尚德机构
SUNLANDS

密训·资料

软件开发工具（全国）

1904

MI XUN ZI LIAO

目录

编前语

试卷说明：

软件开发工具科目，为计算机管理专业（本科）。考试题型有单选、填空、简答、论述和案例分析

题型	分值
单选题	20 个×1 分=20 分
填空题	20 个×1 分=20 分
简答题	6 个×5 分=30 分
论述题	1 个×10 分=10 分
案例分析题	1 个×20 分=20 分

资料说明：

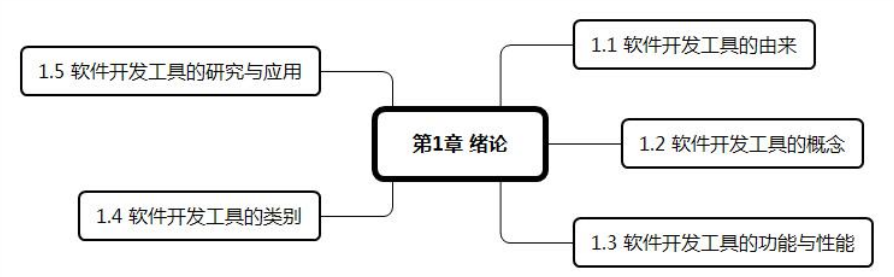
本资料提供的都是考试重点，但是在重点中，按照可能考试的题型、分值、频次进行了标星，对重点进行了等级划分。如下表：

星级	说明
1 星—2 星	单选题、填空题
3 星	简答题
4 星—5 星	论述题、案例分析题

星标数目越多，表示分值越高，知识点越重要。但要注意的是，资料提供的是知识重点，并不是押题，星级是根据往年考试频率及分值标的。资料的逻辑框图里进行了一级标星，这里的标星代表的是某章里相对重要的知识点。

资料对某些知识点的内容进行了精简或转化，主要是为了帮助同学们快速记忆，文中加粗的是需要重点了解的地方。这份资料是为了指导学员考前最后的复习，帮助学员理清知识点间的逻辑，加速记忆。学员在拿到资料后，可以结合逻辑图，根据星标数、资料里的逻辑整理和标识，重点复习。

第 1 章 绪论



1.1 软件开发工具的由来

概念★★	**软件开发工具**就是帮助人们开发软件的工具。
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，对软件开发者提供各种不同程度的帮助的一类新型的软件
产生与演变过程★★★★	（1）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”。机器的每一个动作都需要人们用二进制的字符串，即由“0”和“1”组成的字符串书写出来，十分不便。 （2）在这方面迈出的第一步是汇编语言，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、人们试图用在英语中具有一定意义的单词（或单词的缩写）来代替它，这就是所谓“**助记忆码**”，或**汇编码**。**操作系统差不多与汇编语言同时出现。** （3）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了难以应用的困窘局面**，即第三代语言的时代，第三代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了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。另一重要的进步：**高级程序设计语言实现了对机器的独立性**。第三代程序设计语言一般都是**过程化语言**。20 世纪 60 年代末期开始，人们认识到软件工作重要性的同时，也认识到软件工作的困难性，即“**软件危机**”问题。 （4）20 世纪 70 年代末到 20 世纪 80 年代初，用软件来进一步支持软件开发工作。通过软件来帮助软件开发人员编写文档或画图可以减少很多工作量，但是太表面，初级。**弱点**：有许多工作是通过软件所无法完成的（2）完成某些工作时，只能表现其形式，不能反映内涵（3）难以保持一致性。 （5）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出现，进入专用的软件开发工具的阶段。**软件开发工具种类**：面向特定功能模块的各种代码生成程序（包括报表生成器、菜单生成器、对话生成器等），综合性的第四代语言，专用于某种文档的编写工具，数据字典管理系统（DDMS），专用于画数据流程图、E-R 图或程序框图的绘图软件等。1989 年，IBM 公司宣布一个名为 AD/Cycle 的巨大的理论框架，作为它和它的软件合作伙伴开发一致的、统一的软件开发环境的纲领。是进入集成的软件开发环境阶段的标志。 （6）21 世纪，软件开发工具的特点：面向网络；开源软件的兴起和运用。

1.2 软件开发工具的概念

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。软件应当包括**程序**和**文档**两个不可缺少的组成部分★★

软件是人类知识与经验的结晶 ★★★	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。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：
软件技术的发展 ★★★	主要表现在四个方面：（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。（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。（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。
软件开发工作发展变化的五个阶段及其特点 ★★★★	（1）用机器语言写成一系列机器指令，供硬件执行； （2）用汇编语言开发软件，由汇编程序完成转换； （3）用高级语言开发软件，由编译程序完成转换； （4）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； （5）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。
相关概念 ★★	第四代语言的原义是**非过程化的程序设计语言。** 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**

1.3 软件开发工具的功能与性能

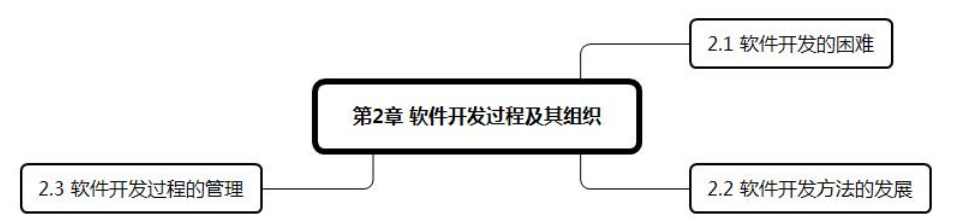
软件开发过程★★	（1）软件开发工作的起点是**初始要求的提出。**软件开发工作首要的任务是根据这种初始要求形成严格的、明确的、可供实际开发使用的功能说明书。（2）**总体设计。结构设计**是把软件划分成若干模块，指定每个模块的功能要求，以及它们之间的相互关系。**总设计的成果**是系统的总体设计文件及各个模块的设计任务书。总设计文件应包括**结构图、模块清单、公用数据结构。**（3）程序的编写与文档的编写是两件并行的工作，统称为**实现阶段。**（4）测试或调试阶段。包括模块的调试与整个软件的联调两个部分。
功能要求 ★★	（1）认识与描述客观系统。主要用于软件开发工作的第一个阶段——需求分析阶段。人们最希望软件开发工具提供的帮助是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）软件项目的管理。明确为项目管理人员提供支持。
性能★★★	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）对软件和硬件环境的要求

