

Sun 认证 JAVA2 系列丛书

SCJD

(Sun Certified Java Developer)

试题详析大全



JAVA2

考试号: 310-027



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

URL: <http://www.phei.com.cn>

上海东方激光照排有限责任公司

出版发行

制 作

考试号: 310-027

Sun 认证 Java 2 系列丛书

SCJD

Sun Certified Java Developer

试题详析大全

策 划 何学仪

主 编 钟 珞

副主编 饶文碧 潘 昊

编 著 钟 珞 饶文碧

潘 昊 宋华珠

吴彦春 陈采军

徐 锐 陈旭辉

电子工业出版社

考试号:310-027

Sun 认证 Java 2 系列丛书

SCJD

Sun Certified Java Developer

试题详析大全

(光盘附书)

策 划 何学仪

编 著 钟珞

责任编辑 苏宁萍 陈 艳

出版发行 电子工业出版社 电子出版部

经 销 各地新华书店

开 本 1092 × 787mm 1/16

印 张 18.75

标准盘号 ISBN 7-900074-83-X/TP·65

前言

Sun MicroSystem 公司的总裁 Scott McNealy 认为 Java 为 Internet 和 WWW 开辟了一个崭新的时代。Java 以其独有的开放性、跨平台性和面向网络的交互性席卷全球,以其安全性、易用性和开发周期短的特点,迅速发展成为当今全球的第二软件开发平台。Java 公用规范(PAS)已被国际标准化组织(ISO)列为当今世界信息技术三大要点之一。

Sun Java 认证是业界唯一经 Sun 授权的 Java 认证。根据认证,在企业方面可以藉此作为招聘人才的评判标准,或是作为衡量员工技术水准的依据;在技术方面,通过这些认证也可以证明个人的技术能力。早日掌握 Java 技术,对每个有志于在 IT 行业发展的人来说是尤为重要的。

书中以 Java 开发员考试目标为依据,按照基本概念、核心内容、试题分析、本章小结和习题几个环节来组织全书内容,介绍了 Sun Java 2 开发员认证及如何准备认证考试、获得高分的策略,并给出了程序实例。全书共分十一章,主要包括 Java 软件开发的方案及步骤、网络数据库的创建及管理、数据库查询 mSQL 及访问数据库接口 JDBC、两层和多层数据库的体系结构和用户界面 UI 的创建。

本书可以作为 Sun Java 2 开发员认证的教程,也可用于大专院校、科研院所的本科生和研究生进行 Java 程序开发的参考手册。阅读本书要求读者通过 SCJP 认证或已经掌握相当于 SCJP 的 Java 编程知识,具有一定的面向对象技术的经验,懂得用户界面(UI)程序设计和 TCP/IP 协议,并具有在 Microsoft Windows 9x/NT/2000 及 Solaris Open Windows 平台上开发程序的经验和 socket/stream 的编程经验。

本书由何学仪策划,武汉理工大学钟珞教授任主编,饶文碧、潘昊副教授为副主编。宋华珠、吴彦春、陈采军、徐锐、陈旭辉等同志参加了全书的编写工作。

希望本书能够给读者以帮助,同时也希望读者能对书中的缺点和不足加以批评指正。

目 录

第一章 Java 2 开发人员认证考试概要

1.1 基本概念	1
1.2 核心内容	4
1.2.1 考试的形式	4
1.2.2 考试过程与范围	6
1.2.3 通过考试方法概述	7
1.3 本章小结	7
1.4 习题	8

第二章 软件开发方法

2.1 基本概念	9
2.2 核心内容	14
2.2.1 软件开发生命周期	14
2.2.2 面向对象软件开发	25
2.2.3 开发 Java 程序	52
2.3 试题分析	56
2.4 本章小结	84
2.5 习题	85

第三章 数据库查询

3.1 基本概念	86
3.2 核心内容	92
3.2.1 关系模型	92
3.2.2 数据库设计	93
3.2.3 关系的规范化理论	95
3.2.4 结构化查询语言 SQL	97
3.2.5 mSQL	97
3.3 试题分析	100
3.4 本章小结	105
3.5 习题	105

第四章 Java 与数据库的连结(JDBC 技术)

