

Java

面试关键与 综合软件项目

开发全程实战

博文博主阿蜜果（谢星星）及ITjob Java高级讲师张传清，无私分享
宝贵Java学习、开发及面试实战经验

谢星星 张传清 编著

配合真正综合的项目实战案例，一例通关JDK 6.0、Eclipse 3.3、
Tomcat 6.0、Struts 1.3及Struts 2.1、Hibernate
3.6、iBATIS、Spring 3.0
面试题重要的不是答案，而是题目背后的玄机——你的解题思维、你
的基本功，甚至你的学习能力。



- ITjob特聘高级讲师、blogjava超人气技术博客博主阿蜜果（谢星星）及ITjob Java高级讲师张传清，无私分享宝贵Java学习、开发及面试实战经验
- 配合真正综合的项目实战案例，一例通关JDK 6.0、Eclipse 3.3、Tomcat 6.0、Struts 1.3及Struts 2.1、Hibernate 3.6、iBATIS、Spring 3.0
- 面试题重要的不是答案，而是题目背后的玄机——你的解题思维、你的基本功，甚至你的学习能力

远标软件开发名企就业系列

Java 面试关键与综合软件项目 开发全程实战

谢星星 张传清 编著



中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

本书为 blogjava 超人气技术博客博主“阿蜜果”呕心之作。本书不仅系统地讲解了 Java 编程的基础知识以及综合软件项目实战开发，而且完全再现了 Java 技术达人阿蜜果曾经精彩实用的学习经验以及丰富的 Java Web 项目开发实战经验。阿蜜果还根据自己多年的面试经验及一线开发经验，精心地把网上浩如烟海的 Java 面试题加以精选及深入解析，这可以让读者深刻理解相关知识的表面及背后含义，从而让读者轻松应对 Java 技术笔试和面试。

本书适用于准备找工作的初中级的 Java 开发人员，还可以作为高校及企业的培训教材和辅导教材。

本书所有示例代码及项目源码请到 ITJob 就业培训 (<http://www.5itjob.com/download/java>) 或中国水利水电出版社网站 (www.waterpub.com.cn/softdown/) 以及万水书苑 (<http://www.wsbookshow.com>) 下载。

图书在版编目 (C I P) 数据

Java面试关键与综合软件项目开发全程实战 / 谢星星, 张传清编著. — 北京: 中国水利水电出版社, 2011. 11

远标软件开发名企就业系列

ISBN 978-7-5084-8975-9

I. ①J… II. ①谢… III. ①JAVA语言—程序设计
IV. ①TP312

中国版本图书馆CIP数据核字(2011)第181270号

策划编辑: 周春元 责任编辑: 张玉玲 加工编辑: 孙丹 封面设计: 李佳

书 名	远标软件开发名企就业系列 Java 面试关键与综合软件项目开发全程实战
作 者	谢星星 张传清 编著
出版发行	中国水利水电出版社 (北京市海淀区玉渊潭南路1号D座 100038) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn
经 售	电话: (010) 68367658 (发行部)、82562819 (万水) 北京科水图书销售中心 (零售) 电话: (010) 88383994、63202643、68545874 全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京蓝空印刷厂
规 格	184mm×260mm 16开本 19.5印张 498千字
版 次	2012年1月第1版 2012年1月第1次印刷
印 数	0001—5000册
定 价	50.00元

凡购买我社图书，如有缺页、倒页、脱页的，本社发行部负责调换

版权所有·侵权必究

对于很多刚毕业想要从事 Java 软件开发工作的学生,以及正在找工作和马上要找工作的程序员来说,被考查 Java 程序开发和设计的能力是这些程序员需要面临的必修课,也是进入软件企业的重要关卡之一,因此对 Java 的面试和笔试有所准备,才能更加得心应手地应对被问到的种种问题。

本书知识地图

本书分篇分章讲述 Java 和 Java Web 技术(是指用 Java 技术来解决相关 Web 互联网领域的技术总和)各方面的知识,本书的知识地图如图 1 所示。