1.4 软件开发工具的类别★★

按工作阶段划分	设计工具（用于实现阶段，最具体，出现最早，数量最多），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），计划工具（保存整个项目的宏观信息，为项目主管人员服务）。

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。
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。

第 2 章 软件开发过程及其组织



2.1 软件开发的困难

软件开发的基本问题 ★★★	程序员做好软件工作的关键：第一个转换是用户对软件功能的理解与程序员对软件功能的理解之间的转换。一般来说，不同行业的人员对于事物的认识方法与描述方法是不同的。第二个转换要解决的是人和机器之间的交流与协调问题。 对于单个的程序员来说，以下两个转换是做好软件工作的关键。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。
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。（人们在实践中认识到，“黑箱”检验方法只能证明程序有错，而不能保证程序的正确性。）（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。
困难产生的原因 ★★★★	（1）这些困难来自大系统的复杂性；（2）许多具有主动性的个人之间的组织与协调，这本身也带来大量的困难；（3）各个应用领域之间的差别也导致这些困难的加重；（4）时间的因素，变化的因素也给软件开发工作带来许多困难。

2.2 软件开发方法的发展

开发过程中的角色	用户——提出需求、验收、使用、要求修改；项目负责人（软件架构师）——分析需求、向程序员分配任务、验收程序员工作成果等，在开发过程中起关键作用；
结构化程序设计方法 ★★★	结构化程序设计的思想产生于 20 世纪 60 年代末。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。**模块划分要求**：（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。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。结构化程序设计的方法主要的服务对象是程序员。**实施的基本思想**：1）限制（甚至不用）GOTO 语句，禁止超越模块边界的 GOTO 语句。2）子程序尽可能的做到只有一入口、一出口。3）程序风格应明确。4）完成有关的文档编撰。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**。
软件工程方法★★★	软件工程的思想主要集中于加强**项目管理者**的工作上。**软件工程方法**主要为项目管理者服务。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。软件工程思想的产生**：软件危机的发生；把软件的质量寄托于程序员的技能与认真负责是不牢靠的；从根本上，要想大幅度地提高软件开发的效率和质量，应吸取人们的成功经验，从组织上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可控制的工程，这就产生了软件工程。**软件工程从传统产业工程**

	方法中吸取的成功经验：对软件工程工作的步骤作出了严格的规定；工作顺序不能颠倒；每一个阶段都有各自的明确的任务；在质量、表达方式等方面要有统一的，并为人们共同遵守的标准；利用有关各方沟通与交流的手段，使参加工作的人们成为一个整体，共同地完成一项大的工程任务。
面向对象的程序设计方法★★★	客观世界的任何事物都是对象，它们都有一些静态属性和相关的操作。Smalltalk属于面向对象程序设计语言。基本思想：（1）客观世界的任何事物都是对象；（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，自然地成为二它的子类的性质，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。 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（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★★★	程序设计：一部分人专门生产**软件组件**，而另一部分人构造整个软件的**结构**，并且把软件组件插入这个结构，以便迅速地完成大型软件的研制工作。**基本思想**是用制造硬件的思路来生产软件，应用硬件制造思路来处理大型软件开发工作的方法。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，并且把软件组织插入结构中，以便迅速地完成大型软件的研制工作。**面向对象程序设计**是提出即插即用程序设计的基础。同时也存在标准化的问题，软件部分的提供方式的问题。**实现难度**：（1）标准化的问题就十分困难（2）软件部件的提供方式也是问题（3）与硬件和操作系统的关系问题.
面向开源软件的程序设计方法★★★	对于应用领域的充分了解是我们判断能否利用某开源软件的主要依据。 **适合利用开源软件进行程序设计情况**：在一些应用面广，流程比较清晰、比较规范的应用领域，开源代码的重用是比较有把握的。例如，网站的设计确实可以从开源代码中找到比较实用的，略加修改就可以使用的代码。这无疑是可以节省人力和物力的。

2.3 软件开发过程的管理

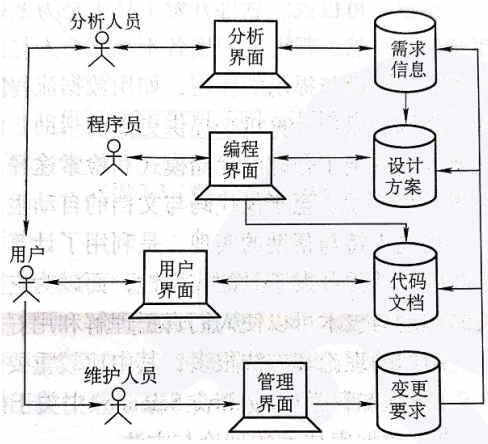
好的软件★★★	（1）正确地实现所要求的功能，准确地给出预定的输出结果（2）用户界面友好，符合实际用户的使用习惯与知识能力（3）具有足够的速度（而不是越快越好），能在符合用户要求的时间限度内，给出所要求的处理结果（4）具有足够的可靠性，能够在各种干扰下保持正常的工作（5）程序易读，结构良好，文档齐全，从而保证系统易于修改。正确地实现所要求的功能，用户界面友好，具有足够的速度，足够的可靠性，程序易读，结构良好，文档齐全是好的软件的概括。
好的程序员与好的项目组★★★	作为项目组的成员，需要能够接受项目组的限制和约束，服从项目组的严格管理。可以说，作为项目组的一员参加大型软件的开发，必须具有高度的**组织纪律性**和**团队精神**。**就单个程序员而言**：(1)具备程序设计所需的基本知识和技能。(2)对项目所在的领城有较深入的了解。(3)熟悉软件开发的技术环境。 **作为项目组的成员,还必须使自己的工作融入整个系统，严格遵守**：(1)仅在本模块内操作。(2)按总体设计的要求传递参数。(3)按统一规定的格式操作数据库或公用文件。(4)按统一的原则使用标识符。(5)按统一的要求编写文档。(6)保持程序风格

	一致。
建立好的项目组 ★★★	（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，并且切实得到遵守；（2）人员之间有严格的分工；（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。

第 3 章 软件开发工具的理论基础

3.1 软件开发过程的信息需求

软件开发过程中的信息流通状况：在软件开发过程中，直接与管理界面相关的人员是维护人员；直接与设计方案关联的界面是编程界面；直接与变更要求关联的界面是管理界面；直接与代码文档关联的是编程界面和用户界面；直接与需求信息关联的界面是分析界面；