4.1 基本概念	106
4.2 核心内容	111
4.2.1 数据库建立连接	111
4.2.2 执行查询语句	111
4.2.3 数据库操作	113
4.2.4 参数的输入和输出	114
4.2.5 动态数据库访问	115
4.2.6 在数据库存取的二层与三层模型上的应用	117

4.3 试题分析	118
4.4 本章小结	132
4.5 习题	132
第五章 Java 网络编程	
5.1 基本概念	133
5.2 核心内容	145
5.2.1 一对一的 Socket C/S 通信	145
5.2.2 TCP 协议通讯的服务方实现	145
5.2.3 TCP 协议通讯的客户方实现	147
5.2.4 一对多的 Socket C/S 通信	149
5.2.5 一对多通信的客户方实现	150
5.2.6 一对多通信的服务方实现	150
5.2.7 UDP 协议通信的服务方实现	150
5.2.8 UDP 协议通信的客户方实现	152
5.3 试题分析	153
5.4 本章小结	163
5.5 习题	163
第六章 GUI 布局管理器	
6.1 基本概念	164
6.2 核心内容	165
6.2.1 GUI 类	165
6.2.2 用 AWT 组件构造用户接口	167
6.2.3 布局管理器	168
6.3 试题分析	171
6.4 本章小结	182
6.5 习题	182
第七章 Swing GUI 组件	
7.1 基本概念	183
7.2 核心内容	186
7.2.1 GUI 设计原则	186
7.2.2 类与面向对象设计	187
7.2.3 创建 GUI	188
7.2.4 模型—视图—控制器结构	200
7.3 试题分析	202
7.4 本章小结	205
7.5 习题	205
第八章 事件处理	
8.1 基本概念	206
8.2 核心内容	206
8.2.1 基本事件处理过程	206

目 录

8.2.2 事件适配器	210
8.3 试题分析	213
8.4 本章小结	219
8.5 习题	219
第九章 Java 程序设计应用实例	
9.1 MSG 实例的快速原型	220
9.2 建立 BrokerTool 的数据库	234
9.3 使用 JDBC API 开发的一个 Java 应用程序 ——客户管理系统	238
第十章 Java 2 开发人员认证考试高分策略	
10.1 编程任务高分策略	278
10.2 更好地为上传编程任务做准备	279
10.3 简答考试高分策略	282
10.4 本章小结	283
10.5 习题	283
第十一章 Java 2 开发人员认证考试示例	
11.1 编程任务示例	284
11.2 简答考试示例	286
附录:Sun Java 2 开发人员认证目标	290
参考文献	292

第一章 Java 2 开发人员认证考试概要

1.1 基本概念

一、Java 特性

Java 语言是一种网络编程语言,它最大限度地利用了网络资源,可以跨平台、跨操作系统、跨网络运行。另外,由于 Applet(Java 小应用程序)易于在网络上快速地下载和发送,且具有不需要修改应用程序就可以增加 Web 页新功能的特性,因此在 Internet/Intranet 中得到广泛应用。此外,Java 还配有丰富的类库,为用户编程提供了极大方便。

Java 语言的主要特点如下:

1. 安全性

安全性可能是 Internet 开发者面临的最重要的问题。用户最怕两件事:保密信息被窃取或计算机系统被黑客破坏。Java 的内置/内置式安全性解决了这两个问题。

Java 的安全性模型有三个基本组成部分:类装入器、字节代码验证器和安全管理器。字节代码验证器确保 Java 程序的编译正确,遵循 VM 的访问限制,字节代码不会访问不可访问的保密数据。下一层安全性是由类装入器提供的。类装入器从网络上获取类时,将把来自不同服务器的类相互分开并与本地类区别开来。通过这样分离,类装入器就可以防止网上装入的类假扮成标准的内置/内置式类,或干扰从其它服务器装入类的操作。安全管理器实现 VM 的安全性策略。安全性策略确定 VM 可以进行什么活动,在什么条件下进行。安全管理器通常用于防止警告性错误。然而,安全管理器本身是用 Java 编写的,所以必要时可以覆盖。当然,可以采取措防止敌意的 Java 程序写入自己的安全策略,这些措施可防止 Web 浏览器运行时将其它安全管理器加入系统。

