

全国高等教育自学考试指定教材辅导用书

高等教育自学考试同步辅导/同步训练

软件开发工具

杨连贺 主编

1 56



人事出版社

全国高等教育自学考试指定教材辅导用书
高等教育自学考试同步辅导/同步训练

软件开发工具

全国高等教育自学考试指定教材辅导用书编委会 组编

主 编：杨连贺

副主编：吴中元

杨胜友

中国人事出版社

图书在版编目(CIP)数据

软件开发工具/杨连贺主编. —北京:中国人事出版社,1998.11

全国高等教育自学考试同步辅导/同步训练(计算机信息管理专业)

ISBN 7—80139—314—7

I. 软… II. 杨… III. 软件工具—高等教育—自学考试—自学参考资料 IV. TP311.56

中国版本图书馆 CIP 数据核字(98)第 32835 号

版权所有,翻印必究。

本书封面贴有防伪标签,无标签者不得销售。

(如有缺页和倒装,本社负责退换)

说 明

为了满足高等教育自学考试社会助学和适应考试的需要，我们组织了高等院校的部分专家学者结合自学考试的特点，编写了这本《软件开发工具》同步辅导/同步训练。

编写依据：

(1) 严格遵照全国高等教育自学考试指导委员会颁布的《软件开发工具自学考试大纲》的命题原则和命题范围。

(2) 以全国高等教育自学考试统编教材《软件开发工具》(经济科学出版社出版，陈禹、方美琪主编)为编写依据。

(3) 以分析研究历年考试试卷为基础。

本书特点：

本书以自学考试大纲规定的考核知识点及能力层次为线索，按照大纲和统编教材的顺序分章进行辅导，覆盖了大纲要求的全部考核内容。全书以章为单位，将自学考试中每一章节可能出现的所有考核知识点汇总在要点·难点解析中，并按照考试题型编写同步练习题，同时给出参考答案，最后附模拟试题三套。

本书的每章均先将要点、难点知识一一列出，然后以同步练习的形式加以巩固，为考生系统地学习教材提供了全面的配套练习。为便于学员查阅答案，各章练习后均附有参考答案。为使考生巩固所学知识，提高应试能力，并检验复习效果，书后还附了三套模拟试题，其目的是为了帮助考生全面掌握知识，顺利通过自学考试。

为了加深读者应试前的复习印象，每章都有少量的重点内容以不同的题型重复出现。

由于水平有限，成书仓促，不当之处在所难免，望广大读者、学者批评指正。

本书供高等教育自学考试个人自学、社会助学和参加国家统一考试使用，无疑也适用于其它相同专业方向的学习。

编 者

1998年10月

目 录

第一章 绪 论	(1)
要点·难点解析	(1)
同步练习	(7)
参考答案	(12)
第二章 软件开发过程及其组织	(18)
要点·难点解析	(18)
同步练习	(23)
参考答案	(27)
第三章 软件开发工厂的理论基础	(35)
要点·难点解析	(35)
同步练习	(40)
参考答案	(45)
第四章 软件开发工具的技术要素	(55)
要点·难点解析	(55)
同步练习	(59)
参考答案	(62)
第五章 软件开发工具的使用与开发	(69)
要点·难点解析	(69)
同步练习	(70)
参考答案	(73)
第六章 软件开发工具的现状与发展	(79)
要点·难点解析	(79)
同步练习	(79)
参考答案	(81)
附录1 Rational 公司的软件工具 Rose	(85)
要点·难点解析	(85)
同步练习	(86)
参考答案	(88)
附录2 一个实用的 CASE 工具——RDCASE	(90)
要点·难点解析	(90)
同步练习	(91)
参考答案	(92)
附录3 PowerBuilder	(94)
要点·难点解析	(94)
同步练习	(94)
参考答案	(97)
附录4 Mark V 系统	(100)

要点·难点解析	(100)
同步练习	(102)
参考答案	(103)
模拟试题 (一)	(106)
参考答案	(109)
模拟试题 (二)	(111)
参考答案	(114)
模拟试题 (三)	(116)
参考答案	(118)

