**第5章 计算机程序设计**

**第一部分 阅读和翻译**

**A部分 移动应用开发**

开发者和技术管理者应该评估多种创建移动应用程序的可行方法，并从技术和商业两个方面进行考虑。如果是为许多种不同的移动操作系统开发的高性能的本地应用程序，那么他们是否应该为此投入时间和金钱呢？他们会创建更好的手机网页应用，并且使用标准的Web技术编写，从而使其在手机浏览器上运行么？ 还是他们应该使用混合式的方法，简化开发过程，使其降低开发成本，从而使不同平台消费能力不一致的用户都能够体验应用？

**本地应用程序（原生应用）**

本地应用程序是为了让移动设备能够运行一个特定的手机操作系统而编程的。 如谷歌的Android，苹果的iOS，或是微软的Windows Phone。他们能提供快速的性能，并且可以存储数据以供离线使用。本地应用程序可以访问所有设备的内容，包括它的联系人、日历和照片，并且可以与它的硬件交互，包括麦克风、摄像头、测量移动和动作的加速计。例如，本地的Instagram应用程序可以访问设备相机进行拍照（见图5-1）。

图5-1本地的Instagram应用程序可以访问设备相机进行拍照

在没有明显修改的情况下，针对一个特定的移动平台或设备开发的应用程序一般将无法在另一个上正常工作。

创建本地应用程序需要编程语言、演示技术，以及针对每个平台的开发工具。

在测试它确保可以正常工作后，开发商可以将本地应用程序部署或者提交到某个应用程序商店进行审批和销售。当部署本地应用程序到Google Play、苹果的App Store或Windows Store时，开发者必须确保他们的应用程序遵循他们的出版商提出的规则和条件。例如，应用程序必须正常运行，并且可能不包含冒犯性内容，当请求当前位置或访问存储在设备上的信息时，该应用程序应该通知用户。

开发商为某个应用程序商店中发布的应用程序支付年费。商店会将销售出去的应用程序的一定百分比作为佣金。

**手机网页应用程序**

手机网页应用程序实际上是提供类似本地应用程序用户体验的网站。开发人员使用标准的Web技术来编程，其中包括HTML5、CSS和JavaScript。手机网页应用程序不部署到应用程序商店;相反，它们部署到一个Web服务器上，用户们可以在手机浏览器上访问它们。因此，用户总是可以使用最新版本的应用程序。创建该应用程序的网站的快捷方式并将其保存为一个设备的主屏幕上的图标或平铺就可以轻松访问手机网页应用程序。许多移动网页应用程序有一个响应式网页设计，使它们能够合适地在不同尺寸的屏幕上显示。有些公司选择开发手机网页应用程序，这样他们就可以编写一个不受应用商店规则限制的应用程序，从而可以在所有设备上使用。手机网页应用程序可以访问一组有限的设备功能，如基本的手势，离线工作，点击通话和GPS，但无法访问本机的功能，如摄像头、麦克风、加速计和设备通知。例如，在浏览器中运行的Instagram手机网页应用程序，它只可以显示照片，但不允许用户使用设备的相机来拍摄照片。

**混合型应用程序**

混合型应用程序结合了本地和移动网页应用程序的特点。像本地应用程序，混合型应用程序针对特定平台开发和并且可以部署到应用程序商店中。它们可以访问许多设备的硬件，例如它的照相机。又如手机网页应用程序，它们是由HTML5、CSS和JavaScript编写而成。开发人员使用开发工具将代码浏览器打包并做好准备，从而将程序部署到流行的应用商店中去。通过这种方式，混合型应用程序是跨平台的，这意味着相同的代码可以在许多移动平台上运行。这种方法往往可以节省开发时间和成本，但可能无法做到在所有设备上提供一致的用户体验或快速的性能。

**B部分 编程的未来**

随着这些趋势（面向方面编程、可视化编程、组件软件，分布式Web应用程序、云计算）蓄势待发，对于编程的未来我们能谈些什么呢？目前还不清楚编程语言在未来将会变成什么样子，但三种趋势似乎有可能。