在专用网络中,需要文件访问和任意网络访问来解决业务要求,应用程序开发者可以对网络上的每个 PC 机改变安全管理器源代码并重新编译到 Web 浏览器中,定制专用网络的安全性策略。如果专用网络有向 Internet 之类的公用网络的管理类,则定制安全管理器的编码更加困难。覆盖安全管理器时必须十分小心,否则公用网上的敌对小程序可能会利用松散的内部安全性策略进行袭击。

Java 1.1 推出后,开发人员能在小程序中加入另一层安全性。核心 API 中加进了扩展安全性 API,使 Java 类可以签名,用户可以用这个签名确定装入的是否为安全的程序。

总之,Java 的内部安全性措施保证 Java 程序在 VM 规则下操作,防止未授权的程序访问包含专有信息的系统资源或危及客户机的完整性。Java 还使开发人员可以给所生成的程序签名,使用户能确保访问可靠的小程序和应用程序。

2. 核心 API

API 即应用程序编程接口,它是应用程序开发者为了更快地构造软件而使用的功能库。

API 分成多个软件包,它们是分别完成相关功能的类。其中一个软件包包含一些核心语言功能,如文本处理和错误处理,没有这个库,Java 程序的编写几乎是不可能的。其它软件包中包含实用程序、联网、I/O、图形用户接口工具和与 Web 浏览器的接口。还有涉及安

全性、数据库访问、可复用软件组件、集合、拖放、访问性和引用对象的软件包,新的软件包也在不断推出。

3. 开放式标准

Java 跨平台能力的最动人之处在于 Java 类文件不需要预先针对每个平台进行编译,同一个编译过的 Java 程序可以用于运行 Java VM 的 PC、Macintosh 或各种系统平台。另一个成为成功的跨平台开发工具的关键是每个平台上有专用的核心功能集。核心 Java API 对所有实施都是一样的,它足够复杂,无需对桌面应用程序编写本地码。

除了公用桌面操作系统外,我们还希望 Java VM 实现在用于嵌入系统和智能卡的芯片上。Sun 已经推出 CPU 芯片 picoJava,提供个人家电、智能电话和网络计算机等设备中直接执行的 Java 字节代码,并宣布计划制造另外两种 Java 处理器——microJava 和 UltraJava。

4. 分布式和动态结构

Windows 操作系统中,部分程序可以换成动态链接库(DLL),以便动态共享或装入,操作系统在执行时完成最后阶段的链接。使用共享 DLL 能节省内存并改进软件的模块化程度。在 Unix 中,这类动态链接是通过共享库完成的。

Java 进一步利用了动态库。VM 类装入器在从盘上取类文件的同时还可以从网络上获取,使 Java 应用程序既是动态的又是分布式的。这种特性使可用 Java 的浏览器能够自动采用新 Web 地址上可用的协议。

5. 面向对象

面向对象编程(OOP)是以可复用、可扩展和可维护的方式编写软件的方法。Java 是面向对象的语言,Java 核心 API 实际上就是一组预制的 OOP 构件,即类库,类库使编程人员进行新项目开发时有一个很好的开始。

6. 多线程

多线程应用程序可以同时独立执行多个线程,这些线程可以交流或合作,用户显示状态为单个程序。多线程常用于完成下列功能:

(1) 维护用户接口响应性:如果应用程序需要完成一个费时的任务,可以用多线程防止任务进行时用户接口变成不响应状态。

(2) 等待唤醒:让程序等待特定时间的最佳方式是将程序放进睡眠线程。

(3) 简单多任务:多线程可以很容易地在一个过程中运行多个任务。

(4) 建立多用户应用程序:多线程常用于建立服务器应用程序,服务器应用程序等待请求到达,然后建立与请求者的对话。

(5) 多进程:许多操作系统支持具有多个处理器的系统,但大多数这种系统无法把单个执行线程分成不同处理器上的多个执行线程。Java 可将一个应用程序分成不同的线程,以便更好地利用处理器能力。

这里所述的每个项目既适用于 Internet,也适用于嵌入系统的应用程序。Java 通过其类库的一部分实现多线程,但 Java 也有保护程序线程安全的语言结构,线程保险程序保证不同的线程不会意外地相互伤害。Java 的关键字 synchronized(同步)可以防止二个线程同时进入同一关键块。

