
- (4) 工具将在怎样的软件、硬件环境下运行,机器的速度、内存、外围设备、

通信条件如何？

知识点二 调查市场

- (1) 工具的功能。
- (2) 工具的性能。
- (3) 工具所使用或依据的开发方法或开发理论与自己的方法、理论是否一致。
- (4) 工具的运行环境。
- (5) 这些工具的文档资料是否齐全？使用时的学习、掌握是否简单易行？
- (6) 这些工具的服务、培训条件如何？
- (7) 价格

知识点三 软件开发工具的购置方法与步骤

- (1) 明确购买工具的目的与要求---正确使用工具的基础工作。
- (2) 明确购买工具的环境条件与制约条件。
- (3) 市场调查。
- (4) 对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。包含了技术洽谈和各种服务条件的确认。
- (7) 安装与试用。

模块三 软件开发工具的使用

知识点一 组织管理工作

购置工具后，使用者必须对它的使用过程进行认真地组织与管理，主要包括四个方面：

- (1) 制定严格的使用制度。
- (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。

知识点二 审计

审计---对一个系统的运行状况及效率进行检测与评价，以便进一步用好这个系统。审计的基础就是日常记录的信息，没有日常信息的积累，审计工作就无法进行。

知识点三 审计的范围

包括工具的使用环境、人员、工作负担、效果、存在的问题、改进的方向等。

模块四 软件开发工具的开发

知识点一 注意点

首先要区分是为自己开发还是作为商品开发，至少在一个时期内二者必居其一。

1、为自己使用而开发软件工具时应注意的问题

- (1) 需要从实际出发，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) 要注意文档的齐全与资料的积累。

2、开发商品化的软件工具时应注意的问题

开发面向实际应用领域的、商品化的软件工具，其难度要比为自己开发工具的难度大得多。

- (1) 必须十分谨慎地设定目标和要求，使之符合实际。

(2) 在前人工作的基础上,把精力集中于进一步的开拓方面,而不是低水平的重复别人已经做过的工作。

(3) 明确工具的适用范围与使用条件。工具的目标越具体、越明确越能发挥作用。

(4) 注重服务与文档—决定其成败的首要因素。

本章题型分析

本章常考题型分为:单选题(常见分数占比为4.30%)填空题(1.50%)简答题(1.70%)

1. (201404 填空题) 购买现成的软件开发工具,还是自己开发专用的工具与具体工作的条件、环境、_____,项目特征都有不可分割的联系。

答案:人员素质

解析:在选择软件开发工具之前,首先要清楚此次引入工具要达到怎样的目标,购买现成的软件开发工具,还是自己开发专用的工具与具体工作的条件、环境、人员素质,项目特征都有不可分割的联系。

2. 软件开发工具性能审计不包括()

A 工具的效率 B 工具能否生成代码 C 工具的响应速度 D 工具的输出方式

答案:B

解析: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。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。其中包括这个工具在效率、响应速度、输出方式等性能方面能否满足本项目组实际的工作要求。

3. (201110 简答题) 在引入软件开发工具之后 , 如何严格使用制度 ?

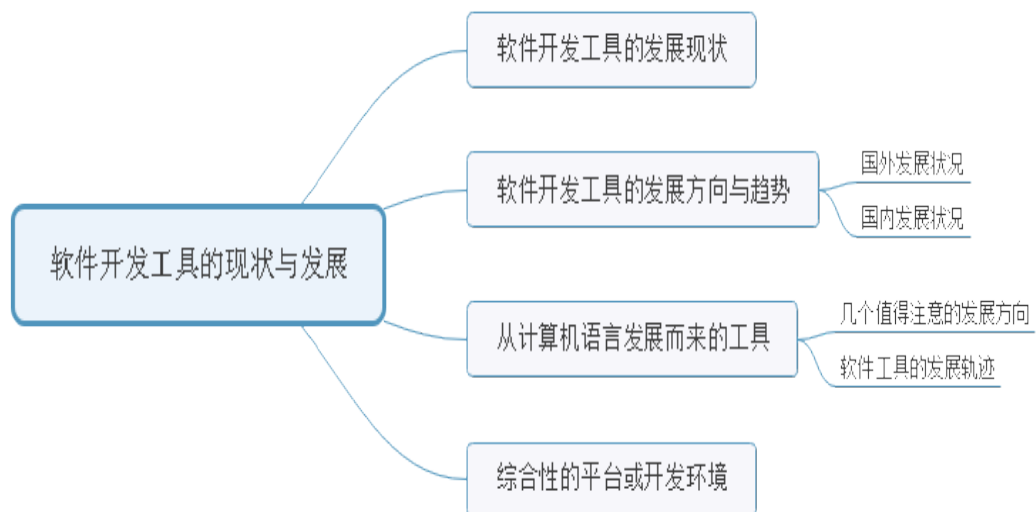
答案 : 在信息库要明确按时填写与本软件开发有关的各种信息来源 , 即由哪些人在什么时候完成这种任务、规定成员的使用权限、维护职责等有关事宜。