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。

软件开发过程中各类人员与计算机之间流通信息的内容：★★★★

（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，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（3）软件成果本身，包括程序与文档。它是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，这类信息是跨开发周期的。

帮助理解和用好软件开发工具的较重要的理论和方法：★★★

（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。

3.2 概念模式及其作用

软件开发工具是引导用户建立正确的、有效的概念模式的一种手段。

常用的概念模式 ★★	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。）、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**）、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。）、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。）、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和了解，它的雏形是编写软件时的变量说明或标识符清单）、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。）、数学与逻辑模型（常用表达方式：决策树和决策表）、计算机模拟模型。
在软件开发工具中的作用★★★	软件开发工具是引导用户建立正确的、有效的概念模式的一种手段。概念模式包括对软件应用环境的认识和理解，对预期产出的软件产品的认识和理解，对软件开发过程的认识和理解，协助开发人员认识软件工作的环境与要求，组织与管理开发工作的过程。
数据流程图的组成和作用★★★	数据流程图由外部实体、数据处理与数据存储组成。 数据流程图面对的是一个系统的信息流程。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。其基本思想是把信息流看做一个组织或系统运作的线索，简明扼要地描述处理的过程。数据流程图不仅应用于描述已有系统的状况，也应用于描述设想中新系统的状况。
实体关系图的组成和作用★★	由实体、联系和属性三部分组成。 实体联系图是一种用于描述静态数据结构的概念模式。经常与数据流程图、结构图配合使用，广泛应用于数据库设计。

3.3 信息库及其一致性

有关信息库的研究主要集中在以下三方面：

（1）信息库的内容应当包括哪些方面。①所述软件的工作环境、功能需求、性能需求、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，包括用户本身提供的，也包括在调查研究中得到的。③逻辑设计阶段的各种调查材料和由此生成的各种文档，包括调查记录、原始数据、报表及单证的样本、绘制的各种图以及最后生成的系统说明书。④设计阶段的各种资料，包括所有的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以及最后形成的设计方案。⑤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（输入数据及输出结果）、验收报告、使用说明等。⑥运行及使用情况的详细记录，包括每次使用的时间、状态、问题，特别是有关错误及故障的记录情况。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、人员变更、资金投入、进度计划及实施情况。这项还包括版本信息，即各个版本的备份、每个版本的推出日期、与以前版本相比的变更说明等。★★★

（2）信息库应当具备哪些管理功能。（3）如何保持一致性。这对信息库来说是最困难的。★★信息库是一个包罗万象的，随着项目进度不断修改与补充的数据集合。它在规模上不一定像数据仓库那么大，分析提炼的要求也有数据仓库不同，这些就是信息库的特点。

3.4 人机界面及其管理

用户界面的基本原则：（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。★★★★

人机交互手段的八个方面是：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。★★

用 H•西蒙的话来说，“对于用户，界面就是系统本身”★

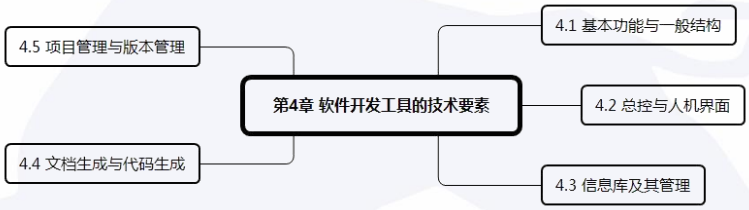
3.5 项目管理与版本管理

无论是质量、进度，还是资源调度，从项目的观点来说，最终都体现为成本的升高或降低。可以说，成本的情况是项目管理状况的综合的最终体现。★

含义 ★★	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、时间性的另一类管理任务。
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。
基本目标 ★★★	（1）使产品（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；（4）控制与降低成本。

软件项目管理的核心要素是质量。无论对于软件产品来说，还是对于应用系统来说，用户的反馈信息是项日管理与版本管理的重要资源。进度和质量对于软件开发来说，是最关键的问题。★

第 4 章 软件开发工具的技术要素



4.1 基本功能与一般结构

基本功能 ★★★	（1）提供描述软件状况及其发展过程的概念模式，协助开发人员认识软件工作的环境和要求，合理地组织与管理开发工作的工作过程。 （2）提供存储和管理有关信息的机制和手段，根据概念模式提供的信息库和人机界面，有效地控制这些信息。 （3）帮助使用者编制、生成及修改各种文档。 （4）通过各种信息的提供，半自动地生成程序代码，进行测试、修改错误。 （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 ★★	（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。

4.2 总控与人机界面

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）。
周期的阶段	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。
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。
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。

4.3 信息库及其管理

信息库的内容★★	信息库是软件开发工具的基础。从技术上说，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。（1）关于软件应用的领域与环境的状况。这些信息一般是在需求分析阶段收集并存入信息库的，它们主要用于分析设计阶段，作为形成下一类信息的原始材料。（2）设计成果，包括逻辑设计与物理设计的成果。这类信息是分析设计人员利用前一类信息，通过人机交互的方式形成的设计方案。它主要包括数据流程图、数据字典、系统结构图、数据库的逻辑设计、各模块的设计要求，以及由此形成的设计文档。（3）运行状况的记录。软件投入运行之后，应当对于它的运行情况进行详细地记录，包括它的运行效率、作用、用户反映、故障情况、故障的原因及处理情况。这些信息对于软件的有效运行与进一步发展是至关重要的。 （4）有关项目管理与版本管理的信息。这属于跨生命周期的信息，包括项目的进度、过程、人员分工、资源投入、版本组织等。
组织方式★	在信息库的管理方面，目前比较好的管理方式是逻辑上统一、物理上分散。
管理功能 ★★★	与一般数据库管理相同的功能：录入更新、使用查询、一致性维护。 与一般数据库管理系统的区别：（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
处理方法★	（1）历史信息的数量太大，占用存储设备过多——采用脱机备份的方法解决（2）历史信息格式不一致——加强标准化+智能方法解决。

4.4 文档生成与代码生成

代码生成 ★★	**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。输出程序代码是这个模块的目标。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环境下可运行的机器指令。一般来说，对话屏幕、输入屏幕、输出报表等类型的模块比较容易生成。**依据的资料**：（1）信息库中已有的有关资料；（2）利用各种标准模块的框架和构件；（3）依据使用者通过屏幕前的操作送入的信息。