7. 内存管理与无用内存的收集

在 C 和 C++ 中,编程人员有责任分配所需内存,并在任务完成后释放内存。如果内存分配与内存释放不完全相符,则程序会立即崩溃或消耗系统资源直到耗尽。

Java 使用无用内存收集程序(garbage collection)解决了这个问题。临时内存自动地被程序激活,随后马上自动放弃。为了提高性能,Java 的无用内存收集器在自己的低优先级线程中运行,提供了效率与实时响应间很好的平衡。

内存管理强化了 VM 的安全性,Java 不用严格意义上的指针,而是 VM 资源的引用,这种变量不允许任何算法,防止编程人员访问 VM 以外的系统资源。

二、Java 1.1 的新特性

Java 1.1 在 1997 年初推出,其新特性如下所示:

1. 国际化:加进新类和方法使编写国际化程序更方便。
2. 安全性:新的安全管理器中加进了几个改进,包括数字化签名。
3. 性能改进:新的 AWT 事件模型和完全改写的 AWT 自然码使 GUI 性能大大提高。
4. 网络与 I/O 改进:新类提出了更多网络功能和更大的定制。新的 Reader 和 Write 类提供了高性能、国际化的缓冲输入和输出。

5. 对象反射:一个特殊 API,可以得到特定对象加进的特许信息,这在调试和其它 VM 改进程序中特别有用。

6. JDBC:Java 数据库互联(JDBC)是链接 Java 程序与数据库的 API,遵循 JDBC 的数据库应用程序是不依赖于任何特定数据库厂商的。与 ODBC 相似,用一个特定厂商的驱动程序将 JDBC 应用程序链接到实现的数据库。一个 JDBC 与 ODBC 的接口于 1996 年由 Sun 发布,它使 JDBC 能够联接大量现有的 ODBC 驱动程序,这对于 Java 作为一个公司的客户机/服务器开发工具的工业标准很关键。

7. Java Beans: Java Beans 是建立可复用软件组件的强大 API。利用 Java Beans, Java 对象可以方便地由非编程人员连接起来,可以用 JBuilder 之类的连接工具完成。

三、Java 2 的新特性

Java 2 增加了更多的改进,有些是 Java 1.1 基础上的改进,有些是全新的。

1. 安全性、Java Beans、反射和性能的改进:Java 2 中许多 Java 软件包都作了改进。安全性结构加进了基于政策的访问控制,改进了权限管理。Java Beans 加进了拖放支持,而反射加进了越过安全性的功能,特别是在使用对象反射时。各种 Java 库的功能也有改进,例如内存分配加快、类装入时的内存使用减少,以及及时(JIT)编译器。

2. Java 基础类:Java 基础类(JFC)包括大量改进,现已支持下列辅助性技术:访问性 API、改进图形和图象的 Java 2-DAPI、Swing 中的一组新 GUI 组件以及 AWT 组件。

3. 集合:集合 API 使对象组更容易处理。在 Java 2 之前,主要使用原先的 Vector 与 Hashtable 类和 Enumeration 接口。现在可以处理平衡树、图形链表和简化数组排序等。

除了这些改变外,整个方法、类和软件包的改变提供了更多的功能。

四、Java 2 软件开发包(SDK)

为了编制 Java 2 程序,你需要 Java 2 SDK 标准版本的发行产品——JDK 1.2 及以上产品。

Java 2 的 Solaris 和 Windows 版可以从如下 Sun Microsystems 的 URL 免费获取:

http://java.sun.com/java_2

在该 URL 可以找到 Java 2 软件及其文档,建议将二者都下载。

JDK 1.2 包括下列实用程序:

1. javac: Java 编译器。
2. java: Java 翻译器。

3. appletviewer(小程序浏览器): 执行 HTML 文件上的 Java 小程序类的 Java 浏览器。
4. javadoc: 根据 Java 源代码及其说明语句生成 HTML 文档。
5. jdb: Java 调试器。
6. javah: 产生可调用的 Java 过程的 C 过程,或建立能被 Java 程序调用的 C 过程的文件头。
7. javap: Java 反汇编器。
8. rmic: 生成支持远程方法调用(RMI)的类文件。
9. rmiregistry: 用于访问特定机器上的 RMI 对象的注册表。
10. rmid: RMI 对象注册和激活的激活系统精灵。
11. Serialver: 序列化实用程序,允许标注持续对象的版本。
12. native2ascii: 在标准 Latin-1 Unicode 字符与其它国际编码系统间转换的特殊程序。
13. jar: Java 压缩文件(JAR 文件)生成器。
14. keytool: 用于安全密钥的生成和管理。
15. jarsigner: 实现 JAR 和类文件的数字签名。
16. policytool: 安全政策配置管理器。
17. tnameserv: Java IDL 临时名称服务器。

1.2 核心内容

1.2.1 考试的形式

Java 2 开发员认证考试包括两部分:(1) 编程任务;(2) 简答考试。与 Java 2 程序员认证考试不同的是:Java 2 开发员认证考试不再只是多项选择题的测试,因此这种认证考试需要花费更多的时间与精力来准备。要获得 Java 2 开发员认证考试的详细说明可浏览如下网页:<http://suned.sun.com/usa/cert-test.html>。

一、编程任务

编程任务是 Java 2 开发员认证考试的第一部分。应试者在 Sun 公司的培训服务中心登记并交纳费用后,Sun 公司便可将应试者的名字加入他们的认证考试数据库,这一过程需要一天时间。之后,应试者可得到由 Sun 公司传递来的编程任务,也可登录到该数据库下载编程任务,登录网页位于:<http://www.galton.com/~sun>。对于不同的应试者编程任务是不同的,它们包括:一个介绍和索引文件,一个需进行数据格式转换的数据文件,一个将在编程开始时被引用的源代码文件及任务需求。

应试者按照编程需求完成程序设计后,便可将其设计结果上传到 Sun 公司相应网站。考试中心的测试员将对应试者传来的程序进行功能测试,检查程序是否能正确地执行所有要求操作。如果应试者未能通过此阶段的测试,这将意味着整个考试的失败。

除了测试程序代码的正确性外,测试员还将按照评分准则对程序进行评分。评估分满分为 155 分,其评分准则如下所示:

1. 总体评估(共 72 分)

(1) 易操作,带说明文档,完成程序设计(15 分):要求整个程序易操作,且是用户友好的。要对难理解的源程序代码进行注解,同时也要避免过注释。

(2) 程序的一致性 & 程序的逻辑结构(20 分):要求整个程序具有一致性,能完全按照设计任务解决问题。要求使用恰当的代码技术,避免使用不必要的复杂算法。

(3) 程序代码的规范化及可读性(15分):要求程序具有常用的凹进书写风格,并尽量使用常规的、或有实际意义的命名。

(4) 程序设计思路的清晰性及可靠性(7分):要求能通过程序中的注释、标识符及算法读懂源代码程序,代码易维护。

(5) 异常处理功能(10分):要求利用标准的 Java 2 异常处理功能进行异常处理,程序应能在不中断程序运行情况下,向用户传递错误信息。

(6) 充分利用核心 Java 2 包(5分):要求尽量利用核心 Java API 的标准类。

2. 说明文档评估(共 10 分)

(1) 执行程序的使用文档:要求提供一个 README 文件,该文件将用来介绍程序的执行方法、标识符的定义情况及每个存在于目录路径中的程序文件。

(2) Java 2 源代码文档:要求在 Java 2 程序中使用 Javadoc 标记进行合理注释,并生成清晰、准确并具有一致性的软件文档。

(3) 代码中不清楚方面的注释:对于软件功能不清楚的代码段要给出能描述代码操作的注释。

(4) 避免过注释:对于软件功能清楚的代码段不要进行注释。

(5) 用户文档:至少要自己满意该文档。

3. 面向对象设计评估(共 6 分)

(1) 合理地使用成员变量、方法:要求变量为局部的成员变量,合理地选择方法、方法参数及方法返回类型。

(2) 合理的成员可访问性和范围:要求成员变量和方法使用可访问修饰符,以合理的范围进行局部变量的声明。

4. 用户接口评估(共 20 分)

