



全国高等教育自学考试

自考密训资料

考前
30天

目录

第一章 绪论.....	1
第二章 软件开发过程及其组织.....	2
第三章 软件开发工具的理论基础.....	3
第四章 软件开发工具的技术要素.....	4
第五章 软件开发工具的使用与开发.....	5
第六章 软件开发工具的现状与发展.....	6
第七章 Eclipse 入门.....	7
第八章 Eclipse 工作台.....	7
第九章 使用 Eclipse 进行 C/C++ 开发.....	9
第十章 调试程序.....	10
第十一章 Eclipse CDT 开发常用功能.....	11
第十二章 CVS 的安装及使用.....	12
第十三章 Eclipse 插件的使用与开发.....	13
第十四章 常用建模工具.....	14

第一章 绪论

知识点名称	知识点内容	
软件开发工具的由来 ★★★★	1. 进入 21 世纪以来,软件开发工具的发展也进入了新的阶段。两个鲜明的特点:(1)面向网络;(2)开源软件的兴起和运用。 2. 为了克服软件危机,人们先后提出了多种程序设计方法:从结构化程序设计、软件工程、面向对象的程序设计方法、即插即用的程序设计方法,直到面向开源软件和互联网的程序设计。 3. 通用软件:好处:帮助软件开发人员编写文档或画图可以减少很多工作量。 4. 语言的发展: (1)第一代计算机语言:机器语言。 (2)第二代语言:汇编语言。用英语单词代替“0”和“1”组成的字符串。汇编语言未能突破与机器指令一一对应的限制。操作系统出现。 (3)20 世纪 60 年代初期,第三代语言:高级程序设计语言(过程化语言)。进步:高级程序设计语言实现了对机器的独立性。20 世纪 60 年代末期开始,出现了软件工作的困难性,即“软件危机”问题。软件开发工具是开发软件用的软件,它是在第三代语言的基础上发展起来的。 (4)第四代语言:非过程化语言。	
软件开发工具的概念 ★★	5. 软件应当包括程序和文档两个不可缺少的组成部分。 6. 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。 7. 对于 CASE 工具有两种理解:计算机辅助软件工程和计算机辅助系统工程。 8. 第四代语言可以被看做软件开发工具中的一类,而不是全部。	
软件开发的过程 ★★	9. 过程:(1)起点是初始要求的提出;(2)第二个阶段为总体设计(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包括结构图、模块清单、公用数据结构(文件或数据库的格式));(3)第三阶段为实现阶段;(4)第四阶段是测试或调试阶段(包括模块的调试和整个软件的联调)。	
软件开发工具的分类 ★★★★	10. 按工作阶段划分	(1)设计工具(用于实现阶段,最具体,出现最早,数量最多)。 (2)分析工具(支持需求分析;分析工具主要指用于支持需求分析的工具,如 Dictionary / 3000。帮助人们绘制数据流程图的专用工具——FLOW)。 (3)计划工具(保存整个项目的宏观信息,为项目主管人员服务)。
	11. 按集成程度划分	集成化的软件开发工具常被称为软件工作环境。
	12. 按与硬、软件关系划分	上游工具相当于分析工具,分析工具与计划工具往往是独立于机器与软件的,而集成化的软件开发工具又常常是依赖于机器与软件的。
软件开发工具的发展的特点★★	13. 进入 21 世纪以来,软件开发工具的发展也进入了新的阶段,两个鲜明的特点: (1)面向网络; (2)开源软件的兴起和运用。	
软件开发工具的功能要求★★	14. 项目管理包括:进度管理、资源与费用管理、质量管理。 15. 软件开发工具的功能要求:(1)认识与描述客观系统;(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)软件项目管理。	
软件开发工	16. 对于软件开发工具来说,以下五项性能是特别重要的:	

具最重要的性能 ★★	(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) 对硬件和软件环境的要求。
----------------------	---