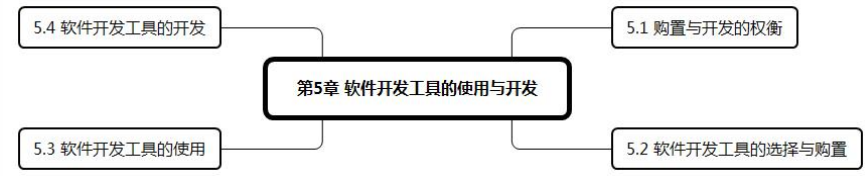
文档生成★	包括文章、表格、图形三大类。其中，最容易生成的是表格，其次是图形，最难处理的是文章。

4.5 项目管理与版本管理

主要内容：（1）研究确定开发工作的方针和方法。（2）开发任务的划分与分工。（3）资源状况。包括：人力、物力、设备、软件、资金。（4）人员情况。包括每个成员或团队的技术水平和工作进度。（5）变更情况。包括需求、环境、人员、技术、设备的变更。（6）质量情况。包括质量标准、如何检验。

项目数据库的内容应当支持项目负责人做好以上各项工作。

第 5 章 软件开发工具的使用与开发



5.1 购置与开发的权衡

购买现成的软件开发工具还是自己开发专用的工具，对于这个问题，不能简单地回答是或不是，它与具体工作的条件、环境、人员素质、项目特征都有不可分割的联系。购买市场上已有的软件开发工具的优点是减轻工作负担。

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
购置或自行开发的权衡因素★★★	（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。

自制工具十分普遍的，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。★★

5.2 软件开发工具的选择与购置

明确目的与要求★★★	在选择之前首先需要明确目的与要求。也就是说，自己首先要搞清楚此次引入软件开发工具要达到怎样的目标。（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、硬件环境下运行。
调查市场★	调查研究中的重点：（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是什么。（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。
购置方法与步骤★★★	（1）明确购买软件开发工具目的与要求。（2）明确购买软件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。

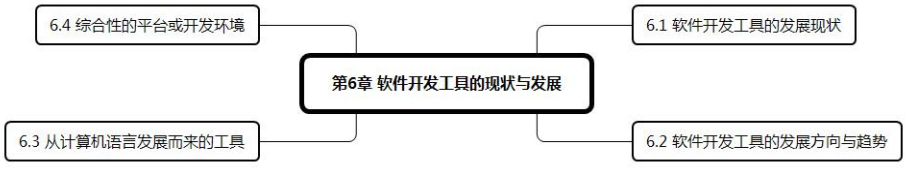
5.3 软件开发工具的使用

严格使用制度★★★	对于有关的各种信息，都要明确其来源、使用权限、维护职责等有关事宜。不言而喻，单纯一个抽象的模型或一个空的信息库对实际工作是毫无意义的。实质性的内容是与本软件开发有关的信息。一个项目组在自己的工作中使用软件开发工具时，必须明确规定各种有关的信息由哪些人在什么时候完成这种任务，而且必须对这些信息的准确性负责。另外，对于已经存入信息库的信息也要规定其使用权限及维护责任，即哪些人可以使用它，可以修改它。如果没有明确的规定，信息库的内容就失去了可靠性，工具的运用也就失去了基础。
记录使用的详细过程★★★	记录内容包括系统运行的次数、时间，信息库的输入与更新时间，各种输出的质量与数量，使用者的反映与满意程度，各种故障的情况及处理。不管用什么方式记录，项目的组织者必须及时地清楚地知道软件开发工具的使用情况、信息库的情况以及人们是否正确地使用了它们。
培训使用人员★	人员培训工作无疑是使用过程中十分重要的、不可缺少的一个部分。组织者应结合实际情况，以本组工作中实际的经验与教训为教材，强调软件开发工具的思想与方法。
经常进行审计与评价工作★★★	**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。**目的**就是为了做到心中有数、用好工具、保证取得实际的应用效果。**对购置开发工具进行审计的主要内容包括**：希望利用的功能，投入的资金和人力，定量计算的收益和无法定量计算的收益，从经济上看是否合理。

5.4 软件开发工具的开发★★★

不成功的商品化软件开发工具往往由于使用手册复杂。属于自行开发工具原则的是开拓新功能。注意事项：（1）首先要区分是为自己所用还是作为商品开发，至少在一个时期内二者必居其一，不应混淆。一般是为自己所用而开发工具，首先要注意从实际出发，设定现实的、有限的目标。（2）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。

第 6 章 软件开发工具的现状与发展



6.1 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。

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。
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，即没有广泛地使用这些工具。

6.2 软件开发工具的发展方向与趋势

值得注意的发展方向 ★★★	（1）**智能化。**所谓智能化，具体来说就是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。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**经验**与**知识**来补充或加工。（2）**网络化。**网络的应用是计算机应用领域中一个重要方向。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、知识重用提供了新的机会。目前利用网络提供条件，提高工作效率的软件开发工具，以及在网络上开发应用软件的工具，正在成为当前发展的一个热点。（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。（4）**标准化。**是指软件构件的标准化以及用标准构件组成大型软件结构的标准化。
软件工具的发展轨迹	（1）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。（2）**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。（3）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。**（4）作为一款著名状件工具,Java 虚拟机(JVM)出现于面向对象时代和互联网时带的交集。

计算机网络的普遍使用对软件开发工具的影响：通过将网络引入软件开发工具,软件开发人员可以更方便地互通信息,共享知识,更便于掌握项目的进展情况、质量状况等，这就给软件重用、知识重用提供了新的机会，也提高了工作放率。

6.3 从计算机语言发展而来的工具

较早期的软件开发工具基本着眼于某一种具体的语言本身，通过添加各种辅助功能发展出来的。对于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。★

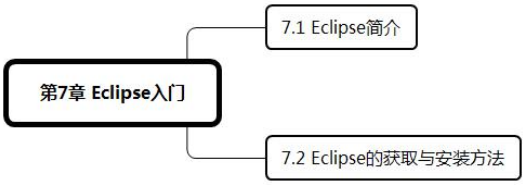
软件开发工具是个相当广泛的庞大谱系的原因：因为在这个谱系的一端是从某些计算机语言，通过增添各种辅助功能发展出来的工具；而在另一端，则是从较为抽象的概念模式或过程模式出发设计的开发平台或开发环境，是计算机语言在开发方向上的延伸。★★★

比较项目	VB	VC	PB	C++Builder / Delphi	JAVA
组件技术	COM，ActiveX	COM，ActiveX，CORBA	COM，JavaBean，Jaguar，UserObject	COM，ActiveX，CORBA	JavaBean,CORBA;ActiveX
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