* 编程语言将继续向自然语言（如英语）的方向发展。今天的编程语言，即使是其中最好的，也是十分捉襟见肘的，不是十分智能的。即使我们没有指定每一个细节，未来的编程工具应该也能够明白我们的意思。当我们考虑人工智能时，我们能够处理问题并且确保计算机用自然语言进行沟通。
* 程序员和用户之间的界限可能会更加朦胧。编程会变得更容易，有充分的理由相信，计算机用户通过工具，从而使他们能够在尚未掌握一门技术语言的复杂性的情况下编写应用程序。
* 计算机将在编程本身发挥日益重要的作用。今天的可视化编程环境可以创建响应于用户点击和命令的程序。未来的编程工具也许能写出整个程序，即使用户提供的只是问题的一个说明。将来，我们可能会看到计算机在无需人工干预下就可以预测问题并且给解决方案编程！

这三个趋势组成了一些现代编程的先驱工作。在20世纪70年代，查尔斯•西蒙尼在Xerox PARC[1]开发了第一个WYSIWYG文字处理软件，并作为微软公司的首席架构师，继续尝试着Word、Excel和其他产品的开发。 在2002年，他离开微软，并创立了致力于创造工具的新公司，这将使得日常计算机用户可以编写复杂的软件程序。其目标是使代码看起来像是设计，即通过简单地创建设计，用户就可以编写软件。据西蒙尼说，“软件应尽可能地像编辑一个PowerPoint演示文稿一样简单。”

西蒙尼设想一个名为意图编程的程序，它允许程序员专注于自己的方案，而不是编码的技术细节（见图5-2）。程序员可能是在卫生保健和海洋学或营销方面的专家，所以也许能够使用建模语言来描述一个设计来解决问题;软件生成器然后会在设计说明的基础上自动的写出实际代码。对于一些用户来说，建模语言可能类似于一个PowerPoint调色板上的用户界面;在他人眼中，它可能会有一个更数学化的外观。

另一个软件界的先驱者，IBM公司的格雷迪•布奇，正在试图针对建模语言将安全功能集成到软件建模工具中，并且构建可以建立程序的完整系统的建模工具成为可能。

无论发生什么事情，一件事似乎是可能的：未来的编程工具与今天的语言相比可能会差别很大。当计算机历史学家回顾历史时，他们会惊叹于我们命令计算机执行任务——即使是最简单的操作，当时对于我们来说都是那么的困难。西蒙尼对未来的软件业持乐观态度：“看看硬件人员针对摩尔定律都克服了什么，现在马上就是软件的时代了。”

在他《What Will Be》一书中，计算机科学家迈克尔•德图佐斯推测将来的方向：“到二十一世纪之末，这一预想很可能会得以实现：真正的编程新方式将会成为每个人的职责，就好比人们喜欢写作一样。曾经是古代文士的职责最终会随处可见。”

图5-2 查尔斯•西蒙尼创建了“国际软件”机构，以改变人们创建的方式

**第二部分 模拟写作:概要**

**一、简介**

概要是演讲材料的一个大体计划。概要描述了各种主题的命令，它们之间的相对重要性和各部分之间的关系。

**二、概要的内容顺序**

概要设计有多种方式。有时按年代安排很好运作；有时候，空间布局最适合。概要的共同特点是从大体到具体的转化。这意味着您先有一个笼统的想法然后得到所支持的具体例子。

**三、概述句的论点声明**

所有的概要都应该从一开始便声明论点或总领下文。这个表达论点的句子提出文章的中央想法。它具体完整而又简要扼要，可以表达出您的论点。

**四、概要的类型**

概要的两个主要类型是标题型概要和句子型概要。在标题型概要中，标题用单词或简要的词组给出。在句子行概要中，所有标题用完全句子表述。

**五、格式**

一个好的切入点是写好一篇文章的重要条件。 概要的格式通常跟随着：

**题目/标题**

**论题：**概述句

I. 论文介绍

A.总领文章的若干句子

B.结束语与你的论点声明(例如有3个要点)

1.第一要点

2.第二要点

3.第三要点

II. 正文——(应该有至少三段)

A. 第一论点——(正文第1段)

1.支持论点的引文或实证

2. 有至少五个句子

B.第二个要点——(正文第2段)

1.支持论点的引文或实证

2. 有至少五个句子

C.第三要点——(正文第3段)

