

Microsoft

微软院校认证课程系列教材

程序设计导论

Microsoft 著



高等教育出版社
HIGHER EDUCATION PRESS

微软院校认证课程系列教材

程序设计导论

Microsoft 著



高等教育出版社

编审委员会 刘志鹏 朱之文 田本和 王军伟 郑祖宪 王 林

Jay Henningsen Angie Fultz Anne Hamilton

组织策划 田本和 尹 洪 张思挚 蒋 斌 Samantha Smith

丁增珣 王静亚

技术编审 蒋 斌 蔡 锴 李 岷 肖凤杰 马国屏

Angie Fultz Anne Hamilton 黄 燕 冯 英

古 锋 李朝晖 喻艺丹 李靖国

Microsoft Official Curriculum 最终用户许可协议

重要须知——请认真阅读——您一旦打开“许可使用内容”包装的密封或以其他方式使用此处的“许可使用内容”，即表示您同意接受本《协议》各项条款的约束：本 Microsoft Official Curriculum 可能包含 Microsoft 或其供应商提供的软件或其他材料（总称“许可使用内容”），其使用应遵守以下各项 Microsoft 提示条款。每个软件程序都受一份最终用户许可协议（《协议》）的约束，而该《协议》是您（个人或单一实体）（“最终用户”）和 Microsoft Corporation（“Microsoft”）之间就允许使用软件及相关介质或印刷材料、“联机”或电子文档和基于 Internet 的服务达成的一份法律协议。本《协议》的修正条款或补充条款可能随软件一起提供。您一旦安装、复制或以其他方式使用“许可使用内容”，即表示您同意接受本《协议》各项条款的约束。如果您不同意，请（a）不要打开“许可使用内容”包装的密封；（b）不要使用软件、文档或其他材料，并且（c）退还“许可使用内容”。

“许可使用内容”随带软件的特别提示

许可证的授予。为与本“许可使用内容”一起使用而提供的任何软件（“软件”）都是 Microsoft Corporation 和（或）其供应商享有著作权的作品。“软件”只授予使用许可，而非出售。任何特定“软件”的使用都应遵守以下各《许可协议》中的一份《许可协议》：

（1）一般使用许可。Microsoft 授予最终用户一份有限的、非独家拥有的、免版权费的许可证，许可其在一台由一位单一用户随时使用或访问的单一计算机上安装和使用“软件”的一份副本，并且最终用户：（a）不得修改“软件”，但下文有明确规定时例外；（b）不得发行“软件”或其任何组成部分；（c）不得出借、出租、租赁、出售、分许可、转让“软件”或将“软件”随附的任何印刷材料用于提供商业运营服务；（d）不得在收费的公立或私立课程中使用“软件”；（e）不得对“软件”进行反向工程、反编译或反汇编；尽管有此项限制，但如果适用法律明确允许上述活动并且仅在适用法律明示允许上述活动的范围内，则例外；并且（f）不得转让“软件”的各项权利，除非本《协议》明确规定。

Microsoft 保留一切其他权利。Microsoft 及其供应商保留“软件”的一切产权和所有权，并且不转让或许可使用“软件”或其任何组成部分的任何权利，除非本《协议》具体说明。

（2）替代使用许可。上述规定的一般使用许可将被任何具体“软件”随附或包括的《许可协议》（如果有）的各项条款取代或替代。除非最终用户首先同意《许可协议》的各项条款，否则将无法安装附带或包括该《许可协议》的“软件”。Microsoft 保留一切其他权利。Microsoft 及其供应商保留“软件”的一切产权和所有权，并且不转让或许可使用“软件”或其任何组成部分的任何权利，除非本《协议》明确说明。

（3）样本代码使用许可。如果将特定代码或一个样本应用程序作为“许可使用内容”中包括的实验室练习的部分提供（“样本代码”），则这类“样本代码”以“现有状况”被提供，并且没有任何类型的保证。Microsoft 授予您一份有限的、非独家拥有的、免版权费的许