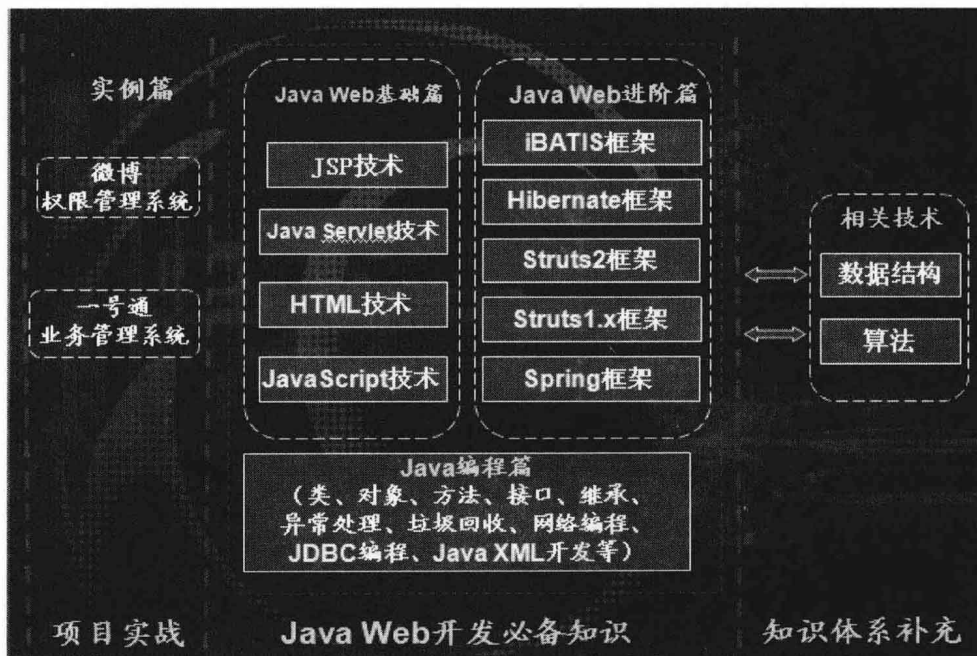


图 1 本书的知识地图

从图 1 中可以看出本书重点讲述的内容。本书每章的风格相似,首先介绍该章的核心、容易混淆和常被面试到的知识,接着通过众多的经典面试题(每小节一般 7~10 个面试题)有的放矢地针对这些知识点进行考查,因此能让读者对这些知识点进行融会贯通的应用。

本书特点

市面上 Java 编程书籍众多，本书不同于其他相关书籍的特点如下：

(1) 题量大。

本书除后两章项目实例章节没有面试题外，其余 14 章的各小节共计近 500 道面试题，题量巨大。并且后期在 ITJob 就业培训网 (<http://www.5itjob.com/>) 上会建立题库的分享页面，面试题库会随着时间的推移不断推陈出新，供读者进一步做面试的充电工作。

(2) 知识全面。

本书知识全面、面试题覆盖性强。不仅包括 Java 编程的基础和高级知识，如语法、继承、接口、多态、线程、I/O 流和网络编程等，还包括 Java Web 编程的知识，如 JSP、Java Servlet、HTML 和 JavaScript 技术等，另外因为近年来兴起的 Java 开源框架也常会被面试官提问，因此本书还分 3 章分别详细讲述了常见的 MVC 框架 (Struts 1.x 和 Struts 2.x)、ORM 框架 (Hibernate 和 iBatis)、Spring 开源框架。

(3) 深入性。

本书由资深 Java 开发和设计人员、技术写作爱好者谢星星，以及热门就业培训中心 ITJob 联袂打造，深入探索 Java 和 Java Web 编程的方方面面。本书的诸多知识点都着眼于让读者抓住技术的特点、本质和原理，因为只有抓住了这些，读者才能更加得心应手地应对相关的面试题，才能对知识点融会贯通，才能将各种技术应用到实际的项目过程中。

III

目 录

前言

致谢

第 1 章 面向对象程序设计	1	2.1.7 面试题	13
1.1 面向对象	1	2.2 Java 的数据类型	16
1.1.1 传统开发方法的问题	1	2.2.1 基本数据类型	16
1.1.2 编程语言发展简史	1	2.2.2 引用类型	17
1.1.3 面向对象概述	2	2.2.3 数组类型	18
1.1.4 面向对象的基本概念	2	2.2.4 集合类型	18
1.1.5 面向对象的基本特征	2	2.2.5 面试题	18
1.1.6 面试题	3	2.3 Java 的异常处理	23
1.2 面向对象分析、设计和编程	4	2.3.1 异常的概念和分类	23
1.2.1 面向对象分析 OOA	4	2.3.2 自定义异常	23
1.2.2 面向对象设计 OOD	4	2.3.3 面试题	23
1.2.3 面向对象编程 OOP	5	第 3 章 Java 的高级特性	25
1.3 Java 技术基础	5	3.1 Java 的继承	25
1.3.1 Java 发展历史	5	3.1.1 继承	25
1.3.2 Java 技术概述	5	3.1.2 访问控制	26
1.3.3 Java 技术的优点	6	3.1.3 方法重载	26
1.3.4 Java 虚拟机	6	3.1.4 方法覆盖	27
1.3.5 面试题	7	3.1.5 多态实现	27
第 2 章 Java 程序设计基础	9	3.1.6 隐藏技术	28
2.1 Java 语法基础	9	3.1.7 面试题	28
2.1.1 Java 中的包	9	3.2 Java 的线程	30
2.1.2 类、属性、构造器和成员方法 的声明	10	3.2.1 线程的概念	30
2.1.3 关键字	12	3.2.2 创建线程	30
2.1.4 运算符与优先级	12	3.2.3 线程的状态与调度	31
2.1.5 流程控制	13	3.2.4 线程的中断与恢复	31
2.1.6 编码约定	13	3.2.5 线程的同步	32
		3.2.6 面试题	32