第二章 软件开发过程及其组织

知识点名称	知识点内容
软件开发方法的发展 ★★★★	17. 软件开发的过程中, 软件工作者提出的方法 : (1) 结构化程序设计 (三种基本模块 : 处理单元, 循环机制, 二分决策机制。 要点 : 子程序尽可能做到只有一个入口和一个出口。 模块划分基本要求 : 模块功能单一化、模块间联系尽可能少、模块规模足够小) (2) 软件工程方法 (思想主要集中于加强项目管理者的工作上) (3) 面向对象的程序设计方法 (产生巨大影响根本原因在于它提供了认识框架) (4) 即插即用的程序设计方法 (基础: 面向对象程序设计) (5) 面向开源软件的设计方法 (依据: 程序员对于应用领域的充分了解)。 18. 在大型软件开发中的 各种角色及其工作 : (1) 用户 : 提出需求、验收软件、使用软件、提出修改要求。 (2) 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。 (3) 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。 (4) 硬件 : 运行、试运行。
结构化程序设计方法 ★★★	19. 结构化程序设计的思想产生于 20 世纪 60 年代末 。 20. 程序的结构可以分解成三种基本模块: 处理单元, 循环机制, 二分决策机制 。 21. 结构化程序设计的方法主要是为 程序员 服务的。 22. 模块之间的联系及互相影响称为 耦合 。一般来说, 应当尽量避免逻辑耦合, 而仅限于数据耦合。 23. 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。
软件工程方法 ★★★	24. 软件工程的思想主要集中于加强 项目管理者 的工作上。 25. 软件工具的思想与方法得到了广泛的宣传是在 20 世纪的 80 年代 。 26. 五个阶段 : 1) 需求分析。2) 分析与设计。3) 编程阶段。4) 测试阶段。5) 使用及维护。
面向对象的程序设计方法 ★★★★	27. 面向对象的程序设计语言 : 是在于合理地选择层次, 使得系统的复杂性保持在可控制的范围内。Smalltalk 属于面向对象程序设计语言。 28. 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, 也都是一些相关的操作; (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) 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, 这就是“遗传性”。(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。

	29. 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 。出现了 面向对象的系统分析(OOA) , 面向对象的系统设计(OOD) , 面向对象的数据库管理系统(ODBMS) 等。
即插即用的程序设计方法★★	30. 基本思想 ：一部分人专门生产 软件组织 ，而另一部分人则构造整个软件的 结构 。 31. 面向对象程序设计 是提出即插即用程序设计的 基础 。
软件开发过程的管理★★	32. 好的软件应具备的特点 ： (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； (3) 具有足够的速度； (4) 具有足够的可靠性； (5) 程序易读，结构良好，文档齐全，易于修改。 33. 怎样建立一个好的项目组 ？ (1) 有严格、成文的工作规范和文档标准； (2) 人员有严格的分工； (3) 严格执行事先制定的时间表； (4) 每个项目完成后有完整的资料。

第三章 软件开发工具的理论基础

知识点名称	知识点内容
软件开发过程的信息需求★★★★★	34. 软件开发过程中各类人员与计算机之间流通信息： (1) 有关系统环境、现状及 需求的信息 。由用户提出，由分析人员采集，经过他的理解，成为需求分析及设计的依据。； (2) 有关软件的 功能设计和物理设计 的各种信息。由分析人员根据需求分析而形成，体现为设计方案； (3) 软件成果 ，包括程序和文档。由程序员根据设计方案，依据某种计算机语言编制出来的，文档常常以联机方式提供； (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。用户在使用过程中逐步形成，由维护人员收集、整理，并实施修改。(跨开发周期的)

	35. 软件开发工具合理存储、正确转化的四类信息是 需求信息、设计方案、变更要求和代码文档 。
常用的概念模式 ★★★★	36. 框图 （是人们在编写软件时最早使用的一种概念模式。它是用来描述程序执行的逻辑过程。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。） 37. 结构图 （当程序模式比较大时，直接用框图表示会过于复杂，使人无法分层次地掌握程序的结构。针对这种情况，人们引入结构图，其中在结构图中，用菱形框连接模块表示的是选择调用。程序的调用方式有三种：顺序调用、选择调用、循环调用） 38. 数据流程图 （数据流程图面对的是一个系统的信息流程。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。） 39. 实体关系图 （是一种用于描述静态数据结构的概念模式。它以实体、关系、属性三个基本概念概括数据的基本结构。它广泛应用于数据库的设计中，常常和数据流程图、结构图等相互配合使用。） 40. 数据字典图 （是一种描述数据内容的概念模式。它用表格的形式列出数据的基本属性以及相互关系，作为人们对于数据的认识 and 了解，它的雏形是编写软件时的变量说明或标识符清单） 41. 时序网络 （时序网络是软件开发中常用的一种概念模式。主要描述系统的状态及其转换方式，状态是指系统在运行中某特定的形态或工作方式，转换是指状态在一定条件下的相互转换。经常应用于一些实时控制方面的软件功能描述。） 42. 数学与逻辑模型 （常用表达方式：决策树和决策表）
信息库及其一致性 ★★	43. 如何保持一致性 。这对信息库来说是最困难的，也是核心。 44. 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 。
人机界面及其管理 ★★	45. H·西蒙：“对于用户，界面就是系统本身”。 46. 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、数据录入和信息显示。
项目管理与版本管理★	47. 从项目的观点来说，最终都体现为 成本的升高或降低 。 48. 用户的反馈信息 是项目管理与版本管理的重要资源。