1.支持论点的引文或实证

2. 有至少五个句子

III. 结论

A.总结文章的若干句子

B.再次声明你的论点和论据

当你将题目和信息照着概要完成后，剩下的就很容易了!检查拼写和标点，然后就可以了。

**六、范例**

A.标题型概要

读大学以及大学之后的选择

**论题：我如何选择大学的课程取决于我今后想从事的工作。**

1. 两种抉择描述如下

A.艺术历史或者化学

1. 专业考虑。
2. 个人考虑。

B.第三年的法语学习

1. 知道外语实用的优点。
2. 知识优势。
3. 必要性的问题。
4. 问题的定义

A.职业生涯的抉择。

B.关于要过怎样一种生活的抉择。

1. 解决问题的权宜之计

A.保留职业的可能性：化学。

B.利用已经获取到的文化知识：法语。

B.句子型概要

读大学以及大学之后的选择

**论题：我如何选择大学的课程取决于我今后想从事的工作。**

1. 关于不久之后选择大学课程我需要做的两个抉择

A.第一个抉择是选修艺术历史课还是化学课。

1. 我曾经计划成为一名专业的化学工程师。
2. 另一方面，我喜欢艺术，并计划旅行多了解艺术。

B.第二个抉择是是否继续第三年的法语学习，这己经超过了大学的基本要求。

1. 法语也许在工程或者旅游方面将会有帮助。
2. 另外，我很想阅读一些不错的法语书籍。
3. 鉴于我也许上其他的课程，这些考虑是否必要？
4. 我的问题往大了说是关于我未来人生的两难抉择。

A.一方面，我想在某个有利可图的职业上做到训练有素。

B.另一方面，我想过一种生活，具有与赚取金钱无关的有能力的价值。

1. 我需要对平衡上述相互冲突的需求做出抉择。

A.我会选择化学来保留职业的可能性。

B.通过选择法语，我会改进和加强我己在另外一种语言获取的文化修养。

**第三部分 听力和对话**

**对话：认识Java运行时环境（JRE）和Java虚拟机（JVM）**

|  |
| --- |
| *（在第一课的Java编程学习中，Mark下载一个简单的程序进行示例学习，但他发现，它无法正常运行。）*  **Mark**：对不起，**Henry**和**Sophie**。你能帮助我吗？  **Henry**：当然可以。什么问题？  **Mark**：为什么这个程序不能运行？它的源代码是正确的。  **Sophie**：是否您的计算机安装[1]Java运行环境？  **Mark**：尚未。什么是Java运行环境？  **Henry**：[2]JRE简单来说，是一个由Sun开发的软件平台，可以让计算机运行由Java编写的程序和应用程序。  **Sophie**：它包含Java虚拟机 Java库和其他一些部件。  **Mark**：什么是Java虚拟机？  **Sophie**：[3]Java虚拟机，Java虚拟机简单来说，是一套计算机软件程序和数据结构，使用一个虚拟机模拟用于执行其他计算机程序和脚本。他可以隐藏细节，忽视计算机硬件环境。  **Mark**：[4]JVM和JRE是什么关系？  **Henry**：JVM，是JRE的实例，当Java程序被执行开始起作用。当执行完成后，有垃圾回收机制。JVM一起发行了一套在标准类库其中执行的Java API。虚拟机和API必须符合对方，因此与Java运行时环境是捆绑在一起的。  **Sophie**：因此，这可以被视为一个虚拟的计算机中，虚拟机是处理器和API的用户界面。  **Henry**：JVM是Java平台一个至关重要的组成部分。使用相同的字节码的所有平台让Java成为“编译一次，到处运行”。  **Mark**：字节码吗？  **Henry**：是的。[5]这事JVM上运行的一种形式，中间语言的计算机通常被称为Java的字节码，这是正常，但不一定产生的Java源代码。如果打算在JVM上运行，必须编入标准的便携式二进制格式。  **Mark**：这是明显的，我们还要一个自动临时代码？  **Sophie**：这通常是在类文件上。  **Mark**：但是，一个大项目可能包括许多不同类的文件。  **Sophie**：是的。为便于分配的大项目，多级的文件可能被打包一起。.jar文件，和短期的Java归档文件。  **Henry**：[6]将JVM运行时执行的.jar文件，仿效的JVM指令集的解释，或者使用JIT编译，缩短形式及时编译器，如Sun的热点。  **Sophie**：JIT？我之前听说过一点点。什么技术不及时使用？  **Henry：**JIT汇编部分的字节代码具有类似的功能，从而减少了所需的时间，以便汇编。这是用在最JVMs今天以实现更大的速度。  **Mark**：这么多我感兴趣的知识！我会尽我所能了解。非常感谢！ |