3.3 Java 的高级 I/O 流	36	5.1.2 数组	64
3.3.1 流的概念	36	5.1.3 链表	65
3.3.2 流的分类	36	5.1.4 栈	68
3.3.3 对象串行化	38	5.1.5 队列	68
3.3.4 使用 java.nio 包	39	5.1.6 树和二叉树	68
3.3.5 面试题	39	5.1.7 面试题	72
3.4 Java 的高级特征	42	5.2 算法	78
3.4.1 this、super、static、final 关键字	42	5.2.1 算法的概念	78
3.4.2 内部类	42	5.2.2 算法的类型	78
3.4.3 抽象类	43	5.2.3 衡量算法效率	78
3.4.4 接口	43	5.2.4 查找算法	79
3.4.5 JDK1.5 的新特性	43	5.2.5 排序算法	80
3.4.6 面试题	43	5.2.6 递归算法	81
第 4 章 Java 的网络编程	47	5.2.7 快速排序算法	82
4.1 网络协议基础	47	5.2.8 面试题	82
4.1.1 计算机网络	47	第 6 章 JDBC 编程	87
4.1.2 OSI 参考模型	47	6.1 JDBC 基础知识	87
4.1.3 TCP/IP 协议	48	6.1.1 JDBC 的概念	87
4.1.4 UDP 协议	49	6.1.2 体系结构和连接方式	87
4.1.5 面试题	49	6.2 JDBC 基础编程	89
4.2 Socket 编程	52	6.2.1 JDBC 核心包	89
4.2.1 Socket 简介	52	6.2.2 编程步骤	90
4.2.2 服务器端套接字 ServerSocket 类	52	6.2.3 使用 DDL、DML 进行基本数据库操作	91
4.2.3 客户端套接字 Socket 类	53	6.2.4 数据库查询操作	91
4.2.4 SocketChannel 类	53	6.2.5 面试题	92
4.2.5 面试题	53	6.3 JDBC 高级编程	94
4.3 Web Service 编程	58	6.3.1 使用预编译语句	94
4.3.1 Web Service 简介	58	6.3.2 事务处理	94
4.3.2 创建、管理和发布 Web Service	59	6.3.3 存储过程	95
4.3.3 面试题	59	6.3.4 批处理更新	96
4.4 网络高级编程	60	6.3.5 处理字符大对象 CLOB	96
4.4.1 InetAddress 类	60	6.3.6 处理二进制大对象 BLOB	96
4.4.2 基于 UDP 的数据报和套接字	60	6.3.7 使用 RowSet	96
4.4.3 URL 和 URLConnection 类	61	6.3.8 使用 JDBC 操作元数据 Meta Data	96
4.4.4 安全网络通信 SSL	62	6.3.9 面试题	97
4.4.5 面试题	62	第 7 章 Java XML 编程	100
第 5 章 数据结构和算法	64	7.1 XML 基础	100
5.1 数据结构	64	7.1.1 XML 的概念	100
5.1.1 数据结构概念	64		

7.1.2	定义 XML 文档	100	10.2.1	HTTP 协议概述	148
7.1.3	XML 文档规则	101	10.2.2	面试题	150
7.1.4	面试题	101	10.3	Servlet 高级功能	152
7.2	使用 Java 解析 XML	103	10.3.1	会话类 HttpSession	152
7.2.1	解析器介绍	103	10.3.2	Cookie 类 Cookie	152
7.2.2	常用解析技术	103	10.3.3	Servlet 过滤器	152
7.2.3	面试题	105	10.3.4	Servlet 监听器	153
7.2.3	面试题	105	10.3.5	面试题	153
第 8 章	HTML 基础和进阶	109	第 11 章	JSP 技术	158
8.1	基本知识	109	11.1	基础知识	158
8.1.1	HTML 概述	109	11.1.1	JSP 技术概述	158
8.1.2	HTML 常用标签	109	11.1.2	JSP 内置对象	159
8.1.3	面试题	110	11.1.3	面试题	159
8.2	表格	113	11.2	JSP 的语法	160
8.2.1	表格标签及属性	113	11.2.1	JSP 模板元素	160
8.2.2	面试题	114	11.2.2	JSP 指令元素	160
8.3	表单	115	11.2.3	Java 脚本元素	161
8.3.1	表单概述	115	11.2.4	JSP 动作元素	161
8.3.2	表单常用的标签及属性	115	11.2.5	面试题	163
8.3.3	表单标签的事件	117	11.3	JSP 的高级功能	164
8.3.4	面试题	117	11.3.1	标准标签库	164
8.4	框架	119	11.3.2	自定义标签	164
8.4.1	框架概述	119	11.3.3	表达式语言 EL	164
8.4.2	框架常用的标签及属性	120	11.3.4	面试题	165
8.4.3	面试题	120	第 12 章	Struts 开源框架技术	169
第 9 章	JavaScript 基础和进阶	123	12.1	Struts1.x 概述	169
9.1	基本知识	123	12.1.1	技术产生背景	169
9.2	数组	128	12.1.2	MVC 模式	170
9.3	文档对象模型	130	12.1.3	工作原理	170
9.4	窗口对象	133	12.1.4	核心类	170
9.5	事件处理	136	12.1.5	面试题	170
9.6	AJAX	138	12.2	Struts1.x 的配置文件	172
第 10 章	Java Servlet 技术	143	12.2.1	配置文件 web.xml	172
10.1	Servlet 基础知识	143	12.2.2	配置文件 struts-config.xml	172
10.1.1	Java Servlet 技术产生背景	143	12.2.3	资源文件.properties	175
10.1.2	Servlet 概述	143	12.2.4	验证文件 validation.xml	176
10.1.3	请求类 HttpServletRequest	144	12.2.5	面试题	176
10.1.4	响应类 HttpServletResponse	145	12.3	Struts1.x 的标签库	178
10.1.5	面试题	145	12.3.1	html 标签库	178
10.2	HTTP 协议的基本概念	148			