第一章 绪 论

要点·难点解析

《软件开发工具》课程并非介绍几种常用的软件开发工具。相反，它介绍的是软件开发工具的理论。它使学员不只是站在一般程序员的角度，而是从软件开发的组织者的高度，全面地认识与考虑软件开发的过程、组织等问题。

本章概括地介绍了软件开发工具的由来、概念、功能、性能、类别与使用环境。通过这一章的学习，学员应当不只是具备程序员的水平，而应当超出程序员的水平，即从单纯的个人编写程序的角度，提高到从软件开发的历史，从有组织的、大规模的生产角度来看待自己的工作。只有这样才能理解软件开发工具产生的必然性，才能认识软件开发工具的价值与作用。

因此，本章涉及的内容较为广泛，既包括历史的回顾，又包括软件开发的组织与管理。对于具有一定实践经验的学员来说，应当结合自己的经历，进行总结与提高。对于缺乏实践经验的学员来说，会感到比较抽象，因此需要通过一些实例的介绍或访问有经验的人员，以便补充一定的感性知识。

作为导言，本章涉及的内容将在以后各章中展开。学员应当在本章学习结束时对全课程内容的框架有所了解，以便更好地进行后面各章的学习。

【本章要点】

1. 软件开发工具的由来

1.1 软件开发工具的界定

软件开发工具的范围大致可以描述为：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。

对于本概念可以从以下几个方面去理解。首先，它是一类软件，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，这里强调对软件开发全过程中各个阶段的支持。

1.2 软件的产生（直接面向硬件的软件开发）

计算机刚刚问世时，人们面对的是只能执行机器指令的硬件设备，即所谓的“裸机”。机器运行时需要人们输入0和1组成的机器码，即机器语言程序。这个阶段称为机器语言阶段，所用的语言称为第一代语言。

很明显，机器语言的使用是十分艰难的。针对难以记忆的0、1字符串，人们试图用英语单词的缩写来代替它，这就是助记符，或汇编码。用汇编码编写程序代替用机器码编写程序，代表了改进计算机应用的基本方向：建立一些专用的“工具”，使某些可以由机器来完成的信息处理工作交给计算机去做，而使得人们的知识、经验转化为计算机的操作。

几乎与汇编语言同时，操作系统的出现从另一方面而改善了人们应用计算机的条件，把原先由操作员担负的大部分职责接过来，改善了计算机使用的环境，也同样为计算机应用的前进发挥了十分重要的作用。

1.3 程序设计语言的产生

早在 60 年代初, 高级程序设计语言就开始成熟与普及, 标志着计算机开始了向社会生活各个领域的全面渗透。高级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。另一个重要的变化是高级程序设计语言实现了对机器的独立性, 从而大大提高了程序的可移植性。因而在短短的二、三十年中, 计算机技术已经深入到了各行各业, 获得了广泛的应用。

1.4 软件开发工作的困难

第三代语言虽然获得了相当广泛的应用, 但人们很快又发现了新的瓶颈——处理过程的描述。原来, 第三代程序设计语言一般都是过程化语言, 需要由程序员一步步安排好机器的执行次序。人们希望机器能够自动地完成更多的工作, 包括自动安排某些工作的顺序, 而做到只要给机器下达做什么的命令, 由机器自己去安排执行的顺序。这就是第四代语言——非过程化语言的思想。另一方面, 从六十年代末期开始, 人们对于软件工作的认识大大深入了。在认识到软件工作的重要性的同时, 也认识到了软件工作的困难性, 即所谓的“软件危机”问题。

1.5 软件开发过程中使用的工具

1.5.1 通用工具的应用

在 70 年代末, 80 年代初, 人们想到了用通用软件来进一步支持软件开发工作, 如编写文档, 绘制流程图等。这时所用的都是通用软件, 故这个阶段可以称为利用通用软件作为辅助工具的阶段。