**练习**

结对工作，并用以下材料进行一个类似的谈话情景：

**认识.NET框架和通用语言运行库（CLR）**

[1] Microsoft .NET框架

[2] .NET Framework是一个软件技术，适用于一些微软Windows操作系统。.NET Framework是一个关键的微软产品，并打算使用新的应用程序创建的大多数的Windows平台。它包括一个大型图书馆的预编码解决共同的规划问题和虚拟机器管理的执行情况编写的程序专门框架。

[3]通用语言运行库（ CLR ）是微软. NET倡议的虚拟机组成部分，在CLR提供的出现一个虚拟机的应用，使程序员不需要考虑这一问题，具体的CPU将执行该计划。

[4] CLR和.NET Framework

[5] 在CLR中运行的.NET Framework程序。类库和在CLR共同组成了. NET Framework 。

[6]在CLR运行某种形式的字节码被称为通用中间语言（印度煤炭公司），其中定义了一个执行环境的程序代码。

[7] 无论是准时或在执行使ngen.exe公用程式,.NET平台的字节码总是编译执行之前。在编译的时候，一个.NET编译器的源代码转换为印度煤炭公司的代码。在运行时，在CLR的及时编译器转换成代码印度煤炭公司原代码的操作系统。另外，印度煤炭公司代码可以编译为本地代码在一个单独的步骤之前运行。这加快了一切后的软件运行，使得印度煤炭公司到本地汇编不再是必要的。

**听力理解：集成开发环境(IDE)**

集成开发环境是一个程序开发环境集成的软件应用，它给计算机程序设计者的开发提供一个开发环境，它通常包括源代码编辑、编译器或解释器、自动补全代码工具和调试器。

集成开发环境的目标是提取必要的配置拼合成一个统一的整体，它可以减少学习一门语言的时间，从而增加开发商的效率。 例如，代码完成后可以通过编译，立即反馈其中的句法错误。

控制台和终端的发展使得开发环境开始变得必要起来。 早期语言没有一个统一的开发环境，它们在提交给编译器之前使用流程图、程序格式化和键盘穿孔机进行操作。 [Dartmouth BASIC](http://en.wikipedia.org/wiki/Dartmouth_BASIC)是第一种使用集成开发环境的语言。 它的集成环境是命令行的，和今天的菜单式图解的环境并不一样。 然而它完整的集成了编辑、文件管理、编辑、调试是典型的现代集成开发环境。

Microsoft Visual Studio, Delphi, JBuilder, [Eclipse](http://en.wikipedia.org/wiki/Eclipse_(software)), 和[NetBeans](http://en.wikipedia.org/wiki/NetBeans) 是当今著名的集成开发环境。

**听写：艾达·拜伦，第一位程序员**

艾达·拜伦·罗夫雷斯夫人，是计算机史的最生动的角色之一。作为浪漫主义诗人拜伦的女儿，艾达·罗夫雷斯夫人被称作数字的“美女”。她是第一个计算机的发明者查尔斯·巴贝奇的合作者。

在1834年11月的一个晚餐会上，艾达听到了巴贝奇的关于新的计算引擎——分析引擎的想法。他设想：一个计算引擎不仅是可以预见的，再不久的将来完全可以实现。与其他人不同，艾达被他的想法深深吸引了。

艾达通过巴贝奇的引擎上翻译了一份梅纳布雷亚将军的文件，后者后来出任了意大利总理。当她向巴贝奇显示了她的翻译后他建议她增加她自己的笔记，结果是三倍的原始文章的长度。在巴贝奇和艾达之间的信件中充满事实和幻想。在她1843年发表的文章上，艾达期望计算机软件在人工智能和计算机音乐方面的发展。

艾达·罗夫雷斯构想了使用打孔卡方法计算柏努利数字，成为第一个计算机程序设计者。美国国防部在1980年以她的名字命名了计算机语言“Ada”。