可证，许可您为了个人使用的目的而安装、使用、修改和复制“样本代码”，条件是您不得：(a) 发行“样本代码”或其任何组成部分；(b) 出借、出租、租赁、出售、分许可或转让“样本代码”；(c) 在收费的公立或私立课程中使用“样本代码”；并且(或者)(d) 转让“样本代码”的任何权利。如果您修改“样本代码”，您应该根据 Microsoft 的请求，自付费用为因您或代表您对“样本代码”做出的任何修改而使 Microsoft 和 Microsoft 的分公司、关联公司、董事、高级主管、员工、代理商和独立供应商面临的任何索赔或诉讼提供辩护，并且您须赔偿 Microsoft 因这类索赔而招致的任何费用、损害赔偿和手续费方面的合理开支（其中包括但不限于律师费和其他专业人士收取的费用），并使其免受任何损害。Microsoft 应：(a) 以书面形式就任何这类索赔或诉讼向您提供合理的及时提示，并且允许您通过 Microsoft 和您双方都接受的律师对这类索赔或诉讼进行答辩和辩护；(b) 在您支付费用的情况下向您提供信息、协助和授权，以帮助您为这类索赔或诉讼进行辩护。您不对 Microsoft 在未经您书面允许的情况下做出的任何和解负责，但您不得以不合理的方式拒绝给予这样的允许。

其他许可限制。安装“软件”仅供最终用户根据适用的《许可协议》使用，并且除非以其他方式在另外一份协议中达成一致意见，否则得不到 Microsoft 或其供应商提供的技术或其他支持服务。法律明确规定：禁止在违反《许可协议》的情况下对“软件”进行任何复制或再发行。明确禁止为进一步复制或再发行软件而将“软件”复制到任何服务器或地点。

美国政府许可使用权利。根据 1995 年 12 月 1 日当天或之后签发的请求而提供给美国政府的所有软件，均根据本协议其他部分规定的商业许可使用权利和限制予以提供。根据 1995 年 12 月 1 日之前签发的请求而提供给美国政府的所有软件，视情况根据 FAR, 48 CFR 52.227-14 (1987 年 6 月) 或 DFAR, 48CFR252.227-7013 (1988 年 10 月) 中规定的“限制权利”予以提供。

免责条款。“软件”仅根据《许可协议》的各项条款对“软件”提供保证（如果提供保证的话）。除非在《许可协议》中提供保证，否则 Microsoft Corporation 和（或）其供应商就“软件”不提供任何的保证和条件，包括适销性、适用性、所有权和不侵权的所有默示保证和条件。

“许可使用内容”随带文档和（或）其他材料的具体说明

允许从“许可使用内容”（“文档”）中打印文档（如实验室说明等），条件是：(a) 将这类文档用于您的个人培训，并且不得再出版或在任何网络计算机上张贴或以任何介质形式广播这类文档，并且(b) 不得对任何文档做出任何修改。

明确禁止对任何介质上包含的作为“许可使用内容”组成部分的录像、录音、图形和（或）任何其他材料（“其它材料”）进行任何复制或再发行。

“许可使用内容”的各组成部分均受商业包装法律和其他法律的保护，并且不得全部或部分予以复制或模仿。除非 Microsoft 明示允许，否则不得复制或转发“许可使用内容”中的任何徽标、图形、声音或图像。

无保证。Microsoft 和（或）其供应商不对“许可使用内容”中不论为任何目的而可能包含的文档或其他材料中的信息、音像或任何其他内容是否合适提供任何保证，无论该类文档、

信息、音像或任何其他内容是何目的。所有这类文档和其他材料均以“现有状况”提供，没有任何类型的保证。Microsoft 和（或）其供应商特此就文档和其他材料不提供任何的保证和条件，包括适销性、适用性、所有权和不侵权的所有默示保证和条件。

有关第三方站点链接的说明

至第三方站点的链接。您可以使用“许可使用内容”链接至第三方站点。第三方站点不由 Microsoft 控制，并且 Microsoft 不对任何第三方站点的内容、第三方站点包含的任何链接或第三方站点的任何更改或更新负责。Microsoft 不对从任何第三方站点收到的网站广播或任何其他形式的传输负责。Microsoft 仅为了您的方便向您提供这些至第三方站点的链接，并且包括任何链接并不暗示 Microsoft 认可相应的第三方站点。

有关全部“许可使用内容”的说明

“许可使用内容”中包括的“软件”、文档和其他材料可能包含不准确的技术内容或印刷错误。可能定期对内容进行修订。Microsoft 可随时在不提供通知的情况下对“许可使用内容”中规定的产品和（或）程序进行改进和（或）更改。