(1) 布局合理:要求布局应满足要求。

(2) 按照人-机交互(HCI)原则进行布局:要求用户接口应使用标准的 GUI 原则进行设计,使用合适的 JDK 1.2 事件接口/适配器及合理的事件处理模型来实现。

5. 数据转换程序评估(共 10 分)

要求数据转换程序满足设计任务中要求,且是实用的。

6. 服务器设计评估(共 37 分)

(1) 类应以线程-安全方式使用(8分):要求在设计服务器及网络接口时应使用线程-安全编程方法。

(2) 正确地实现锁定(15分):在数据库中使用记录锁,并能正确实现。

(3) 错误处理(6分):要求使用合理的错误处理技术。

(4) 查寻算法,清晰并有效(8分):以直接的方式实现对标准、有效的查寻算法的使用。

由于上述评估是由测试员进行的,因此需要较长的等待评分结果时间,应试者一般在提交程序设计结果后 30 天左右才能得到全部的评分结果。

二、简答考试

Java 2 开发人员认证考试的第二部分是由一些(5-10个)简答题组成。与程序员认证考试不同的是开发人员认证考试第二部分的考试内容是应与应试者所完成的第一部分编程设计密切相关的,它至少包括三个方面的内容:

(1) 客观方面:这方面涉及 Java 特性及 Java 库的知识。如:你可能被要求列出一些用于

存储运行时间对象未知数的数据结构,然后说明每种结构各自的优点。

(2) 实践方面:这方面的测试重点是考察你對自己代码的理解,即检查程序是否你自己完成的。

(3) 主观方面:你将被要求论述你的选择。也许你未选择最快或最有效的数据结构,那么你为什么做出这样的选择?

下面给出了一个由 Sun 公司提供的简答题例子。

In various method calls in your assignment submission, you were forced to deal with possible exceptions that could be thrown due to a number of different circumstances.

- A. Describe (in not more than four sentences each) two approaches to indicate to the caller of a method that the method has failed its operation. One of these approaches should be the one you implemented in your assignment submission.
- B. Which of these approaches did you use in your assignment submission?
- C. Describe (in not more than two sentences each) a total of no more than six advantages and six disadvantages of these approaches.
- D. Describe the design goals you were guided by that led you to implement the approach you selected in your assignment submission.

正如大家所看到的那样,这些简答题不能靠猜测来完成,而是要认真、仔细地思考,并能正确、清晰、准确地回答问题。

1.2.2 考试过程与范围

一、考试的过程

Java 2 开发人员认证考试将按下述步骤来进行:

1. 进行编程任务考试的注册(\$250)。
2. 通过电子邮件或从 Sun 公司相应网站下载编程任务。
3. 完成编程任务。
4. 上传完成的任务到 Sun 培训服务认证数据库。
5. 进行简答考试的注册(\$150),考试号为:310-027。
6. 进行简答考试。
7. 四周之后,应试者得到分数。

二、考试的范围

Java 2 开发人员认证考试范围将覆盖 Java 软件开发的一个很广范围。为了完成编程任务并回答问题,应试者必须非常熟悉 Java 语言及其应用程序的开发方法,熟悉 Java API 及其程序设计技术,下面给出了由 Sun 公司所提供的考试范围:

1. TCP/IP 网络
2. I/O 流
3. RMI
4. 对象序列
5. JDBC
6. Swing
7. AWT 事件模式

8. Javadoc
9. 多线程
10. 错误及异常处理
11. 客户机 - 服务器设计
12. 接口设计

1.2.3 通过考试方法概述

参加 Java 2 开发人员认证考试的应试者应考虑以下几个问题,以决定是否为考试作好了准备:

1. 你在一周内用三天以上时间编写 Java 代码吗?
2. 说出设计有效用户接口的一种或两种原则。
3. 你能写出多线程应用程序吗?
4. 回忆你被安排处理未完成项目的最后一次经历,包括处理未注释代码及处理丢失源代码的经历。你是如何处理这些问题的?
5. 在内存存储数据时,如何决定使用数组、向量、散列表?
6. 使用 RMI (Remote Method Invocation, 远程方法调用)、对象序列的相对优点是什么?
7. 二层客户机-服务器方案与三层的有何区别?