第四章 软件开发工具的技术要素

知识点名称	知识点内容
基本功能 ★★★★	49. 软件开发工具的基本功能： （1）提供描述软件状况及其开发过程的概念模式； （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。软件开发工具通过各种信息的提供，使用户能够在较短的时间内半自动地生成所需的代码段落，进行测试、修改错误等。 （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。
一般结构 ★★	50. 软件开发工具的一般结构： （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——两个重要输出及 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 。
总控与人机界面 ★★★	51. 三个技术要点 : 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)。 52. 各阶段任务 : (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) 分析与设计阶段 , 其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, 是实际的程序代码的产生点。(4) 测试阶段 。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。测试阶段的工作内容是安排测试方案, 准备测试数据, 收集与分析测试结果, 并对出现问题的模块做修改和调整。(5) 维护阶段 。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。 53. 面向用户及其原则 : 所谓面向用户, 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 。原则: (1) 总控对各部分的调度与安排应符合概念模式; 向用户提供的统一界面应体现概念模式。(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。
信息库及其管理 ★★	54. 信息库是软件开发工具的 基础 。 55. 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。
代码生成 ★	56. 基本任务是根据设计要求, 自动或半自动地产生相应的某种语言的程序。 57. 软件开发工具容易生成: 对话屏幕 、 输入屏幕 、 输出报表 、 菜单控制 。 58. 人为设计程序: 统计分析、数据加工、决策判断。
文档生成 ★★	59. 与代码生成相比, 数量更大、内容更复杂。包括 文章 、 表格 、 图形 三大类。其中, 最容易生成的是表格, 其次是图形, 最难处理的是文章。
项目管理与版本管理★	60. 针对项目管理在信息处理上困难的情况, 人们依据项目管理的经验, 提出了以 项目数据库 为中心解决这些问题的思路。

第五章 软件开发工具的使用与开发

知识点名称	知识点内容
软件开发工具的使用与开发★	61. 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。
购置与开发的权衡 ★★★	62. 自行开发软件开发工具的优缺点 : 优点: 目标明确, 切实符合自己的需要, 便于进一步扩充和升级, 不存在引进外面产品时不可避免的冲突与不一致; 可以商品化, 成为软件产品出售。缺点: 往往低估开发难度。 63. 购置或自行开发的权衡因素 :

	<p>(1) 准备从事的软件开发工作的性质与要求。这是决定购置还是自行开发的最基本的因素。(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(所谓工作环境包括硬件配置、系统软件、数据库管理系统、网络通信等各种条件。一般的软件开发工具都是在一定的工作环境中工作的,环境不一样就不能正常运行、发挥作用)。(4) 人员因素。</p> <p>64. 软件开发工具开发难度: 自制的; 单向的、专用的; 集成与一体化的工具开发难度依次递增。</p>
调查市场 ★★	<p>65. 调查研究重点: (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必不可少(主要考察两个方面: ①考察该款软件开发工具对应软件结构的理解与认识; ②考察其对开发过程的划分与各阶段任务的理解与认识)。(4) 软件开发工具的运行环境是什么 (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、培训条件如何 (7) 价格。</p>
软件开发工具的开发★	<p>66. 自行开发工具原则: 开拓新功能。</p> <p>67. 不成功的商品化软件开发工具往往由于: 使用手册复杂。</p>
软件开发工具的使用过程★★	<p>68. 人们转变工作方式的过程, 故组织者应进行人员培训工作, 使所有有关工作人员都真正领会软件开发工具所包含的思想与方法。</p>

第六章 软件开发工具的现状与发展

知识点名称	知识点内容
软件开发工具的发展现状★	<p>69. 国外发展状况: 20 世纪 80 年代中期, 专项的、支持某一工作环节的专用工具大量涌现, 人们很快发现了这种分散应用的弱点, 提出了一体化的要求。IBM 于 1989 年提出 AD/Cycle 界于应用系统开发和 CASE 工具的总框架。</p> <p>70. 国内发展状况: 从研究的水平来看, 国内这方面的水平并不比国外低多少。国外同行考虑的各种问题, 我国研究人员也都进行了相应的工作。主要的落后点在于应用, 即没有广泛使用这些工具。</p>
几个值得注意的发展方向★★	<p>71. 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。</p> <p>72. 只有对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、一致性等有了完整深刻的理解, 才能真正实现软件开发工具的一体化。</p>
软件工具的发展轨迹★★★	<p>73. 软件开发工具的发展经历了结构化时代、面向对象时代、互联网时代。Turbo Pascal 出现于结构化时代。</p> <p>74. 2008 年, 国际电子电气工程师学会的权威刊物《软件》的九月/十月号, 以“软件开发工具”为题的一期专刊中概括了 40 年来软件开发工具的发展轨迹, 指出抽象程度最高的软件开发工具是 XMF Mosaic。</p> <p>75. 观念: 实践性很强; 抽象程度越来越高; 历史发展是多样性和趋同性并存的。</p> <p>76. 从几十年软件开发工具发展历史中, 可以看到软件开发工具一个值得注意的特点是多样性和趋同性的并存。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质量和效率。</p> <p>77. 作为一款著名软件工具,Java 虚拟机(JVM)出现于面向对象时代和互联网时代的交集。</p>