免责条款。除非另行说明，否则本《协议》提及的公司、产品、人物、特性和（或）数据均属虚构，并且无意以任何方式代表任何真实的个人、公司、产品或活动。

保留权利和所有权。Microsoft 保留未在本《协议》中明示授予您的一切权利。“许可使用内容”受著作权和其他知识产权法律及条约的保护。Microsoft 或其供应商拥有“许可使用内容”和其中组件的所有权、著作权和其他知识产权。

同意使用数据。您同意：Microsoft 及其关联公司可以收集和使用作为提供给您的产品支持服务的一部分而收集的与“许可使用内容”相关的技术信息（如果有）。Microsoft 可以将此信息仅用于改进我们的产品或为您提供定制的服务或技术，并且不会以能识别您身份的方式披露此信息。

额外软件/服务。除非我们随下列更新、增补、补充组件或基于 Internet 的服务组件一起提供其他应适用的条款，否则本《协议》适用于 Microsoft 在您获得“许可使用内容”的初始副本之日后可能提供给您的或为您准备的“许可使用内容”的更新、增补、补充组件或基于 Internet 的服务组件。就通过使用“许可使用内容”而提供给您的或为您准备的任何基于 Internet 的服务而言，Microsoft 保留停止这类服务的权利。

出口限制。您承认“软件”受美国出口法律管辖。您同意遵守所有适用于“软件”的国际法和国内法，其中包括美国出口管理条例以及由美国和其他国家（地区）政府颁发的最终用户、最终使用和目的地方面的限制。要了解详情，请访问 <http://www.microsoft.com/exporting/> 网站。

许可使用内容的转让。“许可使用内容”的原始最终用户可以将本《协议》和“许可使用内容”永久性地一次直接转让给另外一位最终用户，条件是该原始用户不得保留“许可使用内容”的任何副本，并且必须转让“许可使用内容”的所有部分（包括全部组件、介质及

印刷材料、任何升级版本、各《许可协议》和（如果适用）正版标签）。这种转让不得为非直接转让，如以寄售方式转让。在转让之前，接收“许可使用内容”的最终用户必须同意遵守《协议》的各项条款。如果“许可使用内容”是一个升级版本，任何转让都必须包括“许可使用内容”的所有先前版本。

终止。如果您未遵守本《协议》的各项条款和条件，在不损害其他权利的情况下，Microsoft 可终止本《协议》。如此类情况发生，您必须销毁“许可使用内容”的所有副本及其全部组成部分。

适用法律。本《协议》受中华人民共和国法律管辖。

责任限制。在适用法律所允许的最大范围内，无论损害赔偿是否在履行合约、出现疏忽或发生其他侵权行为时发生，Microsoft 和（或）其供应商绝不就因“许可使用内容”的任何组成部分或所有组成部分的使用或性能、因提供或未能提供服务、或因可从“许可使用内容”得到的信息而引起的或有关的任何特殊的、间接的、或特定的损害赔偿或任何损害赔偿（包括但不限于因营业中断，因使用、数据或利润的丧失，或因任何其他金钱上的损失而造成的损害赔偿）承担赔偿责任。在任何情况下，Microsoft 的全部责任以及您获得的惟一赔偿将限于为“许可使用内容”实际支付的款额或五美元（U.S.\$5.00）以两者中的较高款额为准；但是，如果您已经签订了一份 Microsoft 服务协议，Microsoft 对这类服务的全部责任将遵守该协议各项条款的规定。由于某些国家和地区不允许排除或限制责任，上述限制条款可能不适用于您。

全部协议；规定可分割性。本《协议》（包括随“许可使用内容”提供的本《协议》的任何补充条款或修正条款）是您与 Microsoft 之间就“许可使用内容”和支持服务（如果有）达成的全部协议，并且取代“许可使用内容”或本《协议》中所包含的任何其他标的之所有先前或同时存在的口头或书面的通信、建议和声明。如果任何 Microsoft 支持服务的政策或计划的条款与本《协议》的条款有冲突，以本《协议》的条款为准。如果本《协议》的任何条款被认定为作废、无效、不能执行或非法，其他条款应继续完全有效。

如果您对本《协议》有任何疑问，或者如果您由于某种原因希望与 Microsoft 联系，请使用“许可使用内容”中附带的地址信息与微软（中国）有限公司联系，或在<http://www.microsoft.com> 网站访问 Microsoft。