1.5.2 专用的软件开发工具的出现

用通用软件来帮助软件开发人员编写文档或画图可以减少不少工作量, 但是这种帮助实在是太表面、太初步了。用通用软件帮助软件开发有三个主要的弱点。第一, 有许多工作是通用软件所无法完成的。第二, 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。第三, 用通用软件来帮助人们完成软件开发工作时, 常常遇到难于保持一致性的困难。针对这些问题, 八十年代以来, 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。这期间涌现出许多软件开发工具。在这些工具的推动下, 软件开发工具的有关理论也逐渐发展起来, 并引起了软件界的重视。软件开发工具或开发环境的概念越来越为人注目。

1.5.3 一体化的软件开发环境的出现

专用软件开发工具的出现, 并未解决上面提到的第三个困难, 一致性的保持仍然是悬而未决的难题。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。这些零散的、分散地支持各个工作阶段、各项具体工作的专用工具之间没有有机地联系起来, 从而必然造成冲突与矛盾。这种冲突与矛盾对于用户来说, 造成了沉重的、不堪忍受的负担, 使用工具越多这种负担越重, 以致抵消了使用工具带来的益处。这种情况导致了集成的软件开发工具的产生。

2. 软件开发工具的概念

2.1 软件开发工具的概念

软件开发工具概念的要点是:

- (1) 它是在高级程序设计语言 (第三代语言) 之后, 软件技术进一步发展的产物;
- (2) 它的目的是在人们开发软件中给予各种不同方面、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他稳定的工作阶段。

2.2 深入理解软件开发工具的概念

2.2.1 对软件实质的再认识

软件是广大使用者与计算机之间的桥梁,软件是人类在各个领域中积累的知识的结晶,软件是人类文明与知识得以延续的新的载体,软件是人类进一步成为一个整体,得以进一步相互联系的纽带。软件一头面向计算机硬件,提供可执行的机器指令,一头面向应用领域(即用户),接受所要求的信息处理业务。这种知识的提炼、表述、固化的作用,正是软件的实质所在。

2.2.2 软件开发工作的发展过程

软件开发工作的发展过程可以归纳为四个不同的阶段。

(1) 用机器语言开发软件

把用户已经明确表述出来的算法,用机器语言写成一系列机器指令,供硬件运行使用。这是人们对软件开发工作的最初的认识。

(2) 用汇编语言开发软件,由汇编程序完成转换。

汇编语言产生之后,编程工作改为用汇编语言进行,编好的汇编指令由汇编程序转化为机器指令,再交硬件执行。软件开发工作离硬件远了,离用户近了。

(3) 用高级语言开发软件,由编译程序完成转换

高级程序设计语言不再与机器指令一一对应,而且更加接近人类习惯的自然语言。软件开发离机器更远了,离用户更近了。

(4) 在各种软件开发工具的帮助下开发软件,由编译程序完成转换。

这是软件开发工作发展的新阶段,也使软件开发工作的概念从一次性的、具体的编程,扩大到了长期的、不断更新的过程。

2.2.3 现阶段软件技术的特点

(1) 自动化程度的提高。由于代码生成等技术的应用,在一些特定的条件下,可以较容易地自动生成第三代语言(或更低级的语言)的代码,从而大大节省人力和时间。

(2) 明确地把需求分析包括进了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,离用户更近了。这是软件技术向前迈进的重要一步。

(3) 把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) 在这一阶段的研究中,吸收了许多管理科学的内容与方法,如程序员的组织、质量的控制等。

2.3 对软件开发工具的不同理解

与软件开发工具有关的概念、术语很多,如第四代语言、CASE工具、可视化程序设计、最终用户计算、组件程序设计,即插即用的程序设计等等。

第四代语言(4GL)是应用较为广泛的一个名词,它的原义是非过程化的程序设计语言。CASE工具有两种理解,一种是计算机辅助软件工程,另一种是计算机辅助系统工程。两者的缩写都是CASE,其基本思想与软件开发工具是完全一致的。至于其他几个名词,如可视化编程、最终用户计算、组合编程、即插即用编程等,无非都是软件开发工具范围内的某种思想或某种趋向,与软件开发工具的思想是一致的。