从计算机语言发展而来的工具 ★★★★★	比较项目	VB	VC	PB	C++Builder / Delphi	JAVA
	跨平台特性	无	无	Windows,Solaris,Macintosh	Windows,Linux	所有能够运行 JAVA 虚拟机的操作系统
	组件技术	COM, ActiveX	COM, ActiveX, CORBA	COM, JavaBean, Jaguar, UserObject	COM, ActiveX, CORBA	JavaBean,CORBA ;ActiveX
	数据库支持级别	DAO, ADO, RDO	DAO, ADO, RDO, 但有不少类库可用, 但开发效率低	Transaction, Datasource, Control, 可绑定任何 SQL 语句	DataSource, Table, Query, Midas, ADO	JAVA JDBC API
	开发模式	控件开发模式 (OCX)	应用程序框架开发模式 (MFC)	组件开发模式 (User Object)	源代码组件开发模式 (VCL)	框架开发模式 (JDK 下)+组件开发模式 (IDE 下)
	面向对象	差	好	较好	很好	非常好
	开发效率	较高	很高	很高	高	高
	代码执行效率	一般	很高	较高	很高	低
	发展潜力	差	一般	一般	一般	很好

第七章 Eclipse 入门

知识点名称	知识点内容
Eclipse 简介 ★★★	<p>78. Eclipse 是一个开放源代码、基于 Java 的可扩展集成应用程序开发环境。就其本身而言，它只是一个框架和一组服务，通过插件组件构建开发环境。</p> <p>79. 用途：最初主要用来进行 Java 语言开发，但 Eclipse 并非只有这个用途。</p> <p>80. Eclipse 的体系结构：运行时内核（平台运行库是内核）、工作空间（是负责管理用户资源的插件，包括用户创建的项目、项目中的文件，以及文件变更和其他资源。）、工作台（为 Eclipse 提供用户界面）、其他插件（包括帮助组件、团队支持组件等）等。</p>

第八章 Eclipse 工作台

知识点名称	知识点内容
初识 Eclipse 工作台 ★★	<p>81. Eclipse 工作台：是一个高级用户界面，它为用户提供了一个整体架构和可扩展的用户界面。</p> <p>82. 工作空间：Eclipse 在用户计算机磁盘上划出的一块区域，用来存放用户工作资料，如代码、配置信息等。工作空间以项目为单位组织文件和目录。</p> <p>83. 使用：先找到安装目录（如 F:\Eclipse）下的可执行文件 eclipse.exe，然后用鼠标双击即可打开 Eclipse IDE。</p>

透视图及视图介绍 ★★★★	84. Eclipse 的工作台的组成部分： (1) 菜单栏 （位于整个窗口的顶部，与其他软件一样，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) 工具栏 ：分为主工具栏、视图工具栏。 (3) 透视图 ：分为 视图 、 编辑器 。				
视图 ★★★★★	85. 视图 包括导航器视图（ 又称资源管理器 ）、大纲视图、控制台视图等。 86. 视图 是工作台中一个可视化的组件，提供了用户正在工作台中使用的一些 对象 的详细信息。 87. <u>主要视图介绍</u> ： 88. 导航器视图 ：资源管理器视图，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。 大纲视图 ：显示当前活动编辑器中所打开文件的纲要，如函数、变量等。 控制台视图 ：显示程序的输出内容。 错误视图 ：编写代码时显示程序中的错误，提示用户及时改正。也显示相关警告信息（并不影响程序正确运行），建议用户进行相应的改正。 89. 搜索视图 ：显示用户搜索结果的详细信息。 90. 任务视图 ：显示程序代码中未完成的任务，在代码中加一条以 TODO 开头的注释标记，就可以在任务视图中添加一项任务。				
编辑器介绍 ★	91. 编辑器是工作台中一个 可视化组件 ，编辑器允许用户打开、编辑、查看和保存文档对象。在 Eclipse 中，所有视图共享同一组编辑器。 92. <u>常用的编辑器</u> ：C/C++编辑器、文本编辑器、任务编辑器、二进制文件编辑器、 Web 浏览器 等。它们分别用来打开 C/C++、文本文件、任务文件、二进制文件、网页文件等特定类型的文件。				
管理项目 ★	93. 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。管理项目是程序编写时最常见的工作之一。 94. 从外部导入项目 ：（1）先点击菜单栏中的文件（File）菜单，然后选择导入(Import)选项；（2）从外部导入归档文件（包括通过 Jar 命令或 War 命令打包后形成的压缩文件）；（3）导入文件系统（指操作系统文件夹中的各类文件）；（4）导入首选项文件（可以让用户通过该配置文件来个性化定制 Eclipse）。				
常用快捷键介绍 ★★★★★	作用域	功能	快捷键	功能	快捷键
	全局	查找或替换	Ctrl+ F	后退一步	Alt+←
		撤销	Ctrl+Z	前进一步	At+→
		复制	Ctrl+ C	打开搜索对话框	Ctrl+ H
		剪切	Ctrl+X	保存文件	Ctrl +S
		全部选中	Ctrl+A	打印	Ctrl +P
		粘贴	Ctrl+V	全部保存	Ctrl+Shift+S
		恢复撤销	Ctrl+ Y	编译	Ctrl+ B
		激活编辑器	F12	打开任务	Ctrl + F12
		切换视图	Ctrl + F7	调试	F11