准则和定义

“许可使用内容”是一种专门设计的培训工具，供 Microsoft Certified Technical Education Center (Microsoft CTEC)、Microsoft Certified Partner (MCP)、Microsoft 认证培训讲师 (MCT)、IT Academy 计划成员和 Microsoft 可能随时以书面形式指定的其他机构使用。“许可使用内容”旨在使 Microsoft 的技术培训渠道能够向计算机专业人士提供系统、支持和开发培训课程。为了取得最佳成果，“许可使用内容”应该由 Microsoft 认证培训讲师 (MCT) 在课堂环境或在线学习环境中讲授。

Microsoft Official Curriculum (MOC)：由 Microsoft 开发的系列课程材料，用于提供 Microsoft 产品和技术的培训和解决方案。

Microsoft 认证培训讲师 (MCT)：具备必要的教学和技术能力并且由 Microsoft 认证为

能够通过 Microsoft CTEC 讲授 Microsoft Official Curriculum 的个人。

Microsoft Certified Technical Education Center (Microsoft CTEC)：已经符合 Microsoft 对指定其为下列场所的资格要求：（a）一处 Microsoft Certified Partner (MCP) 营业点、和（b）一处提供 Microsoft CTEC 服务的任何场所。这些培训中心使用 MCT 向学生提供 MOC 课程培训。

Microsoft Certified Partner：已经符合被指定为 Microsoft Certified Partner 的资格要求的任何场所。

IT Academy 计划成员：已经符合被指定为 IT Academy 计划成员的资格要求的任何院校。

目 录

第 1 章 计算机程序介绍	1
1.1 软件开发概述	1
1.1.1 计算机程序的定义	1
1.1.2 软件的工作方式	2
1.1.3 程序的执行过程	2
1.2 计算机程序执行的阶段	4
1.2.1 输入阶段如何工作	5
1.2.2 处理阶段如何工作	5
1.2.3 输出阶段如何工作	5
1.2.4 课堂练习 确定程序阶段	5
1.3 解决问题的技术	5
1.3.1 解决问题的方法	5
1.3.2 流程图	7
1.3.3 示例 画流程图	8
1.3.4 课堂练习 计算总销售量	9
1.3.5 伪代码	10
1.3.6 示例 使用“伪码编译器”	11
1.4 程序类型概述	12
1.4.1 基于 Windows 程序的功能	12
1.4.2 课堂练习 使用 Windows 程序	13
1.4.3 Web 程序的功能	13
1.4.4 课堂练习 使用 Web 程序	14
实验 确定并在代码中识别程序阶段	14
练习 1 确定程序阶段	14
练习 2 在代码里查找程序阶段	15
习题	16
第 2 章 脚本简介	17
2.1 Web 脚本	17
2.1.1 Web 基础概述	17
2.1.2 课堂练习 识别 IP 地址	19
2.1.3 HTML 简介	20
2.1.4 课堂练习 建立网页	22
2.1.5 Web 脚本简介	24
2.1.6 示例 编写简单的 Web 脚本	25
2.2 宏脚本	28
2.2.1 宏简介	28
2.2.2 示例 编写 Word 宏	29
2.3 Windows 脚本	30
2.3.1 Windows 脚本简介	30
2.3.2 示例 编写简单 Windows 脚本	31
实验 编写脚本	32
练习 1 用 Web 脚本建立“Hello World”	32
练习 2 用宏建立“Hello World”	33
练习 3 用 Windows 脚本建立“Hello World”	33
习题	34
第 3 章 数据和变量	35
3.1 变量	35
3.1.1 变量的定义	35
3.1.2 变量类型	38
3.1.3 数据类型	38
3.1.4 命名变量的指导思想	39
3.1.5 变量的声明	39
3.1.6 变量的初始化	40
3.1.7 课堂练习 确定、声明和初始化变量	41
3.2 运算符	43
3.2.1 算术运算符	43
3.2.2 关系运算符	43
3.2.3 逻辑运算符	44
3.2.4 课堂练习 使用运算符	45
3.3 编程语法	46
3.3.1 关键字	46
3.3.2 课堂练习 识别关键字	46
实验 算术运算符	47