12.3.2	bean 标签库	179	13.4.4	延迟加载	204
12.3.3	logic 标签库	180	13.4.5	缓存机制	205
12.3.4	nested 标签库	180	13.4.6	面试题	206
12.3.5	tiles 标签库	181	13.5	与开源 ORM 框架 iBATIS 的比较	209
12.3.6	面试题	181	13.5.1	iBATIS 概述	209
12.4	Struts2 技术	182	13.5.2	iBATIS 的基本使用	209
12.4.1	Struts2 简介	182	13.5.3	面试题	211
12.4.2	Struts2 的配置文件	183	第 14 章 Spring 开源框架应用		212
12.4.3	Struts2 的数据校验	184	14.1	Spring 的基础知识	212
12.4.4	Struts2 的类型转换	186	14.1.1	产生背景	212
12.4.5	Struts2 的标签库	187	14.1.2	Spring 的基本框架	213
12.4.6	Struts2 的拦截器	188	14.1.3	Spring 的核心接口和类	214
12.4.7	Struts2 的国际化	189	14.1.4	面试题	215
12.4.8	面试题	190	14.2	配置文件	215
第 13 章 Hibernate 开源框架应用		191	14.2.1	XML 格式配置元数据	215
13.1	Hibernate 的基础知识	191	14.2.2	数据源配置	216
13.1.1	技术产生背景	191	14.2.3	事务配置	217
13.1.2	对象关系映射 ORM	191	14.2.4	邮件服务配置	219
13.1.3	核心接口类	192	14.2.5	定时任务配置	219
13.1.4	对象的三种状态	192	14.2.6	面试题	220
13.1.5	面试题	193	14.3	控制反转 IoC	222
13.2	Hibernate 的配置文件	195	14.3.1	IoC 的概念	222
13.2.1	配置 hibernate.cfg.xml	195	14.3.2	IoC 的实现原理	223
13.2.2	配置 hbm.xml	196	14.3.3	实例化容器	225
13.2.3	配置一对多或多对一关联	197	14.3.4	实例化 bean	226
13.2.4	配置一对一关联	197	14.3.5	注入方式	226
13.2.5	配置多对多关联	198	14.3.6	面试题	227
13.2.6	面试题	199	14.4	面向切面编程 AOP	228
13.3	Hibernate 的查询功能	201	14.4.1	AOP 的概念	228
13.3.1	OID 检索方式	201	14.4.2	AOP 的实现原理	228
13.3.2	HQL 检索方式	201	14.4.3	AOP 的通知类型	231
13.3.3	QBC 检索方式	201	14.4.4	使用场景	233
13.3.4	本地 SQL 查询	202	14.4.5	面试题	233
13.3.5	连接查询	202	14.5	与其他开源框架的整合	235
13.3.6	面试题	202	14.5.1	Struts1.x 与 Spring 的整合	235
13.4	Hibernate 的其他功能	203	14.5.2	Struts2.x 与 Spring 的整合	236
13.4.1	保存、更新和删除实体对象	203	14.5.3	Hibernate 与 Spring 的整合	237
13.4.2	调用存储过程	203	14.5.4	iBATIS 与 Spring 的整合	239
13.4.3	进行事务控制	204	14.5.5	面试题	239

第 15 章 微博权限管理系统	241	15.5 系统测试	268
15.1 系统需求分析	241	第 16 章 一号通业务管理系统	271
15.1.1 基本概念	241	16.1 系统需求分析	271
15.1.2 权限系统的基本需求	242	16.2 系统架构设计	272
15.1.3 微博权限系统的需求	243	16.3 系统数据库设计	273
15.2 系统架构设计	243	16.3.1 系统实体	273
15.3 系统详细设计	244	16.3.2 数据库总体设计	274
15.3.1 系统实体	244	16.3.3 表结构设计	274
15.3.2 数据库总体设计	244	16.3.4 系统模块结构	275
15.3.3 表结构设计	245	16.4 系统实现设计	276
15.3.4 系统模块结构	248	16.4.1 环境搭建	276
15.4 系统实现设计	248	16.4.2 实现 DAO 组件层	276
15.4.1 环境搭建	248	16.4.3 实现业务逻辑层	279
15.4.2 实现 DAO 组件层	248	16.4.4 实现控制层	284
15.4.3 实现业务逻辑层	251	16.4.5 实现表现层	288
15.4.4 实现控制层	256	16.4.6 配置文件	294
15.4.5 实现表现层	261	16.5 系统测试	297
15.4.6 配置文件	265		