6.4 综合性的平台或开发环境★★★★

Visual Studio 与 Eclipse 相比的优势：（1）基于操作系统是 Windows，优先考虑 Visual Studio。Visual Studio 在微软环境下和其他产品的交互协同能够浑然一体，天衣无缝；开发使用容易；产品间的差别透明度高；在运行速度，代码显示速度等方面也强于 Eclipse。（2）宏观上 Eclipse 更有优势。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。

第 7 章 Eclipse 入门



7.1 Eclipse 简介★★

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。

Eclipse 最初主要用来进行 **Java** 语言开发，但 Eclipse 并非只有这个用途。Eclipse 作为一个框架平台还包括插件开发环境（PDE），这个组件主要针对希望扩展 Eclipse 的软件开发人员，它允许软件开发人员构建与 Eclipse 环境无缝集成的工具。

Eclipse 的体系结构主要包括运行时内核（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）、其他插件等。其中其他插件包括帮助组件（具有与 Eclipse 平台本身相当的可扩展能力）、**团队支持组件**（负责提供版本控制和配置管理支持）等。

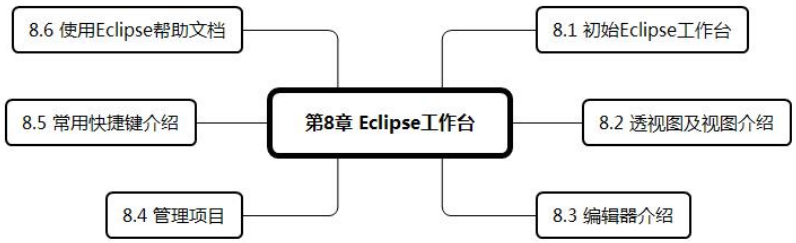
7.2.1 JDK 的获取与安装方法★★★★★

使用 Eclipse，首先需要安装 JDK。JDK 可以在 SUN 公司的网站下载。（1）在浏览器中输入相应网址，在页面的中部找到 JDK，单击 Download 按钮。（2）在新打开的页面，选择 Platform 下拉框为 Windows，并勾选，单击 Continue》按钮进入下一步。（3）在新打开的页面中，单击“jdk-6u15-windows-i586.exe”即可下载。（4）下载完成后，双击运行进入安装页面。（5）在“自定义安装”界面中，可选择安装目录及各个功能模块的按照方式，选择好目录及按照方式后，单击按钮后等待安装。（6）安装完成后，单击按钮即可完成 JDK 的安装。

7.2.2 Eclipse CDT 的获取与安装方法★★★★★

安装完 JDK 之后，开始安装专门用于编写 C 和 C++程序的 Eclipse 开发环境 CDT。它可以通过 Eclipse 网站上下载得到，下载地址为 http://www.eclipse.org/。（1）在浏览器中输入网址。（2）在下载页面，单击即可下载。（3）把下载的 eclipse-cpp-galileo-win32.zip 压缩包解压到磁盘，即完成 Eclipse CDT 的初步安装。（4）此时可以在 Eclipse 安装目录下看到目录结构。双击 eclipse.exe 文件，打开 Eclipse.进入 Java 安装目录，将其中的 jre 文件夹复制到 Eclipse 安装目录，再次运行 eclipse.exe 即可成功。

第 8 章 Eclipse 工作台



8.1 初识 Eclipse 工作台★★

Eclipse 工作台是一个高级用户界面框架，它为用户提供了一个**整体架构**和可扩展的**用户界面**。工作空间是 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。在使用 Eclipse 时，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

8.2 透视图及视图介绍

Eclipse 的工作台，主要有以下几个组成部分：**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**）**、工具栏、透视图。工具栏主要有两种类型，一种为主工具栏，另一种是视图工具栏。★★

透视图★	透视图占了 Eclipse 工作台的大部分空间，包括视图和编辑器。
视图★★	包括**导航器视图**（又称资源管理器视图，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）等。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，错误视图中编写代码时显示程序中的错误，提示用户及时改正，显示警告信息的程序能正确运行。
主要视图介绍★★	导航器视图；大纲视图；控制台视图；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。）；搜索视图（显示用户搜索结果的详细信息）；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。）

8.3 编辑器介绍★★

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。编辑器是用来处理各种文档的，其中用来打开网页文件的是 Web 浏览器。

常用的编辑器：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。

关闭编辑器的三种方法：关闭某个特定的编辑器；关闭某个特定编辑器；可以一次关闭全部打开的编辑器。

8.4 管理项目★★

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。

如果需要将已有的外部项目导入到 Eclipse 集成环境中，先点击菜单栏中的文件（File）菜单，然后选择导入(Import)选项

除了可以从外部导入项目外，Eclipse 还支持从外部导入**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**）**、**文件系统（**指操作系统文件夹中的各类文件**）**和**首选项文件（**可以让用户通过该配置文件来个性化定制 Eclipse**）。**

资源管理器中可以看到 Eclipse 当前加载的所有项目和项目中各个文件夹和文件的细节。用鼠标双击资源管理器中任意一个项目，可以以树形菜单的方式展开该项目下所有文件列表。

8.5 常用快捷键介绍★★

Eclipse 文本编辑器中查找下一个操作的快捷键是 Ctrl+K；重命名的操作的快捷键为 Alt+Shift +R；最基本搜索功能的快捷键是 Ctrl+F；调试程序的快捷键是 F11，运行程序的快捷键是 Ctrl+F11。

8.6 使用 Eclipse 帮助文档★

Eclipse 帮助文档是一种实时更新的联机帮助文档，还提供了动态帮助方式为用户提供有针对性的帮助信息。这是为了用户能更快地在帮助文档中找到相关帮助所提供的方式。

第 9 章 使用 Eclipse 进行 C/C++开发

9.1 安装 MinGW★★

C 语言是一种面向过程的计算机程序设计语言。它既具有高级语言的特点，又具有汇编语言的特点。

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。

方法：（1）下载 MinGW 安装向导。（2）下载完成后，MinGW 在线自动安装程序。（3）在窗口中可以选择下载并自动安装或仅仅下载，建议初次使用的用户选择下载并自动安装。（4）在两个窗口中分别单击 I Agree 和 Next 按钮保持默认设置。（5）在窗口选项中选择 Full，然后单击 Next 按钮。（6）在窗口中，单击按钮选择 MinGW 的安装路径后单击按钮。（7）等待 MinGW 自动安装程序下载并全部安装完成，安全期间保持网络畅通。（8）为了在不指明完整路径的情况下，系统能够认识 MinGW 的执行命令，需要设置系统环境变量。将 MinGW 安装路径下的 bin 文件夹输入带变量值中即可完成环境变量设置。（9）为了确认 MInGW 的安装和环境变量设置是否生效，选择“开始”→“运行”，在弹出的对话框中输入 cmd 后单击“确定”按钮，打开“命令提示符”。步骤完成后，Eclipse CDT 才能够编译并运行 C/C++程序。