练习 计算雇员的年收入	47	6.1 设计用户界面	82
习题	48	6.1.1 用户界面简介	82
第4章 使用程序逻辑	49	6.1.2 用户界面的组件	83
4.1 使用分支	49	6.1.3 优秀用户界面的特性	84
4.1.1 使用 IF 语句	49	6.2 建立用户界面	87
4.1.2 示例 绘制流程图表示 IF 语句	52	6.2.1 示例 使用 Visual Studio .NET 创建应用程序	87
4.1.3 课堂练习 使用 IF 语句	53	6.2.2 在 Windows 应用程序里添加用户 控件	88
4.2 使用循环	55	6.2.3 使用 Visual Studio .NET 在窗体上 添加控件	89
4.2.1 使用 FOR 循环	55	6.2.4 在 Web 应用程序里添加用户 控件	91
4.2.2 课堂练习 显示前 8 个偶数的 乘积	58	实验 建立用户界面	91
4.2.3 使用 WHILE 循环	58	练习 建立 Web 界面	92
4.2.4 决定使用何种循环语句的指导 思想	60	习题	93
4.3 识别语法错误和逻辑错误	60	第7章 数据处理简介	94
4.3.1 逻辑错误简介	60	7.1 数据管理简介	94
4.3.2 课堂练习 识别逻辑错误	62	7.1.1 外部数据存储设备	94
实验 根据面试结果选择候选人	63	7.1.2 文件	95
练习 编写 VB.NET 代码	63	7.1.3 记录	96
习题	64	7.1.4 数据库	96
第5章 使用过程和函数	66	7.2 实体关系模型	97
5.1 使用过程	66	7.2.1 实体	97
5.1.1 过程的定义	66	7.2.2 属性	98
5.1.2 建立过程	67	7.2.3 关系	98
5.1.3 使用过程	69	7.2.4 关系类型	99
5.1.4 向过程传递数据	71	7.2.5 子类型和超类型	100
5.1.5 示例 使用过程	72	7.2.6 表	101
5.2 使用函数	73	7.3 使用键对数据排序	102
5.2.1 函数的定义	73	7.3.1 主键	102
5.2.2 什么是预定义函数	74	7.3.2 备用键	103
5.2.3 建立函数	74	7.3.3 复合键	103
5.2.4 函数的使用	76	7.3.4 外键	103
5.2.5 课堂练习 使用函数	79	7.3.5 用 Microsoft Visio 创建实体 关系图	104
实验 识别函数并向过程传递参数	79	7.3.6 课堂练习 用表描述数据关系	105
练习 1 识别函数	80	7.4 XML	106
练习 2 向过程传递参数	80		
习题	81		
第6章 开发用户界面	82		

7.4.1 XML 简介	106	9.3.5 UML 视图和软件开发生命周期 的关系	132
7.4.2 使用 XML	107	9.4 开发用例图	132
7.4.3 HTML 和 XML 的区别	107	9.4.1 用例简介	132
7.4.4 XML 的常见用途	108	9.4.2 主角简介	133
7.5 建立 XML 文档	109	9.4.3 主角的识别	134
7.5.1 XML 标记	109	9.4.4 用例图简介	134
7.5.2 XML 文档的内容	111	9.4.5 用例图的绘制	135
7.5.3 示例 使用 XML	112	9.4.6 示例 建立用例图	136
习题	113	9.4.7 课堂练习 在内部工作公告过程 中识别主角和用例	137
第 8 章 编程方法	114	9.5 其他 UML 图	138
8.1 面向过程的编程方法	114	9.5.1 绘制类图的方法	138
8.1.1 面向过程编程	114	9.5.2 绘制对象图的方法	140
8.1.2 面向过程编程方法的局限性	115	9.5.3 协作图简介	141
8.2 面向对象的编程方法	116	9.5.4 顺序图简介	143
8.2.1 面向对象编程	116	实验 建立用例图	145
8.2.2 类、对象、消息与封装	117	练习 1 识别用例	145
8.2.3 抽象	119	练习 2 建立用例图	145
8.2.4 继承和组合	120	习题	146
实验 类	121	第 10 章 进阶学习目标	147
练习 识别类	121	10.1 回顾	147
习题	123	10.1.1 脚本类型	147
第 9 章 软件开发过程简介	124	10.1.2 编程基础	148
9.1 生命周期模型	124	10.2 展望	150
9.1.1 瀑布模型	124	10.2.1 .NET 中的高级 Windows 编程	151
9.1.2 螺旋模型	125	10.2.2 ASP.NET Web 应用程序开发	151
9.2 软件开发阶段	125	10.2.3 使用 ADO.NET 的数据库编程	151
9.2.1 构思阶段	126	10.2.4 ASP.NET Web Service	152
9.2.2 计划阶段	126	10.2.5 示例 Windows 应用程序、Web 应用程序和 XML Web Service	152
9.2.3 开发阶段	127	实验 编程基础总结	153
9.2.4 稳定阶段	127	练习 1 开发基于 Windows 的应用程序	154
9.2.5 软件开发过程中的角色	128	练习 2 开发基于 Web 的应用程序	163
9.3 统一建模语言简介	129	词汇表	170
9.3.1 UML 的定义	129		
9.3.2 UML 的用途	130		
9.3.3 UML 视图	130		
9.3.4 UML 图	131		