		切换编辑器	Ctrl +F6	运行程序	Ctrl+F11
		切换透视图	Ctrl F&	删除	Delete
	文本编辑器	上滚一行	Ctrl+ ↑	格式化代码	Ctrl Shift +F
		下滚一行	Ctrl+ ↓	重命名	Alt + Shift +R
		查找上一个	Ctrl+Shift+K	以元素为单位移动光标	Ctrl+方向键
		查找下一个	Ctrl+K		

第九章 使用 Eclipse 进行 C/C++开发

知识点名称	知识点内容
使用 Eclipse 进行 C/C++ 开发★★★	95. C 语言是一种 面向过程 的计算机程序设计语言，既有 高级语言 的特点，又具有 汇编语言 的特点。
安装 MinGW ★	96. 为了能够使用 Eclipse CDT 编译且运行 C 和 C++程序，必须要安装一个 C/C++编译器 。常用的有 MinGW 编译器。MinGW 是指用来生成纯粹的 Win32 可执行文件的编译环境，它是以 GNU 为基础的开发 C/C++项目的工具集，能够提供 C/C++所需要的头文件和库文件。
新建 C/C++项目 ★★★★★	97. 单击菜单栏中的 文件（File） 菜单，选择 新建（New） 子菜单下的 项目（Project） 选项，弹出选择向导窗口。在对话框中打开 C / C++文件夹，选择“C Project”后单击“Next”进入下一步。 98. 在 C Project 窗口中选择 C 语言相关 属性 。在项目名称标签中输入 HelloWorld，在 项目类型 标签中打开 Executable ，选择 Hello World ANSI C Project，在工具箱标签中选择 MinGW GCC，完成之后单击“Next”按钮进入下一步。
新建 Source Folder ★★★★★	99. 鼠标右键点击 资源浏览器 中的新建的项目，在弹出的菜单中，选择“New”子菜单里的源文件夹“Source Folder”选项。在弹出的窗口中需要输入 文件夹名称 ，把光标定位到文件名一栏中输入源文件夹名称“src”，然后点击完成 Finish 即可。
新建 C++Class ★★★★★	100.在 src 文件中新建源代码文件。用鼠标右键单击 src 文件夹 ，在弹出的菜单中选择新建（New）子菜单里的 类（Class） 选项，在 类名称（Class Name） 标签中输入类名称后单击“Finish”按钮即可完成新建操作。
编译 C 程序 ★★★★★	101.编写好源代码后，接下来需要将.c 文件编译为 可以运行的.exe 文件 。若编译错误，Eclipse 编辑器中错误行代码显示“X”符号，提示用户此处出现语法错误。同时资源管理器中对应的文件图标同样显示“X”符号。
运行 C 程序 ★★★★★	102.编译成功后将在资源管理器中看到 Release 文件夹，其包含了刚被编译的可执行文件。接下来运行刚编译的文件，在资源管理器中用鼠标右键单击 HelloWorld 项目，弹出菜单，在运行方式（Run As）子菜单中选择 运行设置（Run Configurations） 。进入新建窗口后，在 Build Configuration 选项中选择 Release ，并单击“Search Project...”按钮进行设置。
使用 浏览功能	103.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。