解析 : 在引入软件开发工具之后 , 应当认真进行的组织管理工作包括 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 , 其中严格使用制度需要明确信息来源 , 规定什么人完成何种任务 , 规定使用权限及维护责任

本章具体问题是人们是使用还是自行开发软件开发工具 , 其中购置与开发平衡为重点内容。

第六章 软件开发工具的现状与发展

思维导图



模块一 软件开发工具的发展现状

软件开发工具的兴起是在 20 世纪 80 年代中期。

知识点一 国外发展状况

1) 专项工具的出现 (如 : 管理数据字典的 CDD、Dictionary/3000 ; 画流程图的 Flow ; 依赖各种机器的代码生成器和第四代语言)

2) 一体化工具的出现

标志 : IB M 的 AD/Cycle 框架。到 1992 年 , 已 有 30 余种产品推向市场。

目前真正一体化的 , 又能应用于各种平台的工具尚不多。比较多的有两种工具 :

- 用于特定平台的、主要用于设计阶段的工具——这种软件多依赖于软件平台 , 如某种数据库管理系统 : o RACLE 的 CASE , IN FOMAX 的 New Era , IB M 的 VisualAge , PowerBilder , PowerHouse
- 侧重于分析方法的、独立于平台的工具——从方法论角度看 , 这类工具更值得注意。如 : Rational 的 Rose , Mark V 在美国国防部系统得到广泛应用
- 基于互联网和开源软件的工具也值得关注。如 : 帮助搜索和检索开源软件的工具 , 自动生成和检查 XML 的工具 , 基于 SOAP 的工具 , 面向 UDDL 的搜索工具

知识点一 国内发展状况

从研究水平上看 , 国内并不比国外低 , 主要的差别在于没有广泛地使用这些工具。总的来说 , 我国的软件产业还没有形成一批稳定的、高度有组织的软件开发群体 , 因而能够真正发挥软件开发工具作用的环境与社会需求还不具备。

我国软件产业当务之急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 , 这才能使软件开发工具的应用具有实践的基础。

国内软件开发工具的代表：

★ 专项工具如：清华大学 AutoDBaseⅢ和许多报表生成器、自动屏幕设计工具
自动菜单设计工具

★ 一体化工具如：Quick MIS

模块二 软件开发工具的发展方向与趋势

知识点一 几个值得注意的发展方向

(1) 智能化——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信息、模糊信息具有更强的处理能力。由于在软件开发工作中，存在着大量不确定的因素，人们常常需要用知识与经验来补充加工。

如：Knowledge Ware 公司开发的以知识处理为基础的工具。

(2) 网络化——网络使人们更便于掌握项目的进展状况、质量状况等。利用网络提供条件提高工作效率的软件开发工具，以及在网络上开发应用程序的工具是当前发展的热点。通过网络。人们可以更方便地互通信息，共享知识，这就给人们所梦想的软件重用、知识重用提供了信息机会。

如：专门用于网络环境的 HTML、Java

(3) 一体化——一体化的趋势早在 20 世纪 80 年代后期已经十分明显。只有对软件开发中涉及的各种信息以及开发过程中它们的发生、变化、关系、一致性等有了深刻的理解，才能真正实现软件开发工具的一体化。信息库的内容和处理功能的讨论，从理论上为实现一体化提供了基础。