第 1 章 计算机程序介绍

本章中介绍的基本编程概念能够帮助读者理解表达程序逻辑的技术。在用计算机解决现实问题时将会使用这些逻辑。

学习完本章后，读者将能够：

- 了解软件如何工作
- 识别计算机程序执行的 3 个阶段
- 描述各种解决问题的技术
- 定义计算机程序和编程语言

1.1 软件开发概述

计算机程序也称为软件。要理解软件开发，就需要知道什么是计算机程序以及程序是如何工作的。

计算机具有快速而精确地完成数学计算的能力。要执行这些计算，计算机需要运行专门为某个任务编写的计算机程序。

计算机程序定义了一组指令集。计算机根据指令集中的指令要求对输入数据进行处理，处理的结果将作为输出数据。计算机程序员是为各种任务编写指令集的专家。

在本节中，将学习计算机程序的定义和计算机程序可以实现的功能。

学习完本节后，将能够：

- 区分计算机程序
- 描述软件如何工作
- 描述程序的执行过程

1.1.1 计算机程序的定义

计算机程序是一组精确地告诉计算机执行什么操作和什么时候执行操作的连续指令集。由于各项任务的复杂程度和时间长度存在差异，因此计算机程序的大小也各不相同。

程序可使计算机具备特定的功能。任务可以由单个程序或者协同工作的多个程序完成。当几个程序一起工作可以提供更为强大的功能时，常常把这些程序组合成一个软件包。例如 Microsoft® Word 和 Microsoft® Excel 就组合在一个名为 Microsoft Office 的软件包里一同出售。

计算机程序可分为 3 种类型。

(1) 应用程序

商务问题的解决方案通常由一个或多个小任务组成。要解决一个商务问题，需要多个程序

一起构成一个应用程序。例如，财务应用程序通常包含雇员工资处理程序、薪水支票生成程序和报表打印程序。

(2) 操作系统

任何应用程序都需要一个可以支持其运行的操作系统。操作系统负责与硬件通信以支持各种程序命令的执行。操作系统向应用程序提供其请求的资源，比如内存、磁盘空间，并协助应用程序访问诸如监视器和打印机等输出设备。

(3) 设备驱动程序

操作系统需要通过设备驱动程序来与硬件设备通信。设备驱动程序是帮助操作系统与计算机中的硬件组件进行通信的应用程序。设备驱动程序管理硬件设备，比如打印机、监视器、调制解调器和磁盘驱动器。设备驱动程序将与硬件和操作系统直接交互，因此编写设备驱动程序时需要特别小心。设备驱动程序中的缺陷很容易导致计算机无法响应。

示例：

当用 Microsoft Word 打开一个文件时，Microsoft Word 向操作系统发出一个命令，要求从指定路径得到该文件，随后操作系统通过设备驱动程序与硬盘通信，以访问该文件。

1.1.2 软件的工作方式

程序员使用高级语言，按照指定的语法和关键字编写程序。

编写程序需要开发环境。一般来说，开发环境由编辑器、编译器和解释器组成。

要编写程序使计算机执行特定任务，就需要向计算机提供具有以下信息的指令集：

- 提供的输入 例如，在航班预定应用程序里指定航班日期、航班时间、舱位等级和目的地
- 期望的输出 例如，想要在航班预定应用程序里查看座位剩余的情况
- 需要的处理 例如，在航班预定应用程序里，所需完成的处理包括接收数据、检查座位是否有剩余并查看输出