★★★★	<p>104. 打开变量声明: 将光标定位到编辑器中相应的变量位置, 单击菜单栏中的浏览菜单, 选择打开变量声明选项, 也可使<F3>快捷键。</p> <p>105. 打开类型层次结构: 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。</p> <p>106. 将光标定位到编辑器中相应的变量位置, 单击菜单栏中的浏览 (Navigate) 菜单, 选择打开类型层次结构 (Open Type Hierarchy) 选项, 也可使用<F4>快捷键。</p> <p>107. 打开调用层次结构: 如将光标定位到编辑器中的 add()函数; 然后, 点击菜单栏上的浏览“Navigate”菜单, 选择打开调用层次结构“Open Call Hierarchy”选项, 即可看到 add()函数的调用层次, 即 main()->add()。快捷键: Ctrl+Alt+H。</p> <p>108. 打开文件包含浏览器: 单击菜单栏中的浏览 (Navigate) 菜单, 选择打开文件包含浏览器 (Open Include Browser) 选项, 也可使用 Ctrl+Alt+I 快捷键。</p> <p>109. 打开元素: 单击菜单栏上的浏览“Navigate”菜单, 选择打开元素 (Open Element), 快捷键: Ctrl+Shift+T。</p>
抽取常量 ★	<p>110. 在很多复杂的程序中, 也会存在对同一个数字或字符串的多次调用, 每当出现这种情况我们应当用常量替换这个数字或字符串, 以增加程序的可读性和易修改性。抽取常量本质上是将程序中的数字、字符等定义为常量, 使得对于常量的调用和修改变得简单。</p>
使用 Search 菜单进行搜索 ★	<p>111. Eclipse 作为一个高度集成化的平台, 除了最基本的编辑器功能之外, 还提供了大量实用的个性化功能, 在 Eclipse CDT 特有的 Search 功能中可以执行文件、任务和 C / C++搜索功能。</p> <p>112. (1) 打开 Search 对话框 (2) 执行 C/C++搜索 (3) 执行文件搜索 (4) 执行任务搜索</p>

第十章 调试程序

知识点名称	知识点内容
下载并安装 GDB ★★	<p>113. 为了能够使用 Eclipse CDT 调试 C/C++程序, 必须要安装一个 C/C++调试器—GDB 调试器。</p> <p>114. GDB 调试器安装步骤:</p> <p>(1) 打开浏览器, 在地址栏输入下载地址后, 选择上方的 Files 标签, 进入下一步。2) 在 Files 标签中的下载列表找到 bin 文件, gdb-7.0-mingw32-bin.tar.gz。</p> <p>(2) 下载打开压缩包, 进入 bin 文件夹下找到 gdb.exe。</p> <p>(3) 将 gdb.exe 解压到 MinGW 安装目录下的 bin 文件夹下即可。</p>
CDT 调试器 ★★	<p>115. Eclipse CDT 调试器允许用户使用设置断点、暂停、单步执行代码等方法来控制程序的运行过程。</p> <p>116. 自定义调试视图: 单击菜单栏中窗口 (Window) 菜单, 选择首选项 (Prefereces) 选项, 在弹出的窗口左侧选择运行/调试 (Run/Debug) 树形菜单即可设置调试视图的相关属性。</p>
设置行断点 ★★★	<p>117. 断点: 可以使程序在运行到断点位置时自动暂停并且显示程序的当前状态。Eclipse 支持行断点、方法断点、事件断点。</p> <p>118. 激活状态的断点以蓝色圆圈显示, 非激活状态的断点以白色透明圆圈显示。</p> <p>(1) 添加行断点。在资源管理器中, 打开需要调试的文件, 将光标移动到编辑器左侧边缘的标记区域上, 用鼠标右键单击, 从弹出菜单中选择 Toggle Breakpoint 即可设置一个行断点。在 GDB 的窗口 Show View 中, 打开调试 (Debug) 文件夹, 选择断点 (Breakpoint) 后单击“OK”按钮即可激活</p>

	<p>断点视图。</p> <p>(2) 删除行断点。将光标移动到编辑器左侧边缘的标记区域上，用鼠标右键单击需要删除的行断点，弹出菜单选项。在弹出的菜单中，选择 Toggle Breakpoint 即可删除该行断点。删除断点的方法：鼠标双击、鼠标右键弹出菜单和在断点视图中删除。</p> <p>(3) 激活和禁用行断点。</p>
<p>设置方法断点</p> <p>★</p>	<p>119. 添加/删除。删除方法断点方式：双击、鼠标右键弹出菜单、在断点视图中删除。在 GDB 添加或删除方法断点时，单击菜单栏中的窗口菜单，在显示视图子菜单下的选择其他 (Other) 选项，弹出 Show View 窗口。</p> <p>120. 激活/禁用。将光标移动到编辑器左侧边缘的标记区域上，用鼠标右键单击需要激活的方法断点，弹出断点菜单。在弹出的菜单中，选择激活断点 Enable Breakpoint 即可激活该方法断点。</p>
<p>设置事件断点★</p>	<p>121. 事件断点：在程序运行过程中发生特定事件（如异常触发、线程启动、线程退出）将程序暂停。</p>
<p>设置断点动作★</p>	<p>122. Eclipse 规定了四种可用动作：声音动作、日志动作、重新启动程序动作和外部工具动作。</p>
<p>调试程序</p> <p>★★★</p>	<p>123. 调试透视图：主要包括调试视图、变量视图、断点视图、编辑器、大纲视图、控制台视图。</p> <p>124. 调试 C 程序：单步遍历程序主要包括单步跳入、单步跳过和单步返回。</p>