3. 软件开发工具的功能与性能

3.1 软件开发的过程

3.1.1 需求分析阶段

根据初始要求形成严格的、明确的、可供实际开发使用的软件功能说明书。

3.1.2 总体设计阶段

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几个主要内容。

3.1.3 实现阶段

完成程序的编写与文档的编写。

3.1.4 调试阶段

进行模块的调试与整个软件的联调。

3.2 软件开发工具的基本功能

3.2.1 描述系统

主要是用在软件开发工作的第一个阶段——需求分析阶段。需求分析工作不确定程度高，更需要经验，更难形成规范化，因而人们希望软件开发工具能在这方面给予更多的帮助。

3.2.2 管理信息

在软件开发的各阶段都要产生及使用许多信息。有些信息到了其它阶段还要使用，故信息保存与管理问题显得非常突出。

3.2.3 生成代码

在整个软件开发工作过程中，程序编写工作占了相当比例的人力物力和时间，提高代码的编制速度与效率显然是改进软件工作的一个重要方面。

3.2.4 生成文档

文档编写工作也是软件开发工作中十分繁重的一项工作，不但费时费力，而且很难保持一致。在这方面，计算机辅助的作用可以得到充分的发挥。

3.2.5 项目管理

这一功能是为项目管理人员提供支持，包括进度管理，资源与费用管理，质量管理。

3.3 软件开发工具的性能指标

3.3.1 表达能力

因为软件项目的情况千变万化，软件开发工具要能够适用于某些软件项目，就要能适应软件项目的种种不同的情况，否则就不可能对软件开发提供有效的、实际的帮助。

3.3.2 逻辑能力（一致性的保证）

软件开发工作要求软件开发工具不但要为人们存储大量的有关信息，而且要有条不紊地管理这些信息，而管理的主要内容就是保持它的一致性，至少在出现不一致的情况时要能够给出警告与提示。

3.3.3 人机界面水平（易用程度）

对于软件开发工具来说，是否易用是一项重要的性能指标。

3.3.4 防错能力（可靠性）

软件开发工具应当具有足够的可靠性，即在各种各样干扰下仍能保持正常工作，而不致丢失或弄错信息。

3.3.5 对硬软件环境的要求

软件开发工具对环境的要求不应当超出它所支持的应用软件的环境要求，有时甚至还应当低于应用软件的环境要求。软件开发工具的环境要求应当尽量降低，以有利于广泛使用。

4. 软件开发工具的分类

4.1 按支持的工作阶段划分

4.1.1 设计工具:是最具体的工具,在实现阶段对人们提供帮助。

4.1.2 分析工具:主要用于支持需求分析。

4.1.3 计划工具:从更宏观的角度去看待软件开发。

4.2 按集成程度划分

4.2.1 单向工具:专用的、面对某一工作阶段或某一工作任务的工具。

4.2.2 集成工具:集成化的、面对软件开发的全过程的工具。

4.3 按与硬件、软件的关系划分

4.3.1 依赖于特定机器或软件的工具。

4.3.2 独立于机器与软件的工具。

5. 软件开发工具的研究与使用

5.1 软件开发工具的研究与使用

5.1.1 需求状况

5.1.2 设备状况

5.1.3 人员状况

5.1.4 社会环境

5.2 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,以便能够在实际工作中正确地选择与使用软件开发工具,在必要时能够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之目标。这就是学习软件开发工具的目的。

【考核重点(最高能力层次)】

1. 软件开发工具产生及应用的必然性及发展趋势

这一要点可从软件及软件开发工具发展的历史去理解。

软件开发工具是计算机技术发展的产物。随着以计算机为代表的现代信息技术迅速地应用到社会生活的各个角落,社会对于各种软件的需求也日益紧迫。各行各业都要求软件开发人员迅速地、高质量地提供各种各样的软件产品,而手工编程进行软件开发难以满足这一要求。作为对策之一,软件开发工具应运而生。它以计算机自身处理信息的能力为基础,在软件开发的各个阶段,对软件开发的各个方面提供了各种各样的帮助,从而成为今天的软件工作人员必须具备的重要技术手段。