9.2.1 新建 C/C++项目★★★★★

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（Select a wizard）。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。

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 文件夹，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。

在 Basic Settings 窗口中设置源代码的注释部分内容。分别在作者、版权声明等标签中输入相应内容后，单击 Next 按钮进入下一步。

为创建一个 C Project，在“Select Configurations”窗口中可以选择调试和发布文件夹以及高级设定，保持默认设置后单击“Finish”按钮即可新建 C Project。

9.2.2 新建 Source Folder ★★★★★

用鼠标右键单击资源浏览器中的 Hello World 项目，在弹出的菜单中选择新建（New）子菜单里的源文件夹选项。

在弹出的 New Source Folder 窗口中，输入文件夹名称，把光标定位到文件名中输入名称后单击“Finish”按钮即可完成。

9.2.3 新建 C++Class★★★★★

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。

9.2.4 编译 C 程序★★★★★

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。

9.2.5 运行 C 程序★★★★★

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。

9.2.6 使用浏览功能★★

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。

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，也可使用**<F3>**快捷键。

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。

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。

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。

单击菜单栏上的浏览“Navigate”菜单，选择打开元素（Open Element），也可以使用快捷键“**Ctrl+Shift+T**”。

9.3.1 重命名变量★★★★★

在一个复杂的项目系统中，存在继承、联合等复杂的应用和交错使用的代码，而编程中调整代码以避免冲突也是很频繁的工作。Eclipse 重构中的重命名功能解决了变量、类、函数等重命名的所有问题。

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。如果修改引起代码冲突，Eclipse 会弹出提示冲突情况的窗口，可以选择继续应用修改，或取消操作。一般在这种情况下，用户应修改重命名的变量或方法，避免冲突。

9.3.2 抽取方法★

在面向对象的程序设计思想中，组件和方法的重用是一个很重要的概念。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，同时使得方法的调用变得清晰和简单。

9.3.3 抽取常量★★

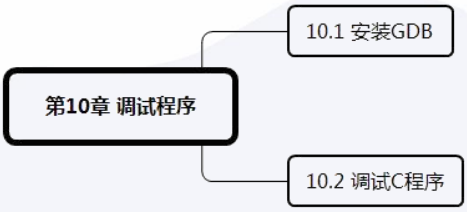
在很多复杂的程序中，也会存在对同一个数字或字符串的多次调用，每当出现这种情况我们应当用常量替换这个数字或字符串，以增加程序的可读性和易修改性。抽取常量本质上是将程序中的**数字**、字符等定义为常量，使得对于常量的调用和修改变得简单。

9.4.2 使用 Search 菜单进行搜索★★

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。

（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

第 10 章 调试程序



10.1.1 GDB 简介★

GDB 是 GNU 开源组织发布的一个强大的 UNIX 下的 C/C++程序调试工具。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。GDB 可以帮助用户实现的功能：（1）启动 C/C++程序，可以按照用户的自定义的要求运行和暂停程序（2）可让被调试的程序在用户所指定的调试的断点处停住。（3）当程序被停住时，用户可以检查此时引起程序中断的原因。（4）动态地改变程序的执行环境。（5）单步调试程序，在每个断点显示程序中各个变量的状态和值。

10.1.2 下载并安装 GDB★★

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。在安装 GDB 时，将 gdb. exe 解压后安装的文件夹是 bin。

安装 GDB 时，下载后寻找的文件注意**选择 bin 文件**而不是 src 文件进行下载。单击即可进入下载页面。

在安装 GDB 中，第一步打开浏览器，在地址栏输入下载地址后，选择上方的 Files 标签，进入下一步。该标签下包含了诸如 GCC、GDB 等软件的各个版本下载列表。

10.2 调试 C 程序★★

CDT 调试器	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。Eclipse 允许用户自定义调试视图，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。

设置行断点	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。用户可以在编辑器或断点视图中添加或删除断点，断点视图会清楚地显示每个断点的状态，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。（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 添加方法断点时，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”按钮即可激活断点视图。（2）**删除行断点。**在 GDB 删除行断点时，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。在 Eclipse CDT 调试器中,删除一个断点通常有 3 种方法,分别是：鼠标**双击**、鼠标右键弹出菜单和在断点视图中删除。（3）激活和禁用行断点。
设置方法断点	**（1）添加/删除方法断点。**当调试程序不需要在某个断点位置暂停时，可以删除该断点。删除方法断点与删除行断点类似，双击、鼠标右键弹出菜单、在断点视图中删除三种方式。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。**（2）激活/禁用方法断点。**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。
设置事件断点	断点是程序调试中的重要概念，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当前的状态。Eclipse 支持行断点，方法断点和事件断点。在程序运行过程中，发生特定问题时将程序暂停，称为事件断点。
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。
调试程序	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。单步遍历程序可以让用户逐行地运行整个程序。在多数情况下，单步遍历调试程序可以帮助程序员解决许多程序中棘手的问题。单步遍历程序主要包括单步跳入、**单步跳过**和**单步返回**。

第 11 章 Eclipse CDT 开发常用功能

11.1.1 设置首选项★★

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。

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。**

11.1.2 设置编辑器布局★★

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

11.2 定制工作台

（1）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。（2）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。（3）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。（4）复位透视图（5）保存透视图。（6）删除透视图

11.3 格式化代码★★

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。

设定完成之后，用户即可格式化代码。

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。

11.4 生成历史记录★★

Eclipse 不仅支持使用版本管理软件进行版本控制，还自带有保存历史记录功能，可以支持用户随时将程序恢复到之前某个时间点。

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。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 12 章 CVS 的安装及使用