第十一章 Eclipse CDT 开发常用功能

知识点名称	知识点内容
<p>设置首选项</p> <p>★★</p>	<p>125. 任务标记 (Task Tags)：自定义任务标签，默认的任务标签格式为 TODO Normal。</p> <p>126. Eclipse 的常规首选项设置窗口，可以对 Eclipse 进行一般性的设置。首选项包括外观、键、搜索、工作空间、浏览器。</p>
<p>设置编辑器布局</p> <p>★</p>	<p>127. 在编辑器中，打开一个文件（如 Test.c），双击 Test.c 标签区域即可全屏显示该编辑器区域，再次单击该标签区域会恢复到初始状态。除了可以最大/最小化显示编辑器之外，用户还可以改变编辑器在 Eclipse 中的位置。</p>
<p>定制工作台</p> <p>★★★</p>	<p>128. 定制工具栏。工具栏的主要作用是向用户提供最常用功能的快捷按钮，在 Eclipse 中，工具栏是可以定制的。默认情况下工具栏区域被竖直的虚线分成若干部分，此时，工具栏处于解锁状态，也就是说用户可以随意地更改工具栏的状态。Lock the Toolbars 意为“锁定工具栏”，将√去除为解锁操作。</p> <p>129. 定制快捷键。Eclipse 提供了一系列快捷键来方便用户的操作。尝试对 (About) 设定一个快捷键，首先选中 About，在描述框中会出现该功能的简单介绍——打开关于对话框 (Open the about dialog)。将光标定位到绑定 (Binding) 输入框，同时按下“Ctrl”和“=”键即可设定成功。单击菜单栏中的窗口菜单，选择首选项 (Preference) 选项，在弹出的窗口左侧部分树形菜单中，选择“General”→“Keys”，即可打开定制快捷键界面。</p> <p>130. 定制透视图。Eclipse 定制透视图编辑器在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。选项卡中只有三项：Lock the Toolbars 锁定工具栏；Customize Perspective 定制透视图；Hide Toolbar 隐藏工具栏。</p>

格式化代码 ★★★	131. 格式化代码 之前首先需要设定代码格式。单击菜单栏中的窗口，选择首选项选项，在弹出的对话框左侧部分，展开 C++ 树形菜单，选择代码格式。 132. 设定完成之后，用户即可 格式化代码 。 133. 代码模板 是 Eclipse 为用户提供的可自定义的结构化代码格式。使用代码模板用户可以省去许多重复的代码编写工作。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。Eclipse 允许用户在注释、代码和文件三个维度设置代码模板。注释和文件三个维度设置代码模板可以提高程序的质量和效率。
生成历史记录 ★	134. 利用 Eclipse 自带的“保存历史记录”功能进行版本比较时，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。 135. 单击 Replace 按钮即可把该文件恢复到历史记录的版本。需要注意的是，Eclipse 只会记录较短时间内的文件版本，如果需要进行长期的版本控制，则需要借助类似 CVS 的版本控制软件完成。 136. 在打开的窗口中，可以看到 Hello World 项目中曾经存在过的所有文件以及它们的各个版本。选中需要恢复的文件后单击 Restore 按钮即可完成。

第十二章 CVS 的安装及使用

知识点名称	知识点内容	
常用版本控制软件 ★★★★★	Rational ClearCase	重量级管理软件，功能多样。涵盖的范围包括版本控制、建立管理、工作空间管理和过程控制。
	CVS	简单易用、功能强大、开源、跨平台、支持并发版本控制，免费；缺少相应的技术支持。
	VisualSourceSafe	简称 VSS，入门级工具，易学易用，得到微软稳定的技术支持；只用于 Windows 系统；安全性不高。
	PVCS	MERANT 公司核心产品，主要支持 team 开发中的一系列管理问题。
	Perforce	美国 Perforce 软件公司产品，易用性强，速度快
	CCC	最早的配置管理工具之一，元老级的版本控制软件
	Star Team	用于管理配置和变更的集成环境，能够与众多工具平台进行无缝对接
	RCS	元老级版本控制软件，属于单一文件的版本维护系统，适用于任何正文文件的版本维护。
	SCCS	和 RCS 类似，也是早期的基于单一文件的版本维护系统。
	Hansky Firefly	可轻松管理、维护整个企业的软件资产，包括程序代码和相关文档。是新兴的优秀版本控制软件。
CVS Control Panel 的设置★★	137. CVS 是一款常见的简单、易用、功能强大且开源的版本控制软件。CVS 安装完成后，首先需要进入 CVSNT Control Panel 进行一些简单的配置。其中，About 页面提供了 CVS 产品的版本、来源以及服务状态等信息。CVS 资源库即软件资源的存放地，通俗地说就是放置开发的代码并对其进行管理的一个文件夹。	
代码同步 ★★★★	名称	说明
	签出	获得工作备份