面向对象程序设计

【本章导读】

OO (Object Oriented, 面向对象) 是当前计算机界关心的重点, 它 20 世纪是 90 年代软件开发方法的主流, 它的概念和应用已超越了程序设计和软件开发, 扩展到了很广泛的范围。面向对象的分析、设计和编程因其优越性而受到广大开发人员的追捧, 成为主流的程序分析和设计方法。

本章首先对面向对象的重要内容进行讲解, 包括发展背景、基本概念和基本特征等知识; 接着讲述了面向对象的分析、设计和编程技术; 最后对当下流行的面向对象编程语言——Java 技术的发展历史、概述、优点, 以及 Java 技术中的虚拟机进行了重点讲述。为后续章节讲述 Java 编程语言打下了坚实的基础。

1.1 面向对象

1.1.1 传统开发方法的问题

传统开发方法存在的问题如下:

- (1) 软件重用性差: 重用性是指同一事物不经修改或稍加修改就可多次重复使用的性质。
- (2) 软件可维护性差: 软件工程强调软件的可维护性, 强调文档资料的重要性, 规定最终的软件产品应该由完整、一致的配置成分组成。在软件开发过程中, 始终强调软件的可读性、可修改性和可测试性是软件的重要可维护性质量指标。
- (3) 开发出的软件不能满足用户需要: 在开发需求模糊或需求动态变化的系统时, 所开发出的软件系统往往不能真正满足用户的需要。

1.1.2 编程语言发展简史

编程语言从最初的复杂机器语言发展到如今的高级面向对象语言, 经历了近 50 年的历史, 产生了四代计算机语言。

- (1) 第一代 (1954~1958): FORTRAN I、ALCOL 58、Flowmatic 等语言, 主要应用于数学

和工程应用，几乎全部使用数学词汇。

(2) 第二代 (1959~1961): FORTRAN II、ALCOL 60、COBOL、Lisp 等语言，重点在于算法抽象，集中于告诉机器应该做什么。

(3) 第三代 (1962~1970): PL/1 (FORTRAN+ALCOL+COBOL)、Pascal、Simula (最早出现类和对象思想的语言)，对数据 (类) 抽象，描述各种类的类型。

(4) 第四代 (1970 至今): Smalltalk、C++、Ada、Java、Eiffel、PHP 等语言，都是面向对象或基于对象的语言。

1.1.3 面向对象概述

面向对象 (Object Oriented, OO) 是当前计算机界关心的重点，它是 90 年代软件开发方法的主流。面向对象的概念和应用已超越程序设计和软件开发，如数据库系统、交互式界面、应用结构、应用平台、分布式系统、网络管理结构、CAD 技术和人工智能等领域。

起初，面向对象专指在程序设计中采用封装、继承、多态等设计方法。可是这个定义显然不能再适合现有情况。面向对象的思想已经涉及到软件开发的各个方面，例如 OOA (Object Oriented Analysis, 面向对象的分析)、OOD (Object Oriented Design, 面向对象的设计)、OOP (Object Oriented Programming, 面向对象编程)。

1.1.4 面向对象的基本概念

面向对象的基本概念包括：

(1) 对象：是人们要进行研究的任何事物，从最简单的整数到复杂的飞机等均可看做对象，它不仅能表示具体的事物，还能表示抽象的规则、计划或事件。

(2) 对象的状态和行为：对象具有状态，一个对象用数据值来描述它的状态。对象还有操作，用于改变对象的状态，对象及其操作就是对象的行为。对象实现了数据和操作的结合，使数据和操作封装于对象的统一体中。

(3) 类：具有相同或相似性质的对象的抽象就是类。对象的抽象是类，类的具体化就是对象，也可以说类的实例是对象。类具有属性，它是对象的状态的抽象，用数据结构来描述类的属性。类具有操作，它是对象的行为的抽象，用操作名和实现该操作的方法来描述。

(4) 类的结构：在客观世界中有若干类，类之间有一定的结构关系。通常有两种主要的结构关系，即“一般—具体”结构关系和“整体—部分”结构关系。“一般—具体”结构称为分类结构或者称为“is a”关系；“整体—部分”结构称为组装结构或者称为“has a”关系。

(5) 消息和方法：对象之间进行通信的结构叫做消息。在对象的操作中，当一个消息发送给某个对象时，消息包含接收对象去执行某种操作的信息。发送一条消息至少要包括说明接收消息的对象名和发送给该对象的消息名（即对象名、方法名）。一般还要对参数加以说明，参数可以是认识该消息的对象所知道的变量名，或者是所有对象都知道的全局变量名。

1.1.5 面向对象的基本特征

程序语言发展至今，已经出现了多种风格不一的编程语言，大体主要分为以下两种风格的语言：

- (1) 面向过程语言（以 C 语言为代表）。
- (2) 面向对象语言（Java、Smalltalk、C++等）。

语言的好坏是针对不同方面的应用程序而言的。例如,面向过程语言适用于计算密集型操作(复杂算法等)的设计,而面向对象语言适用于广阔的应用程序范围,这也是 Java 编程语言能迅速发展并占据广大市场的原因之一。

只要是具备以下特点的语言都可以将其看做是实现了面向对象的语言:

(1) 抽象 (Abstraction): 表示一个对象与其他所有对象区别的基本特征,为不同的观察者提供不同的角度来观察对象。也可以将抽象定义为:从许多事物中舍弃个别的、非本质的特征,抽取共同的、本质性的特征。一个类就是这样一种抽象,它反映了与应用有关的重要性质,而忽略其他一些无关内容。任何类的划分都是主观的,但必须与具体的应用有关。

(2) 封装 (Encapsulation): 是一种信息隐蔽技术,是一个划分抽象的结构和行为的过程。封装和抽象是互补的,抽象着重于对象的行为,而封装着重于对象行为的实现。抽象实现抽取众多对象的公共特征和行为,而封装则是在不同的抽象之间设置明显的分隔线,从而导致每一个观察者关注内容的明显分离。没有抽象的封装没有意义,没有封装的抽象是不完整的。

(3) 继承 (Inheritance): 一个类共享一个或多个类中定义的结构和行为,表示的是一种泛化/特化的层次关系。它是类和类之间的关系之一,是面向对象系统的最基本要素。在现实中有许多这样的实例,例如人类是一种哺乳动物,因为人类具备所有哺乳动物的特征,也就是说,人类继承了哺乳动物的特点。

(4) 多态性 (Polymorphism): 是指相同的操作、函数或过程可作用于多种类型的对象,并获得不同的结果。不同的对象收到同一消息可以产生不同的结果,这种现象称为多态性。多态性允许每个对象以适合自身的方式去响应共同的消息,它增强了软件的灵活性和重用性。

1.1.6 面试题

1. 问:什么是面向对象?现在常用的面向对象语言有哪些?

答案:面向对象 OO = 面向对象的分析 OOA + 面向对象的设计 OOD + 面向对象的编程 OOP。通俗的解释就是万物皆对象,把所有的事物都看做一个个可以独立的对象,它们可以自己完成自己的功能,而不是像 C 那样分成一个个函数。

现在纯正的面向对象语言主要有 Java 和 C#, C++ 也支持 OO, C 是面向过程的。

2. 问:面向对象的基本特征有哪些方面?请简短说明。

答案:主要有如下四个方面:

(1) 抽象:忽略一个主题中与当前目标无关的那些方面,以便更充分地注意与当前目标有关的方面。抽象包括两个方面:一是过程抽象,二是数据抽象。

(2) 继承:一个类共享一个或多个类中定义的结构和行为,表示一种泛化/特化的层次关系。

(3) 封装:是一种信息隐蔽技术,是一个划分抽象的结构和行为的过程。封装和抽象是互补的,抽象着重于对象的行为,而封装着重于对象的行为的实现。

(4) 多态性:多态性是指相同的操作或函数、过程可作用于多种类型的对象上并获得不同的结果。不同的对象收到同一消息可以产生不同的结果,这种现象称为多态性。

3. 问:在类的结构中,“is a”和“has a”的不同点是什么?并请分别说明这两种结构在现实生活中的实例。

答案:is a:代表类之间的继承关系。比如 PC 机是计算机,工作站也是计算机。PC 机和工作站是两种不同类型的计算机,但都继承了计算机的共同特性。因此在用 Java 语言实现时,应该将

PC 机和工作站定义成两种类，均继承计算机类。

has a: 代表对象和它的成员的从属关系。同一种类的对象，通过它们的属性的不同值来区别。比如一台 PC 机的操作系统是 Windows，另一台 PC 机的操作系统是 Linux。操作系统是 PC 机的一个成员变量，根据这一成员变量的不同值可以区分不同的 PC 机对象。

4. 问：Java 中的类和对象的区别与联系是什么？

答案：对象是人们要进行研究的任何事物，从最简单的整数到复杂的飞机等均可看做对象。类是具有相同或相似性质的对象的抽象。因此，对象的抽象是类，类的具体化就是对象，也可以说类的实例是对象。

1.2 面向对象分析、设计和编程

1.2.1 面向对象分析 OOA

OOA (Object Oriented Analysis, 面向对象的分析) 模型的结构为:

(1) 第一层: 对象—类层。

表示待开发系统的基本构造块。图符的外层边界表示实例边界，实际上它表明对象是非空的，而图符的内层边界则表示类边界。在某些情形下，定义这种对象是很有用的，它们被称为模板类或抽象类。

(2) 第二层: 属性层。

对象的属性和实例连接共同组成了 OOA 模型的属性层，通常把对象所存储的数据称为对象的属性。类的实例之间互相约束，它们必须遵从应用领域的某些限制条件或事务规则。

(3) 第三层: 服务层。

对象的服务和消息通信组成了 OOA 模型的服务层。把对象所做的工作称为服务或方法。系统的不同对象分别执行一定的工作或功能，它们之间通过消息通信，即所谓的协同。

(4) 第四层: 结构层。

该层负责捕捉特定应用领域中的结构关系。泛化、特化等结构和整体——部分结构表达了人类的一种基本组织方式，即自然的整体和部分的结构关系，从而把一些部分的聚合构造造成整体。例如，一辆汽车由发动机、传动装置和刹车装置组成。