1.1.3 程序的执行过程

(1) 二进制符号

只有在理解指令时计算机才能运行指令集、执行任务。

计算机只能理解以二进制码编写的指令，即由 0 和 1 组成的序列。因此所有存储在计算机上的信息都以 0 和 1 组成的二进制码的形式存在。

(2) 机器语言

计算机使用“开”状态和“关”状态的组合表示计算机中组件的状态：1 代表开状态，而 0 代表关状态。

每个计算机指令都对应了一个计算机应该执行的物理操作，这些操作也是用二进制码表示

的。用 0 和 1 的序列表示的语言是计算机内部语言，又称为机器语言。例如，某种计算机的机器语言可能包括如表 1-1 所描述的指令集。

表 1-1

命 令	机 器 语 言
ADD	00000001
SUBTRACT	00000010
MULTIPLY	00000100
DIVIDE	00001000
READ FROM KEYBOARD	00010000
WRITE ON SCREEN	00100000
WRITE ON PRINTER	01000000

(3) 中央处理器

中央处理器也称为 CPU，是计算机的处理引擎。中央处理器执行用户提供的指令集。根据这些指令集，中央处理器使用内部的算术逻辑单元来执行 3 种基本操作：

- 数学运算，比如加、减、乘、除
- 把数据从一个内存地址移动到另一个内存地址
- 判断条件并根据判断结果转向一个新的指令集

(4) 编程语言

因为 0 和 1 形式的指令序列不容易记忆，程序员一般不使用机器语言编写程序。他们使用高级编程语言编写程序。高级编程语言是以简单的人类语言表达的指令集。因此，当需要在显示器上显示输出时，使用 WRITE ON SCREEN 命令，而不是 00100000。

可以使用的高级编程语言很多，包括 Microsoft Visual C++®、Visual Basic® .NET 和 Microsoft Visual C#™ 等。每种语言都有独特的优势。必须根据试图解决的问题来判断应该使用哪种编程语言。

与其他语言类似，编程语言拥有自己的词汇表。词汇表是具有特定意义的词的集合。编程语言词汇表里的词称做关键字，编程语言的文法规则称为语法。编写程序时遵循某一编程语言的关键字和语法，可以确保使用该语言编写的程序指令能被正确地解释和执行。

解决问题时可以使用各种编程语言编写计算机程序。但一般情况下，根据任务的复杂程度和一些其他因素，程序员会选择一种合适的编程语言编写程序。

示例：

目前一些流行的编程语言包括 Java、C、C++、Visual Basic、JavaScript 和 Word Basic。其中，JavaScript 和 Word Basic 比较简单，常用来编写简单的脚本和宏完成简单的任务，比如映射网络驱动器或格式化一个 Word 文档。另一方面，由于 C 语言常被用来编写操作系统软件，

因此 C 被认为是最复杂的语言之一。

(5) 编辑器、编译器和解释器

编辑器是用来编写程序的工具。编辑器可以是类似 Microsoft Notepad 的简单文本编辑应用程序，也可以是类似 Microsoft Visual Studio® .NET 的复杂程序。

编译器负责把代码从编写程序的高级编程语言转换成机器语言。这个过程可以帮助计算机理解使用高级语言编写的指令。用编译语言编写的程序必须先编译才可以运行。

常见的编译语言有 Fortran、COBOL、C 和 C++。

编译执行的语言在执行之前要直接转换为特定 CPU 的指令集，生成二进制代码。因此编译语言具有较高的 CPU 指令集依赖性和操作系统依赖性。但编译后的程序执行效率要比解释语言高得多。

解释器一边转换高级编程语言所编写的程序，一边运行该程序。由于解释语言不需要直接同机器码相关，所以实现起来较为简单，但程序执行的速度比编译语言低。

常见的解释语言有 Basic、APL、Perl 和 Python。一些流行的解释语言如 Perl 和 Python 在多种操作系统上都可以使用。

微软 .NET 中使用了新的程序执行技术，结合了编译执行和解释执行的优点。.NET 程序被编译器编译成为微软中间语言 (MSIL)。与一般编译语言不同的地方在于 .NET 编译器所产生的 MSIL 并不是本地代码，不能直接在任何现存的 CPU 上执行。.NET 程序启动后，在执行的过程中由 MSIL 编译器将 MSIL 编译成为本地代码，再调入 CPU 中执行。虽然每次执行程序的过程中都要经过 MSIL 编译器对 MSIL 代码进行编译，但是由于 MSIL 语句相对简单，和机器码非常接近，因此编译速度远比一般编译语言快。在程序执行时 CPU 中运行的是本地代码，因此也比解释执行的语言快得多。由于结合了编译语言和解释语言的特点，.NET 程序比解释语言的执行速度更快，接近编译语言。