与其它新技术一样,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。回顾它的产生与发展的经过,就能很自然地理解它产生的必然性。

(1) 机器语言阶段 以0和1两个字符编程,非常繁琐。

(2) 汇编语言阶段 以助记符代替0、1编程,效率大大提高,但仍依赖于机器。

(3) 高级语言阶段

高级语言突破了与机器指令一一对应的限制,实现了对机器的独立性,从而大大提高了程序的可移植性。高级程序设计语言使计算机的应用大大普及了。

(4) 第四代语言阶段

实现了非过程化的处理思想。

从六十年代末开始，人们在认识到软件工作的重要性的同时，也认识到了软件工作的困难性，从而提出了软件危机问题。围绕这个问题，人们从开发方法的研讨、语言的改进、人员的组织等各方面想了许多方法，先后提出结构化程序设计、软件工程、面向对象的设计方法及即插即用的程序设计方法等，许多专家作了大量研究工作，涌现了许多基于第三代语言的软件开发工具。

(5) 利用通用软件作为辅助工具的阶段

在 70 年代末，80 年代初，人们用通用软件来进一步支持软件开发工作。这时所用的都是通用的、并非专门为软件开发工作而设计的软件，故这个阶段可以称为利用通用软件作为辅助工具的阶段。

(6) 专用软件开发工具阶段

用通用软件来帮助软件开发人员编写文档或画图可以减少不少工作量，但是与整个软件开发工作相比，这种帮助实在是太表面、太初步了。用通用软件帮助软件开发有三个主要的弱点。第一，有许多工作是通用软件所无法完成的。第二，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。第三，用通用软件来帮助人们完成软件开发工作时，常常遇到难于保持一致性的困难。针对这些问题，八十年代以来，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。这期间涌现出许多软件开发工具。在这些工具的推动下，软件开发工具的有关理论也逐渐发展起来，并引起了软件界的重视。软件开发工具或开发环境的概念越来越为人注目。

(7) 集成的软件开发工具阶段

专用软件开发工具的出现，并未解决上面提到的第三个困难，一致性的保持仍然是悬而未决的难题。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。这些零散的、分散地支持各个工作阶段、各项具体工作的专用工具之间没有有机地联系起来，从而必然造成冲突与矛盾。这种冲突与矛盾对于用户来说，造成了沉重的、不堪忍受的负担，使用工具越多这种负担越重，以致抵消了使用工具带来的益处。这种情况导致了集成的软件开发工具的产生。

回顾软件开发工具从无到有，从分散到集成的发展过程，可以清楚地看到软件开发工具的出现决不是偶然的，它是软件发展的必然趋势，是软件技术发展的一个新的阶段。

2. 软件与软件开发工作的实质

这一要点可从软件及软件开发工作的发展变化去理解。

软件开发工作的发展变化可以归纳为四个不同的阶段。

(1) 用机器语言开发软件

把用户已经明确表述出来的算法，用机器语言写成一系列机器指令，供硬件运行使用。这是人们对软件开发工作的最初的认识。

(2) 用汇编语言开发软件，由汇编程序完成转换

汇编语言产生之后，编程工作改为用汇编语言进行，编好的汇编指令由汇编程序转化为机器指令，再交硬件执行。软件开发工作离硬件远了，离用户近了。

(3) 用高级语言开发软件，由编译程序完成转换

高级程序设计语言不再与机器指令一一对应，而且更加接近人类习惯的自然语言。软件开发离机器更远了，离用户更近了。

(4) 在各种软件开发工具的帮助下开发软件，由编译程序完成转换

这是软件开发工作发展的新阶段，也使软件开发工作的概念从一次性的、具体的编程，扩大到了长期的、不断更新的过程。这就引出了项目管理、版本控制、软件重用等一系列新问题。