(4) 标准化——由软件部件、组合软件的想法引起的。为了解决大型软件开发

的困难，需要把软件开发工作分成两部分：软件构件（如硬件的芯片）的开发，用这些构件组成大型软件（如用芯片组成主机）。做到这一点，软件构件必须实现标准化，用构件组成的大型软件的结构也必须符合一定的标准。围绕这一思想提出的一些标准如：CORBA、COM101

知识点二 软件工具的发展轨迹

软件开发工具 40 年来的发展轨迹，经历三个时代。

从结构化时代，经过面向对象的时代，到今天的互联网时代。从中可以确认和佐证软件开发工具的目标和作用是什么？我们需要软件开发工具，就是要更快更好地开发软件，就是为了提高软件开发的效率和质量。

“软件开发工具”包含了从非常具体的一些计算机语言到内容丰富、功能齐全的开发环境的相当广泛的庞大谱系。

- 谱系的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；
- 另一端是从较为抽象的概念模式或过程模式出发设计的开发平台或开发环境。

模块三 从计算机语言发展而来的工具

知识点一 注意点

较早的软件开发工具（谱系左端）基本是基于某种计算机语言，通过增添各种辅助功能发展出来的，它们更多地针对计算机语言特性进行了支持，以最大限度提高相应的计算机语言开发效率为依据，可以理解为计算机语言在开发方向上的一个延伸。

典型代表：VB、VC、PB、C++ Builder/Delphi、JAVA

本时期的工具基本着眼于某种具体语言本身，对 软件开发过程中涉及的交互、文 档管理、代码版本管理的支持略显不足。

模块四 综合性的平台或开发环境

知识点一 Visual Studio 与 Eclipse 比较

在大中型项目中，由于项目复杂，开发周期长，交互性高，大多数开发团队都倾向于使用集成开发环境(IDE)，其中比较受欢迎的软件开发环境有 Visual Studio（微软）和 Eclipse。

1) Visual Studio 的特点

Visual Studio 与微软公司其他产品的交互协同浑然一体，天衣无缝，并获得了更多 Windows 操作系统级的支持，速度上比 Eclipse 有较大的优势。如果基于操作系统是 Windows，开发中又需要使用大量的微软公司的产品，那么可以优先考虑 Visual Studio。

2) Eclipse 的特点

Eclipse 是一款免费的、面向各平台开发者的软件开发环境，在各种操作系统上表现差异非常小。安装后的核心部分大小只有数十兆，开发者可以根据需求再添加同样免费的插件。Eclipse 具有“大平台，小核心，多插件”的特点，更富有灵活性，如果开发成果最终需要被部署在非 Windows 平台上而又希望开发环境最大程度模拟运行环境，使用 Eclipse 能够打消来自操作系统方面的忧虑。

本章题型分析

本章常考题型分为：单选题（常见分数占比为 1.70%）填空题（1.30%）简答题（3.20%）

1.(201404 单选题) 与 Eclipse 比较，Visual Studio 的独特优势是

A 具有编译、运行功能

B 运行速度快

C 具备了代码援助、语法高亮、错误预防功能

D 令开发人员把精力集中到代码逻辑和算法优化上

答案：B

解析：Eclipse 最初主要用来进行 Java 语言开发，此外还支持如 C/C++、C#、PHP、J2EE、Javascript 等编程语言的插件，如 CDT。与 Eclipse 比较，Visual Studio 的独特优势是运行速度快

2.（201604 填空题）从目前软件开发工具的发展势头看，智能化、一体化、网络化、_____是值得重视的几个动向。

答案:标准化

解析：智能化：具体地说就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的信息、模糊信息具有更强的处理能力。一体化：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网络化：通过网络，人们可以更方便地互通信息，共享知识

标准化：标准化的问题是由软件部件、组合软件的想法引起的。

3. (201210 简答题) 简述软件开发工具的智能化的含义。

答案：在软件开发工具的研究和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信息和模糊信息具有更强的处理能力, 提高信息处理的功能与效率。

解析：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。所谓智能化, 具体来说就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信息和模糊信息具有更强的处理能力, 提高信息处理的功能与效率。

本章主要为软件开发工具技术发展现状, 其中软件开发工具的发展方向与趋势为重点掌握知识点。