12.1 CVS 介绍★★★

在版本控制软件中，可以将用户分为管理员和程序员两种角色，只有**管理员**可以将程序冻结

（Freeze）和解冻（Unfreeze），被冻结的程序是不允许修改的。

充分利用版本控制软件能够对软件开发进行卓有成效的管理，具体表现在：（1）随时将程序恢复到以前某一时间点。（2）实现程序的互斥性修改。（3）对程序修改进行有效的管理（4）将开发环境与测试环境、运行环境进行有效的隔离（5）评估软件开发人员编写的程序质量，控制软件开发的进度（6）管理文档

12.2 常用版本控制软件★★

不同于 CVS 和 VSS，ClearCase 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**。

Rational ClearCase	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；支持绝大多数操作系统；安装、配置、使用相对较复杂，需要进行团队培训。
SCCS	和 RCS 类似，也是早期的基于单一文件的版本维护系统。
CVS	简单易用、功能强大、跨平台、支持并发版本控制，免费；缺少相应的技术支持。
VisualSourceSafe	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	系列软件是 Merant 公司出品实现配置管理的 CASE 工具，可以为配置管理提供良好的自动化支持。其中 PVCS Version Manager 是用来实现文件的版本管理的，它是整个套件的核心。
Star Team	用于管理配置和变更的集成环境。
Firefly	管理、维护整个企业的软件资产，包括程序代码和相关文档。
Hansky Firefly	管理、维护整个企业的软件资产，包括程序代码和相关文档。
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。

12.3 CVS 的安装与配置★★

CVS 的安装	执行 CVS 安装程序后，在欢迎页面单击“Next”按钮进入下一步。在窗口中，选择需要安装的组件，按照安装程序提示逐步单击“Next”按钮即可完成按装。
CVS Control Panel 的设置	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**并对其进行控制管理的一个文件夹。
CVS 访问权限的设置	一般是开发小组的管理者给小组成员分配一些账户，而小组成员通过这些账户对 CVS 资源库进行访问，管理员对这些账户有**更改和删除**的权力。一般来说，给机器安装 CVS 的 Windows 系统管理员，同样也拥有 CVS 管理员的权限。所以安装好 CVS 之后，管理者可以直接使用该账户的用户名和密码登录到 CVS，对其进行管理维护。
将软件载入 CVS 资源库	载入成功后，CVS 就开始对我们的代码起到管理维护作用。（1）打开 Eclipse 的 CVS Perspective，创建一个新的 CVS 资源库连接。Host 为 CVS Server 所在的机器 IP 地址或 Host Name ;Repository path 即之前提到的查找该资源库使用的名字；连接该资源库还需被赋予权限。（2）回到 Java 视图，将项目载入 CVS 库中。用鼠标右键单击项目，在弹出的菜单中选择“Team”→“Share

	Project”.
CVS 的日常使用	（1）提出（Check Out）。首先从 CVS 资源库中复制一个软件镜像到本机的 workspace 中，代码和 CVS 上的代码一致。然后，在这份代码的基础上进行修改，CVS 能分辨出两者的区别，当提交代码时，需要进行一些融合工作。这些操作的术语为 Check Out。（2）代码同步（Synchronize）。经过一段时间的开发，本地的代码和 CVS 资源库的代码可能会产生差别，这种差别可能使本地代码与资源库代码之间存在增减或冲突等关系。此时，如果想要将本地代码与资源库代码一致，就需要选择**代码同步，即**“Team”→“**Synchronize with Repository**”。

CVS 常用术语★★

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 13 章 Eclipse 插件的使用与开发

13.1 插件简介★★

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。插件实际上是对

	原有软件的扩展，替应用程序增加一些所需的特定的功能。除了运行时的内核外，Eclipse 都是由若干插件组成的。
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。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。一个典型的插件文件夹一般包括：（1）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。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。

13.2 使用 PDE 进行插件开发★★

PDE 简介	随着插件功能的不断发展，插件清单文件可能会有上百行，开发者需要自己来协调不同插件的命名和属性设置等问题，所以插件开发环境应运而生。PDE 新增了透视图来帮助开发者创建、开发、测试、调试和部属插件。
PDE 基本操作	在默认的资源透视图中，单击“窗口”→“打开透视图”→“其他”，出现“打开透视图”窗口。从列表中选择“插件开发”，单击“确定”按钮打开 PDE 透视图。PDE 透视图主要包括清单编辑器和插件大纲两部分。（1）清单编辑器。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）。（2）插件模板。分类：Hello，world 模板；带有样本帮助内容的插件；具有编辑器的插件；具有弹出菜单的插件；具有多页面编辑器的插件；具有视图的插件；具有属性页面的插件；具有增量项目构建器的插件。

13.3 常用插件扩展点★★

在 Eclipse 中，常见的扩展点很多，Eclipse 用户界面的三个基本构成元素：视图、编辑器和透视图。

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。Org.eclipse.ui.views 扩展点用于为工作台定义更多视图。视图是工作台面内的可视组件。用户可以选择“窗口”→“显示视图”菜单显示视图，也可以从视图局部标题栏中关闭视图。
编辑器扩展点	通过扩展 org.eclipse.ui.sditors 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。
透视图扩展	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

点	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。用户可通过调用“窗口”菜单的“打开透视图”子菜单来选择透视图。

13.4 常用插件介绍与使用★★

表 13-1 常用插件

插件名称	类别	功用
CDT	代码类	提供功能完全的 C/C++集成开发环境
MDT	建模类	用于创建工业标准原型，提供了基于原型来开发模型的仿真工具
PDT	语言类	为 Eclipse 平台提供一个 PHP 开发工具框架。包含了开发 PHP 的所有开发组件，易于扩展
VE	UI 类	允许通过一个完全的 WYSIWYG 图形化编辑器来创建 SWT/AWT/Swing 应用程序
EMF	建模类	EMF 用于定义和实现结构化模型的框架
GEF	UI 类	是一个功能强大的可视化模型编辑框架，用于快速开发图形编辑器
XMLBuddy	XML 类	用于编辑 XML 文件
Code Analysis	代码分析类	分析 Java 工程的依赖性，拥有自己的透视图，以清晰的方式通过一系列图表来显示分析结果
Log4E	代码管理类	为了更好地配置项目日志
Lomboz	J2EE 类	可以帮助开发者使用 Eclipse 建立、测试、部署 J2EE 应用
DBEdit	数据库类	提供视图、数据库编辑和设计功能
Raman VideoPlayer，Eclipse games	娱乐类	提供视图和游戏类插件

EMF 是基于 Eclipse 的模型框架。它是 Eclipse MDA 的一个重要组成部分，是 Eclipse 中许多项目的基础， EMF 可以将模型转换成高效的，正确的，和易于定制的 Java 代码。EMF 可以实现的四个功能是：输入、代码生成、默认的持久化机制和**模型编辑器**。