1.2 计算机程序执行的阶段

可以把计算机执行的所有操作分成 3 个不同的阶段：

- 输入
- 处理
- 输出

无论如何使用计算机，所有计算机都按照相似的过程工作：一个人把数据输入计算机，计算机处理该数据，然后计算机产生输出。该输出在显示器上显示或在纸上打印。这样的过程被称为“输入—处理—输出”循环或 I—P—O 循环。

在本节中，将学习输入、处理和输出这 3 个阶段。

学习完本节后，将能够：

- 描述输入阶段如何工作
- 描述处理阶段如何工作
- 描述输出阶段如何工作

1.2.1 输入阶段如何工作

输入阶段是计算机程序执行的第一个阶段。在输入阶段中，需要向计算机提供数据。

示例：

要在航班票务柜台预定某个航班上的一个座位，乘客必须提供关于座位要求的信息，比如目的地、出发日期和时间及座位的类别等条件。负责预定航班的职员把该信息输入计算机。上述操作是航班预订程序的输入阶段。

1.2.2 处理阶段如何工作

程序执行的第二个阶段是处理阶段。在处理阶段中，计算机在输入阶段提供的数据上执行操作。

示例：

在航班预订程序的输入阶段完成后，计算机用输入的数据来判断在指定航班中是否有符合条件的座位。

1.2.3 输出阶段如何工作

程序执行的第三个阶段是输出阶段。在输出阶段中，计算机显示输入后经过处理的信息。

示例：

在航班预定程序的处理阶段处理输入阶段指定的数据以后，计算机在显示器上或在打印的文档中显示剩余座位的情况。

1.2.4 课堂练习 确定程序阶段

在本练习里，读者将能识别、确定程序执行的各个阶段。

在航班预定程序里，航班预定职员在_____阶段使用鼠标和键盘。在_____阶段，计算机使用 CPU 和内存来决定航班座位的剩余情况。在_____阶段，显示器显示座位空余情况。在_____阶段中，如有座位可打印出机票。

1.3 解决问题的技术

所有的程序都是为解决现实生活中的问题而开发的。要获得一个问题的解决方案，就需要解决问题的技术。

学习完本节后，将能够：

- 确定解决问题的步骤
- 描述解决问题过程中伪代码的用途
- 描述解决问题过程中流程图的用途

1.3.1 解决问题的方法

为了解决一个问题，程序需要执行一系列的步骤。该步骤序列称为算法。有两种常用方式

来表示算法：

- 伪代码
- 流程图

在开始编写程序之前，必须根据一系列的逻辑步骤来分析所要解决的问题。这些步骤包括：

(1) 识别输出和输入要求

要开发逻辑来解决问题，必须识别要求的输出。再根据输出要求，识别必要的输入。

(2) 确定相关的处理方法

必须确定输入—输出过程中需要执行的任务。要做到这一点，可以把问题分解成任务，并把任务安排在“输入—处理—输出”循环序列里。需要识别所有有条件的任务，这些任务的执行结果将取决于对条件判断的结果。

(3) 实现逻辑

在识别输出和输入要求以及相应的处理方法以后，可以用算法来表示程序逻辑。

(4) 验证逻辑

可以用简单的输入值来检查程序逻辑，从而确保该逻辑是正确的。另外，可以检测代码里的任何错误并在实际编写程序前更正它们。无论什么时候检查程序的逻辑，必须同时使用有效的和无效的数据以确保该逻辑在两种输入条件下都能工作。

示例：

Contoso 有限公司根据雇员在公司工作的持续时间给予奖励。在公司工作时间少于 3 年的雇员根据方案 A 给予奖励，其他的雇员都根据方案 B 给予奖励。

表 1-2 是示例中所描述问题的解决步骤。

表 1-2

步 骤	结 果
识别输出和输入要求	输出： 1) 雇员编码 2) 雇员姓名 3) 适用的奖励方案 输入： 1) 雇员编码 2) 雇员姓名 3) 服务年数
确定相关的处理	处理： 1) 提示接收雇员详细资料 2) 接收雇员详细资料 3) 查看雇员服务的年限 4) 根据服务的年数显示奖励方案和雇员详细资料