(5) 第五层: 主题层。

相当于全局系统的子系统或子模型。由于 OOA 模型的结构庞大而复杂，因此众多的对象有时很难处理。于是，可以把对象归到各个主题层中，可以把有关的对象用一个边框框起来加以实现。例如，在一个控制系统中，众多的对象可分为“管理”和“控制”两个主题。

1.2.2 面向对象设计 OOD

OOD (Object Oriented Design, 面向对象的设计) 是对 OOA 的细化，它强调的是复杂系统的正确和有效的构建，通过不同的方法来着重于系统的逻辑设计、物理设计。如果说分析是产生系统的类和对象，那么设计则确定类和类之间的关系、对象和对象之间的关系、类的状态转换、模块之间的依赖性等。

OOD 与 OOA 的区别:

- (1) OOA 偏重于理解问题, 描述软件要做什么, 而 OOD 偏重于解决方案, 描述软件要如何做。
- (2) OOA 只考虑理想的设计, 不关心技术与实现底层的细节, 而 OOD 需要得到更具体详细、更接近于真实代码的设计方案。
- (3) 在设计结果的描述上, OOA 偏重于描述对象的行为, 而 OOD 偏重于描述对象的属性与方法。
- (4) OOA 只关注功能性需求, OOD 还需要关注非功能性需求。

1.2.3 面向对象编程 OOP

OOP (Object Oriented Programming, 面向对象程序设计, 又称为面向对象编程) 是一种实现方法, 程序被组织成对象的协作集合, 每一个对象代表某一个类的实例, 而类则是通过继承关系联系在一起。

OOP 阶段必须满足三个标准条件:

- (1) 使用对象而不是算法作为其基本逻辑构件。
- (2) 任何一个对象都必须是某一个类的实例。
- (3) 类通过继承关系和其他类相关。

1.3 Java 技术基础

1.3.1 Java 发展历史

1991 年, 电视机、机顶盒和录像机的开发设计需要一种可移植、方便、高效的计算机语言。为了满足这种需求, 由 Sun 公司的 Patrick Haughton 和 James Gosling 领导的技术小组开发了 Java。20 世纪 90 年代中期, Sun 推出了 Sun Java Development Toolkits 1.0, 简称 JDK1.0。JDK1.0 是一个功能强大的软件包, 可以用来开发小应用程序和多种操作系统 (Sun Solaris、Windows NT、Windows 95、Macintosh) 的应用程序。

1998 年 Sun 推出 Java 2 Platform, 它定义了所有 Java 技术的概念和标准, 既包括已经实现的技术也包括尚未实现的技术; 既包括 Sun 实现的, 也包括其他公司实现的。目前, Java 2 SDK1.6 是 Java 2 Platform 的最新定义。这个定义又可以细分为四个版本:

- (1) 标准版: J2SE, 用于开发普通的小应用程序和应用程序。
- (2) 企业版: J2EE, 又称为 Java EE, 用于企业级应用。
- (3) 微型版: J2ME, 又称为 Java ME, 用于开发移动电话、机顶盒、个人数字设备等。
- (4) JavaCard: 适用于智能卡的 Java 平台。

1.3.2 Java 技术概述

Java 技术由以下这些技术层面组成:

- (1) Java 编程语言: 定义变量、表达式、逻辑控制等基本规则。
- (2) Java 类库: Java 软件工具包 (SDK) 为程序员提供了几千个类, 包括基本的数学函数、数组和字符串、窗口、GUI、I/O、网络等。
- (3) Java 运行环境: 包括 Java 字节代码的加载器、校验器、Java 虚拟机。

(4) Java 虚拟机: Java 技术出于跨平台、可移植的考虑,没有将程序的源代码编译连接成 CPU 的指令序列,直接交给计算机执行。Java 技术在不同的硬件、不同的操作系统之上定义了完全相同的支持 Java 程序运行的虚拟计算机。Java 源程序被编译成字节码,编译后的文件名后缀是.class,在 Java 虚拟机上运行。

(5) Java 工具: 编译器、注释器和文档生成器等工具。

(6) Java 小应用程序: 小应用程序 (Applets) 是一种存储于 WWW 服务器上的用 Java 编写的程序,它通常由浏览器下载到客户系统中,并通过支持 Java 运行环境的浏览器运行。它由超文本标识语言的 Web 网页来调用。

(7) Java 应用程序: Java Applications 是一种独立的程序,它不需要任何 Web 浏览器来执行。它们是普通的应用程序,可运行于任何具备 Java 运行环境的设备中。

(8) 其他 Java 程序: JavaBean、Servlet 和 JSP 等。

1.3.3 Java 技术的优点

Java 技术主要具有以下优点:

(1) 面向对象: Java 是面向对象的程序设计语言,与面向过程的语言相比,它更能反映人类对世界的认识和处理模式,具有良好的代码重用性。处理复杂、庞大而且不断变化的信息系统必须使用面向对象的程序设计语言。

(2) 平台无关: 通过采用虚拟机技术真正实现了与平台无关,编译过的 Java 文件既可以在 Windows 上运行,也可以在 Linux 和 UNIX 等操作系统上运行。