从上述软件及软件开发工作的发展变化，可以体会其中贯穿始终的基本线索，那就是一头面向计算机硬件，提供可执行的机器指令，一头面向应用领域（即用户），接受所要求的信息处理业务。这种知识的提炼、表述、固化的作用，正是软件和软件开发过程的实质所在。

3. 软件开发工具的概念

这一要点可从软件开发工具的范围去理解。

软件开发工具的范围大致可以描述为：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。

软件开发工具的要点是：

- (1) 它是在高级程序设计语言（第三代语言）之后，软件技术进一步发展的产物；
- (2) 它的目的是在人们开发软件中给予各种不同方面、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他稳定的工作阶段。

4. 软件开发工具的基本功能

- (1) 认识与描述客观系统（描述系统）
- (2) 存储及管理开发过程中的信息（管理信息）
- (3) 编写或生成代码（生成代码）
- (4) 编制或生成文档（生成文档）
- (5) 管理软件项目（项目管理）

5. 软件开发工具的主要性能指标

(1) 表达能力或描述能力（表达能力）；(2) 保持信息一致性的能力（逻辑能力）；(3) 使用的方便程度（人机界面水平）；(4) 工具的可靠程度（防错能力）；(5) 对硬件和软件环境的要求。

【本章难点】

因本章是对全课程内容的综合与概括，故广度大于深度，难点较少。相对而言，对软件开发工具概念的理解稍难一些，学员应从不同的角度去理解。

同 步 练 习

一、判断题（正确的在题后括号内划“√”，错误的划“×”）

1. 设计工具常常依赖于特定的软件。 ()
2. 集成化的软件开发工具常常依赖于特定的软件。 ()
3. 计划工具是为主管人员服务的。 ()
4. 规模大的程序就可称为软件。 ()
5. 集成化的软件开发工具常常称为软件工作环境。 ()
6. 白盒测试能发现程序中的所有错误。 ()
7. 成功的测试可以保证一个软件没有错误。 ()
8. 一旦使用了软件开发工具，程序员就不再需要编写程序了。 ()
9. PCTOOLS 是一个常用的软件开发工具。 ()

10. 随着软件开发方法的发展, 软件开发工作离用户越来越远, 离机器越来越近。 ()

二、单项选择题 (在每小题的四个备选答案中选出一个正确的答案, 并将其号码填在题干的括号内)

1. 从 () 语言开始, 软件摆脱了对硬件的依赖。
A. 第一代 B. 第二代 C. 第三代 D. 第四代
2. 下列概念中, 其外延最大的是 ()。
A. CASE 工具 B. 4GL C. 软件开发工具 D. 过程化语言
3. 软件开发工具一般是以 () 语言为基础的。
A. 第一代 B. 第二代 C. 第三代 D. 第四代
4. 出现得最早、数量最多的软件开发工具类型是 ()。
A. 计划工具 B. 分析工具 C. 设计工具 D. 软件开发工具
5. 软件开发项目管理人员所需要的主要软件开发工具是 ()。
A. 计划工具 B. 分析工具 C. 设计工具 D. 编程工具
6. 软件开发工具的认识与描述客观系统的功能主要是对 () 阶段的支持。
A. 需求分析 B. 总体设计 C. 程序编制 D. 运行维护
7. 为了普及和推广计算机的应用, 大量工作应集中在 ()。
A. 硬件领域之中 B. 软件领域之中
C. 语言和界面完善方面 D. 程序设计方法学
8. 经过软件开发人员数十年的努力, ()
A. “软件危机”已经克服
B. 证明“软件危机”不可克服
C. 证明软件工具是克服“软件危机”的途径之一
D. “软件危机”仍无好转
9. () 从更宏观的角度去看待软件的开发。
A. 计划工具 B. 分析工具 C. 设计工具 D. 软件开发工具
10. 设计工具是指在 () 阶段对人们提供帮助的工具。
A. 需求分析 B. 总体设计 C. 实现 D. 测试
11. 软件生产的成败更多地依赖于 ()。
A. 领导者的指挥才能 B. 程序员个人的编程能力
C. 合理地组织与协调 D. 用户的配合
12. 软件开发工具有对信息进行管理的功能, 管理的主要内容是保持它的 ()。
A. 一致性 B. 共享性 C. 快捷性 D. 节省空间