	提交	将对工作备份的修改反映到资源库中
	标签	对某个时刻的快照赋予一个标识名称，这个名称称为标签
	快照	在某一时刻，模块中文件状态的静态影像
	更新	将资源库中的最新状态反映至工作备份
	输入	将处于资源库中的软件模块登录到资源库
	输出	从资源库中取出模块。使用 export 方式取出的模块复制不包含版本管理的相关信息，对该模块复制的修改也不能反映到资源库。
	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
	冲突	在资源库与工作备份之间状态不一致的状态下进行签入或更新操作时，版本管理系统可能会尽量进行合并，如果版本管理系统不能完全处理上述不一致，就称为产生了冲突
	分支	分支是一种特殊的标签。从分支中签出的资源是可以被修改的。引入分支是为了更好地支持项目的并行开发过程
	修订版	CVS 版本管理系统用修订版来管理文件的修改历史，修订版用版本号来表示，即修订版号。
	资源库	资源文件的集合，在 Eclipse 中被称为 CVS Repository
	模块	资源文件的组织形式，在版本管理系统中的表现形式为目录

第十三章 Eclipse 插件的使用与开发

知识点名称	知识点内容
插件简介 ★★	138. 定义：插件是一种遵循其所依附的软件的接口规范所编写出来的程序。 139. 构成：每个插件都是由一个 插件清单文件 （plugin.xml）和一些 可选文件 组成。 140. Eclipse 插件清单文件 描述了插件的名字、版本号以及使用的或本身定义的扩展点等信息。 141. 一个典型的插件文件夹一般包括： 142. plugin.xml。插件清单文件，主要有 plugin、runtime、requires 和 extension 四个标签。其中，plugin 标签的属性提供的是插件的基本信息，最重要的是 id，它要求不能和现有的 Eclipse 插件 id 有冲突。（2）plugin.properties。插件的一般信息，记录插件的属性设置，容纳被 plugin.xml 引用的字符串。（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。 143. 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。如果发现某个插件有多个版本，只有一个（ 通常是高版本号 ）将被使用。
PDE 基本操作 ★★	144. PDE 透视图主要包括 清单编辑器 和 插件大纲 两部分。 145. 清单编辑器 是一个多页编辑器，主要由六个页面组成：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）；② 依赖性页面 （该页面指定了当前插件运行时所需的插件列表）；③ 运行时页面 （显示插件提供给其他插件使用的所有包，以及插件运行时类路径中的库和文件夹。）；④ 扩展页面 （从总体上显示了当前插件可用的扩展点）；⑤ 扩展点页面 （指定扩展点的三个值：扩展点标识；扩展点名称；扩展点模式）；⑥ 构建页面 （包含构建、打包和导出插件时所需的

	所有信息)。
常用插件 扩展点 ★★	<p>146. Eclipse 插件用户界面的三个基本构成元素是视图、编辑器和透视图。</p> <p>(1) 视图扩展点: 在 Eclipse 中,同一时间只能显示一个编辑器,但是可以显示多个视图,这是视图和编辑器最显著的区别。</p> <p>(2) 编辑器扩展点: 通过扩展 org.eclipse.ui.s Editors 来将新编辑器添加至工作台。编辑器是工作台页面内的可视组件。工作台能够创建内部编辑器和外部编辑器。工作台与外部编辑器之间的集成更为松散。</p> <p>(3) 透视图扩展点: 透视图就是将已有的视图、操作集及编辑器进行组合和布局,从而支持特定的用户需求。透视图工厂用于定义透视图的初始布局和可视操作集。</p>

第十四章 常用建模工具

知识点名称	知识点内容
常用建模 工具★	147.UML 采用了面向对象的技术和方法。
UML 建模 介绍★★	148.UML (统一建模语言)是一种面向对象的建模语言。是一种功能强大、易于表达、定义良好且普遍适用的建模语言。
UML 语言与 建模工具 ★	<p>149.UML 的作用就是从静态和动态方面用模型图来从各个角度全面描述要开发的系统,为人们提供一套功能强大的捕获信息技术要点的工具。</p> <p>150.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。</p>
使用 Rational Rose 建模 ★★★	<p>151.使用 UML 建模时一般分为用例视图设计、逻辑设计和物理设计三大部分。</p> <p>152.用例视图设计:用例图、活动图、状态图。</p> <p>153.逻辑设计:类图(表现系统的静态信息)、顺序图和交互图(表现系统的动态信息)。</p> <p>154.物理设计:部署图等。</p>
UML 中的 视图★★	155.UML 中的视图是从不同角度来观察待建模的系统,其中, 用例视图 是从系统外部使用者对系统的需求来描述该系统工程所应具备和完成的功能。
面向对象 分析方法★	156.在面向对象分析方法中, 对象 是一些属性及操作行为的封装体,它是问题空间中一些东西的抽象。