在考虑上述问题后,如果应试者已经能写出中等以上大小的、基于 GUI 的、客户机-服务器设计的应用程序,并具备完备的 Java API 知识,那么便可进入 Java 2 开发人员认证考试的第一阶段一下载编程任务。

由于该认证考试最关键的一步是使设计代码能正确运行,且能满足所有任务指令,因此应试者在获得编程任务后,要认真阅读这些文件和任务指令,并将要点记录下来,以便按指令编程。

完成程序设计后,应试者应对程序进行测试,其方法如下所示:

1. 确认设计的程序能满足全部任务需求。
2. 确认没有进行多余的设计。
3. 自检测,改进代码。

之后,应试者便可上传完整的程序设计结果至 Sun 公司相应站点,并准备进入考试的第二部分一简答考试。

回答每个简答题时,应以清晰、完整、简捷为答题准则。另外以下几点将有助于应试者通过第二部分的考试:

1. 答案要求尽量简短,每句都为要点。避免使用长的、不切题的、进行时态的句子。
2. 确认答案易懂。
3. 不要进行不必要的解释。
4. 努力评价在程序设计时自己是如何协调编程任务与所选方法的。

在回答了全部问题之后,如果还有剩余时间,则要再次阅读每题,并检查每题的答案,以便为每题提供一个最恰当的答案。

1.3 本章小结

本章首先介绍了 Sun 公司 Java 2 开发人员认证考试的形式,包括:编程任务和简答考试两

部分,及编程任务考评的标准。然后介绍了 Java 2 开发员认证考试的过程与范围。最后简单地介绍了通过考试的方法。

1.4 习 题

1. Java 2 的新特性有什么?
2. Java 2 开发员认证考试包括哪些内容?
3. 试述 Java 2 开发员认证考试范围、过程。
4. 编程部分的考试评分准则有哪些?
5. 简答考试所涉及到的主要内容包括哪些?

第二章 软件开发方法

2.1 基本概念

一、软件工程学

软件工程学强调软件产品的生产特性,采用工程化的方法,围绕生产过程自动化、软件产品的可靠性展开对软件的生产方式、生产管理、产品设计方法、生产工具系统和产品质量保证等的研究,研究的重点是软件设计方法论及工程开发技术,并将计算机科学理论和工程方法相结合,重视对软件工程学有指导作用的理论,抽象和设计等重要概念的定义。

软件工程学是研究软件生命周期的一切活动,包括需求分析、计划、设计、实现、测试、维护 and 管理的科学。它研究的最终目的是为经济有效地开发软件系统提供科学的方法和工具。软件工程学既强调软件开发的工程特点,又强调软件设计方法的科学性、先进性。

本书以传统的和面向对象软件工程相结合,突出最新研究成果,使读者领会为什么面向对象方法优于传统的方法。

二、软件生命周期

软件工程采用的生命周期方法学就是从时间角度对软件开发和维护的复杂问题进行分解,把软件生命的漫长周期依次划分为若干个阶段,每个阶段有相对独立的任务,然后逐步完成每个阶段的任务。采用生命周期方法学开发软件的时候,从对任务的抽象逻辑分析开始,一个阶段一个阶段地进行开发。前一个阶段任务的完成是开始进行后一个阶段工作的前提和基础,而后一阶段任务的完成通常是使前一阶段提出的解法更进一步具体化,加进了更多的实现细节。每一个阶段的开始和结束都有严格标准,对于任何两个相邻的阶段而言,前一阶段的结束标准就是后一阶段的开始标准。在每一个阶段结束之前都必须进行正式严格的技术审查和管理复审,从技术和管理两方面对这个阶段的开发成果进行检查,通过之后这个阶段才算结束;如果检查通不过,则必须进行必要的返工,并且返工后还要再经过审查。审查的一条主要标准就是每个阶段都应该交出“最新式的”(即和所开发的软件完全一致的)高质量的文档资料,从而保证在软件开发工程结束时有一个完整准确的软件配置交付使用。文档是通信的工具,它们清楚准确地说明了到这个时候为止,关于该项工程已经知道了什么,同时确立了下一步工作的基础。此外,文档也起备忘录的作用,如果文档不完整,那么一定是某些工作忘记做了,在进入生命周期的下一阶段之前,必须补足这些遗漏的细节。在完成生命周期每个阶段的任务时,应该采用适合该阶段任务特点的系统化的技术方法——结构分析或结构设计技术。