三、多项选择题 (在每小题的多个备选答案中, 选出若干正确的答案, 并将其号码填在题干的括号内)

1. 按支持的工作阶段划分, 软件开发工具划分为 ()。
A. 分析工具 B. 设计工具 C. 编程工具 D. 计划工具 E. 测试工具
2. 下列语言中接近自然语言的有 ()
A. 第一代 B. 第二代 C. 第三代 D. 第四代 E. 第五代
3. 下列语言中依赖于机器的有 ()
A. 第一代 B. 第二代 C. 第三代 D. 第四代 E. 第五代

4. 总体设计文件包括 ()。
 - A. 结构图
 - B. 框图
 - C. 模块清单
 - D. 公用数据结构
 - E. 模块设计任务书
5. 模型建立是哪些阶段的任务? ()
 - A. 需求分析
 - B. 分析设计
 - C. 编码
 - D. 测试
 - E. 维护
6. 软件开发工具应提供的各类支持为 ()
 - A. 认识与描述客观系统
 - B. 存储及管理开发过程中的信息
 - C. 编写或生成代码
 - D. 编制或生成文档
 - E. 计算机系统资源的管理
7. 我们可以从更高的层次来概括软件的功能和特点, 软件是 ()
 - A. 人类知识的一切来源
 - B. 广大使用者与计算机之间的桥梁
 - C. 人类进一步相互关联的纽带
 - D. 人类文明与知识得以延续的新的载体
 - E. 人类在各个领域中积累的知识的结晶
8. 计划工具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 () 等信息有条不紊地管理起来。
 - A. 进度
 - B. 资源
 - C. 质量
 - D. 验收情况
 - E. 成本
9. 集成化软件开发工具的代名词有 ()
 - A. 软件开发环境
 - B. 一体化软件工具
 - C. CASE 工具
 - D. 第四代语言
 - E. 软件工作环境
10. 软件开发工具的性能包括 ()
 - A. 效率、响应速度
 - B. 表达能力或描述能力
 - C. 保持信息一致性的能力
 - D. 使用的方便程度
 - E. 可靠性
11. 为了缓和软件危机, 人们提出了很多卓有成效的方法, 如 ()
 - A. 过程化程序设计方法
 - B. 结构化程序设计方法
 - C. 软件工程方法
 - D. 面向对象方法
 - E. 即插即用的程序设计方法
12. 与软件开发工具概念相关的术语有 ()。
 - A. 第三代语言
 - B. 第四代语言
 - C. CASE 工具
 - D. 可视化程序设计
 - E. 最终用户计算
13. AD/Cycle 失败的原因何在? ()
 - A. 对 C/S 体系结构接受太苛刻
 - B. 坚持在自己的特殊硬件平台上开发
 - C. 程序员水平不高
 - D. 软件危机的必然结局
 - E. 项目组内部人员协调不好
14. 用通用软件帮助人们进行软件开发的主要弱点是 ()。
 - A. 许多工作通用软件无法完成
 - B. 通用软件所完成的工作只能表现出其表面形式
 - C. 难于保持一致性
 - D. 通用软件功能不够强
 - E. 通用软件不支持软件开发所用的概念模式