(3) 改进的解释性: 为弥补解释语言速度较慢的不足,Java 采用预编译的方法,将源程序生成字节代码。另一方面,有些 Java 运行环境采用 JIT 编译器将字节代码编译成机器码直接运行,这种运行环境对于重复执行的服务器端软件特别有效,可以达到 C 语言的速度。

(4) 安全: 在编译时删除了指针和内存分配,在运行时检查字节代码,拒绝执行非法的内存访问、超越权限的访问等,可以防御黑客攻击。

(5) 多线程: 允许一个应用程序同时做多个任务。

(6) 动态性: Java 允许下载代码模块,因此当程序运行时也能动态升级。

(7) 简单和健壮性: 它继承了 C++ 语言的优点,去掉了 C++ 的难点,又加入新的特性,许多高级语言都要由程序员进行指针运算和存储器管理,这项工作既复杂又容易出错。Java 不需要开发人员进行指针运算和存储器管理,简化了设计,减少了出错的可能性。

1.3.4 Java 虚拟机

JVM (Java Virtual Machine, Java 虚拟机) 是一个想象中的机器,在实际的计算机上通过软件模拟来实现。它在 Java 运行环境中处于核心地位,它使 Java 可以跨平台运行。Java 虚拟机隐藏了计算机硬件和操作系统的复杂性,使开发人员只面对单一的支持 Java 的计算机。

当开发人员写好 Java 源程序后,使用编译器将源代码转换成 JVM 的指令序列 (字节代码),保存为.class 文件。执行 Java 程序时,JVM 负责解释字节代码,JVM 的指令转换成真实的机器指令并执行。Java 虚拟机由以下几部分组成:

(1) 指令集: 支持大约 248 个字节码,每个字节码执行一种基本的 CPU 运算,Java 指令集相当于 Java 程序的汇编语言。

- (2) 寄存器：用于保存机器的运行状态，与微处理器中的某些专用寄存器类似。
- (3) 栈：分为三个区域，即局部变量区、运行环境区和操作数区。
- (4) 垃圾收集堆：是一个运行时数据区，类的实例（对象）从中分配空间。Java 语言具有无用单元收集能力，它不给程序员显式释放对象的能力。Java 不规定具体使用的无用单元收集算法，可以根据系统的需求使用各种各样的算法。
- (5) 类文件格式。
- (6) 存储区。

1.3.5 面试题

1. 问：下列语句（ ）是正确的。

- A. JVM 的自动垃圾收集会防止内存溢出
- B. 程序能够建议垃圾收集器执行收集操作，并可强制让其执行
- C. 垃圾收集是平台独立的
- D. 当所有的线程都没有对象引用的时候，该对象将被置为 null 值

解析：本题考查 JVM 中的垃圾回收机制。自动垃圾收集并不会每次都阻止内存溢出，因此 A 选项是错误的。程序能够使用 `System.gc()` 等方法建议收集器立即进行收集，但无法强制执行收集操作，所以 B 选项是错误的。垃圾回收器不是平台独立的，所以 C 选项是错误的。

答案：D

2. 问：在 Java 中，GC 是什么的英文简写？表示什么意思？为什么要有 GC？

答案：GC 是 `Gabage Collection` 的简写，表示垃圾收集器。使用 GC，程序员不用担心内存管理，因为垃圾收集器会自动进行管理。一般情况下，Java 会自动进行垃圾收集，如果需要手动请求垃圾收集，可以调用下面的方法之一：`System.gc()` 或 `Runtime.getRuntime().gc()`。

3. 问：Java 虚拟机的运行原理是什么？

答案：在 Java 中引入了虚拟机的概念，即在机器和编译程序之间加入了一层抽象的虚拟的机器。这台虚拟的机器在任何平台上都给编译程序提供一个共同的接口。编译程序只需要面向虚拟机，生成虚拟机能够理解的代码，然后由解释器来将虚拟机代码转换为特定系统的机器码执行。

在 Java 中，这种供虚拟机理解的代码叫做字节码，它不面向任何特定的处理器，只面向虚拟机。每一种平台的解释器是不同的，但是实现的虚拟机是相同的。Java 源程序经过编译器编译后变成字节码，字节码由虚拟机解释执行，虚拟机将每一条要执行的字节码送给解释器，解释器将其翻译成特定机器上的机器码，然后在特定的机器上运行。

4. 问：什么是 Java 的类加载器？

答案：类加载器为程序的执行加载所需要的类。类加载器将本地文件系统的类名空间与来自网络的类名空间相分离，以增加安全性。当全部类被加载后，可执行文件的存储器格式被确定。这时，特定的存储器地址被分配给变量并创建检索表格。由于存储器格式在运行时出现，因而 Java 解释器增加了保护以防止对限制代码区的非法访问。

5. 问：什么是 JVM？它由哪几部分组成？

答案：JVM (Java Virtual Machine, Java 虚拟机) 是一个想象中的机器，在实际的计算机上通过软件模拟来实现。它在 Java 运行环境中处于核心地位，它使 Java 可以跨平台运行。

JVM 包括一套字节码指令集、一组寄存器、一个栈、一个垃圾回收堆和一个存储方法域。