第 14 章 常用建模工具

14.1 UML 建模介绍

统一建模语言（UML） 是一种面向对象的建模语言，提供了描述软件系统模型的概念和图形表示法。

面向对象方法基础★★★	（1）当前人们所要开发的信息系统不同于以前，它们在功能等诸多方面都变得很复杂且灵活多变，系统的边界也更难以界定。**复杂性**、**多样性**和**相互关联性**是各个信息系统的重要特征。于是，面向对象方法应运而生。面对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**。面向对象方法

	简称为 OO 方法。它由面向对象分析（OOA）、面向对象设计（OOD）和面向对象程序设计（OOP）组成。 特点：强调对现实世界的理解和模拟，把现实世界到信息世界的转化工作减少到最小 ,所以特别适用于系统分析和系统设计。（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**。**多态**指不同事物具有不同表现形式的能力，多态机制使具有不同内部结构的对象可以共享相同的外部接口，这样又很好地支持了消息机制，而不同对象接收到同一个消息可产生完全不同的结果也是多态的一种体现形式。**类**是对一组几乎相同的对象的描述；**方法**是驻留在对象中的过程。面向对象思想是高级程序语言中重要的编写思想。**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。**抽象**是人们认识客观世界中复杂性的一种基本方法。**方法**是驻留在对象中的过程。现在面向对象方法中，完成一件事的方法就是向有关对象发送消息。（3）**面向对象法与传统方法相比**：解决了信息系统工程中的两个主要问题：软件维护的复杂性和提高生产效率。使软件开发的风险降低、知识重用度提高。更加接近自然。更侧重建模而非分析流程。（4）**一般步骤**：1）标识和定义对象及类； 2）组织类间的关系； 3）在类层中构造框架； 4）建立可重用的类库和应用程序框架。（5）面向对象的系统分析设计方法：Booch 方法；OMT 方法；Coad/Yourdon 方法；OOSE 方法。
组件思想★	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，能够与其他组件或硬编码一道很快生成定制的应用程序。目标是粗粒度的复用，核心是接口。
UML 语言与建模工具★★	逻辑视图用于描述系统内实现的逻辑功能。它既描述了系统的静态结构关系，也描述了系统内的动态协作关系。**UML 可以应用在以下系统**：信息系统；技术系统；嵌入式实时系统；分布系统；系统软件；商务系统。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**。**UML 组成**包括**视图**（用例视图；逻辑视图；组件视图；配置视图）、**图表**（用例图；类图；对象图；状态图；顺序图；协作图；活动图；组件图；配置图）、**模型元素**、**基本机制**（有修饰、注释和说明三种方式）。

14.2.2 面向对象的分析设计和 Rational Rose★★

采用 Rational Rose 建模的过程：（1）确认应用系统的功能需求，并为事务处理原则建模；（2）对抽象的对象映射需求，提供设计模板并创建惯用的模板；（3）分辨和设计对象（或划分三层模型的服务）；（4）对软件的组成部分映射成对象并设计组件在网络上如何分布。★★★

就整个软件系统的整体结构而言，当前应用最多的还是客户机/服务器方式。当然，浏览器/应用服务器/数据服务器方式也是当前较为流行的体系结构。Rational Rose 采用的三层解决方案，是由**用户接口层**、**事务处理原则层**和**数据层**组成的应用模型。这种较抽象的分层结构满足了当前应用的需求。★★

采用 Rational Rose 实现建模的主要问题：（1）何时需要建模；（2）兼容性问题；（3）对 UML 的支持程度；（4）对大型项目的特殊支持；（5）采用可视化建模。★★★

14.2.3 Rational Rose 可视化建模的特点★★

1）支持 UML 的建模。 2）采用基于组件的开发。3）支持多语言开发。 4）支持双向工程。5）全面的团队支持。 6）简单易用。7）提供可视化的差异比较以及合并工具。 8）提供框架

向导。9）提供扩展接口，以实现定制的 Rose。 10）基本报告生成。可以生成数据词典。11）COBRA/IDL 生成。 12）数据库模式生成。13）微软存储库集成。 14）实现 Oracle 8 的正向和逆向工程。15）支持 Forte 附加项，以实现在此环境下的分析、构建企业级应用。

14.3 使用 Rational Rose 建模★★

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。用例视图设计主要是借助用例图、活动图、状态图来了解用户的需求。逻辑设计要用到**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）**。在系统物理设计阶段，要借助**部署图**等视图，确定系统的物理体系结构。

需求分析	**用例视图**（Use Case View）主要通过用例来描述系统的功能性要求，它是系统中与实现无关的视图。**活动图**本质上是流程图，很好地描述了系统的活动、判定点、先后顺序和分支等，是一种能够清晰描述系统功能流程的工具，是用例图的很好补充。
系统分析与设计	**（1）顺序图**是强调消息时间顺序的交互图。顺序图描述类系统中类和类之间的交互，它将这些交互建模成消息交换。换句话说，顺序图描述了类以及类之间相互交换以完成期望行为的消息。（2）协作图。是强调参加交互的各对象的组织。只对相互间有交互作用的对象和这些对象间的关系建模。（3）组件图。在 UML 建模过程中，系统设计相关的主要视图为类图和组件图。系统组件图描述了软件的各种组件和它们之间的依赖关系。（4）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。

配置图很好地描述实施时整个系统的结构与层次，并且描述系统中软件和硬件的物理结构。

14.4.1 安装 EclipseUML★★★

EclipseUML 是 Eclipse 提供的用于 UML 建模的一整套插件。安装方式：（1）用户可以利用 Eclipse 提供的自动升级程序安装组件。（2）用户自行到网上下载，下载地址为 http://eclipsedownload.com/。1）打开浏览器，在地址栏输入上述地址，单击页面中的下载图标即可下载。2）下载完成后，单击“开始”→“运行”，在弹出的对话框输入 cmd。3）输入安装命令后，在界面中选择安装语言，默认英语。然后单击“OK”按钮进入下一步。直接单击“Next”按钮进入下一步。4）在地址栏输入 EclipseUML 的安装位置，然后单击“Next”按钮进入下一步。5）等待一段时间后，安装程序即可完成安装。

14.4.2 使用 EclipseUML 进行建模★★★

（1）新建 UML 项目（2）新建文件夹（3）用例图示例（4）类图示例（5）状态图示例。