把软件生命周期划分成若干个阶段,每个阶段的任务相对独立,而且比较简单,便于不同人员分工协作,从而降低了整个软件开发工程的困难程度;在软件生命周期的每个阶段都采用科学的管理技术和良好的技术方法,而且在每个阶段结束之前都从技术和管理两个角度进行严格的审查,合格之后才开始下一阶段的工作,这就使软件开发工程的全过程以一种有条不紊的方式进行,保证了软件的质量,特别是提高了软件的可维护性。总之,采用生命周期方法学可以大大提高软件开发的成功率,软件开发的生产率也能明显提高。

三、软件维护

1. 软件维护的定义

所谓软件维护就是在软件已经交付使用之后,为了改正错误或满足新的需要而修改软件的过程。我们可以通过描述软件交付使用后可能进行的四项活动,具体地定义软件维护。

因为软件测试不可能暴露出一个大型软件系统中所有潜藏的错误,所以必然会有第一项维护活动:在任何大型程序的使用期间,用户必然会发现程序错误,并且把他们遇到的问题报告给维护人员。我们把诊断和改正错误的过程称为改正性维护。

计算机科学技术领域的各个方面都在迅速进步,大约每过几个月就有新一代的硬件宣告出现,经常推出新操作系统或旧系统的修改版本,时常增加或修改外部设备和其他系统部件;另一方面,应用软件的使用寿命却很容易超过十年,远远长于最初开发这个软件时的运行环境的寿命。因此,适应性维护,也就是为了和变化了的环境适当地配合而进行的修改软件的活动,是既必要又经常的维护活动。

当一个软件系统顺利地运行时,常常出现第三项维护活动:在使用软件的过程中用户往往提出增加新功能或修改已有功能的建议,还可能提出一般性的改进意见。为了满足这类要求,需要进行完善性维护。这项维护活动通常占软件维护工作的大部分。

当为了改进未来的可维护性或可靠性,或为了给未来的改进奠定更好的基础而修改软件时,出现了第四项维护活动。这项维护活动通常称为预防性维护,目前这项维护活动相对说比较稀少。

从上述关于软件维护的定义不难看出,软件维护绝不仅限于纠正使用中发现的错误,事实上在全部维护活动中一半以上是完善性维护。国外的统计数字表明,完善性维护占全部维护活动的 50%—65%,改正性维护占 17—21%,适应性维护占 18%—25%,其他维护活动只占 4% 左右。

应该注意,上述四类维护活动都必须应用于整个软件配置,维护软件文档和维护软件的可执行代码是同样重要的。

2. 维护的特点

(1) 结构化维护与非结构化维护的对比

在图 2-1 中描绘了作为维护要求的结果可能发生的事件流。

如果软件配置的唯一成分是程序代码,那么维护活动从艰苦地评价程序代码开始,而且常常由于程序内部文档不足而使评价更困难。诸如软件结构、全程数据结构、系统接口、性能和(或)设计约束等微妙的特点是难于搞清的,而且常常误解了这一类特点。最终对程序代码所做的改动的后果是难于估量的:因为没有测试方面的文档,所以不可能进行回归测试(即为了保证所做的修改没有在以后可以正常使用的软件功能中引入故障而重复过去做过的测试)。可惜,我们正在进行非结构化维护,并且正在为此而付出代价(浪费精力和受挫折),这种维护方式是没有使用良好定义的方法论开发出来的软件的必然结果。

如果有一个完整的软件配置存在,那么维护工作从评价设计文档开始,确定软件重要的结构特点、性能特点以及接口特点;估量要求的改动将带来的影响,并且计划实施途径。然后首先修改设计并且对所做的修改进行仔细复查。接下来编写相应的源程序代码,使用在测试说明书中包含的信息进行回归测试,最后,把修改后的软件再次交付使用。刚才描述的事件构成结构化维护,它是在软件开发的早期应用软件工程方法论的结果。虽然有了软件的完整配置并不能保证维护中没有问题,但是确实能减少精力的浪费并且能提高维护的总