四、填空题

1. 软件产品的_____已成为开发者和使用者共同关注的十分重要的问题。
2. 软件开发工具的范围大致可以描述为：在高级程序设计语言的基础上，为提高软件开发的质量和效率，从_____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。
3. 第一代语言又称为_____，第二代语言又称为_____，第三代语言又称为_____。
4. 高级设计语言实现了对_____的独立性，即它不依赖特定的硬件系统。
5. 高级语言一般都是_____语言，即需要由编写程序的人一步一步地安排好机器的执行次序；第四代语言则是_____语言。
6. 第四代语言的原意是_____的程序设计语言。
7. 软件和硬件的关系是：_____，二者缺一不可。
8. 软件应当包括_____两个不可缺少的组成部分。
9. 知识的_____作用，正是软件和软件开发过程的实质所在。
10. 明确地把_____纳入软件开发工作的范围，是软件技术向前迈进的重要一步。
11. 软件开发的过程可分为_____。
12. 需求阶段的文档是_____。
13. 总体设计的文档是_____及_____。
14. 总体设计文件应当包括_____。
15. 实现阶段包括两项并行的工作：_____。
16. 随着软件开发工具的发展，软件开发离_____越来越近，离_____越来越远。
17. 测试或调试阶段包括_____。
18. 软件开发工具的基本功能是_____。
19. 项目管理包括_____三个基本内容。
20. 软件开发工具的性能包括_____。
21. 按工作阶段划分可以把软件开发工具分为三类：_____。
22. 设计工具是指在_____阶段对人们提供帮助的工具。
23. 分析工具主要是指用于支持_____的工具。
24. 计划工具从更宏观的角度去看待软件开发，它不仅从_____的角度，帮助人们组织与实施项目，而且把有关_____等信息有条不紊地管理起来。
25. 软件开发工具按集成程度划分为_____的工具以及_____的工具。
26. 按与硬件、软件的关系划分，软件开发工具划分为_____的工具和_____的工具。
27. 软件开发工具有对信息进行管理的功能，管理的主要内容是保持它的_____。

五、名词解释

- | | |
|-----------|----------|
| 1. 软件开发工具 | 2. 机器语言 |
| 3. 汇编语言 | 4. 高级语言 |
| 5. 第一代语言 | 6. 第二代语言 |
| 7. 第三代语言 | 8. 第四代语言 |
| 9. 非过程化语言 | 10. 软件 |

- | | |
|-------------|-----------|
| 11. 程序 | 12. CASE |
| 13. 软件工作环境 | 14. 计划工具 |
| 15. 分析工具 | 16. 设计工具 |
| 17. 单项工具 | 18. 集成工具 |
| 19. 上游工具 | 20. 下游工具 |
| 21. 软件生命周期 | 22. 可视化编程 |
| 23. 软件功能说明书 | 24. 结构设计 |
| 25. 模块测试 | 26. 总体测试 |
| 27. 需求分析 | 28. 所见即所得 |
| 29. 软件危机 | |

六、简答题

1. 什么是软件开发工具？
2. 简述软件发展的几个阶段。
3. 简述 OS 对软件发展所起的作用。
4. 为什么说高级语言出现后计算机才真正获得了应用？
5. 何谓第四代语言的非过程化思想？
6. 软件工作的重要性何在？
7. 利用通用软件作为辅助工具时有哪些弱点？
8. 集成化的开发工具是在什么背景下产生的？
9. 什么是程序？
10. 什么是软件？
11. 软件与硬件的关系如何？
12. 软件的实质是什么？
13. 简述软件发展的新阶段的特点。
14. 什么是 4GL？
15. 软件开发分为那几个阶段，各阶段的文档是什么？
16. 简述软件开发工具的功能要求。
17. 简述软件开发工具的性能要求。
18. 简述软件开发工具认识与描述客观系统的功能。
19. 简述软件开发工具的项目管理功能。
20. 软件开发工具按软件开发阶段可以划分为哪几种？
21. 软件开发工具按集成程度可以划分为哪几种？
22. 软件开发工具按与软硬件的关系可以划分为哪几种？
23. 学习软件开发工具目的何在？

七、论述题

1. 试论述软件开发工具应具备的功能。
2. 试论述软件开发工具的主要性能指标。
3. 试论述软件开发方法发展的新阶段的特点。