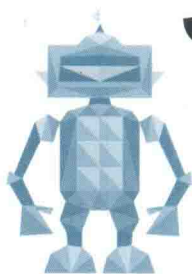
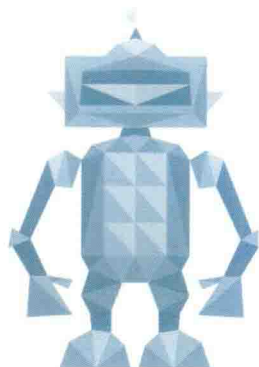


经典畅销书系“深入理解Android”系列又一重磅巨著  
从源码角度深度剖析Android Java虚拟机ART架构、设计和实现原理，  
深刻揭示JVM工作流程与机制

# 深入理解 Android Java虚拟机ART



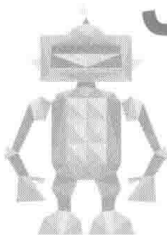
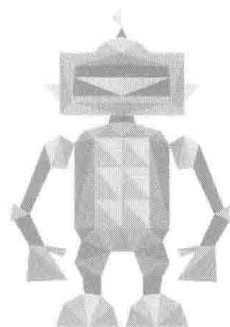
Understanding  
Android Internals  
ART JVM

邓凡平（中国民生银行信息科技部） 著



机械工业出版社  
China Machine Press

# 深入理解 Android Java虚拟机ART



Understanding  
Android Internals  
ART JVM

邓凡平（中国民生银行信息科技部） 著



机械工业出版社  
China Machine Press

## 图书在版编目 (CIP) 数据

深入理解 Android: Java 虚拟机 ART/ 邓凡平著. —北京: 机械工业出版社, 2019.4  
(移动开发)

ISBN 978-7-111-62122-5

I. 深… II. 邓… III. 移动终端 - 应用程序 - 程序设计 IV. TN929.53

中国版本图书馆 CIP 数据核字 (2019) 第 036089 号

## 深入理解 Android: Java 虚拟机 ART

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 张锡鹏

责任校对: 李秋荣

印刷: 北京瑞德印刷有限公司

版次: 2019 年 4 月第 1 版第 1 次印刷

开本: 186mm × 240mm 1/16

印张: 59

书号: ISBN 978-7-111-62122-5

定价: 169.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88379426 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294

读者信箱: hzit@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

HZBOOKS | 华章科技 | Science & Technology



## 重塑科技引领动能，打造科技金融银行

近几年，随着移动互联、云计算、大数据及人工智能等新兴技术的广泛应用，全社会已经进入了一个由移动互联网构成的高度数字化时代，而数字化浪潮则将金融机构带入了由线上支付商、互联网巨头以及其他金融科技创新企业共同竞争的新金融生态圈。

从支撑到伙伴，再到引领，科技角色重新定位。一直以来，科技在银行中扮演更多的是支撑者的角色，在确保系统安全稳定运行的基础上配合业务进行需求开发。面对时下金融科技蓬勃发展的浪潮以及互联网企业给传统银行业带来的冲击，银行业科技唯有主动出击，驱动业务转型，才能在冲击中乘风破浪，重塑商业银行在新时代的竞争优势。

面对数字化发展趋势带来的冲击和挑战，民生银行信息科技部积极践行科技引领理念，建设智慧银行。一方面在人工智能、云计算、物联网、区块链等新兴技术领域深入研究，大胆创新业务模式，实现重点领域突破。另一方面积极营造创新机制，推动“轻组织”改革，鼓励员工创新，并成立了专门的创新研发组织，推动科技创新。

本书作者来自民生银行信息科技部的创新技术研究院物联网团队。该团队主要研究物联网技术并积极探索其在金融领域中的应用。众所周知，物联网离不开数以亿计的各色终端设备，而基于安卓系统的智能设备又属于其中功能最为强大和完善的一种。对安卓系统的了解将有助于我们更好地将这种类型的设备应用于物联网或相关领域。

本书聚焦于安卓智能设备上 Java 虚拟机这一核心的底层技术实现。通过扎实和细致的源代码分析，对 JVM 进行了较为全面和深入的剖析，其详尽程度以及深厚的理论功底使得本书在理论研究和实践应用方面都颇具特色，相信本书可以有效提升读者在相关领域的开发能力。

牛新庄 博士。现任中国民生银行总行信息科技部总经理，民生科技有限公司执行董事、总经理。国务院互联网+行动专家咨询委员会专家，也是国内顶尖数据架构与科技治理专家。同时担任浙江大学等高校的兼职教授和客座教授。曾获“IBM 杰出软件专家奖”“中国杰出数据库工程师奖”“IT168 技术卓越奖”。拥有 OCP、AIX 等 20 多项国际认证。著有《DB2 性能调整和优化》等书。

## 本书主要内容及特色

本书是笔者“深入理解 Android”系列的第四本。本书将关注 Android 系统中至关重要的部分——Java 虚拟机 ART。市面上介绍 Java 虚拟机的书籍非常多，但鲜少有书籍能从虚拟机源代码出发对其进行详细分析。随着 Android 设备的大规模普及，ART 虚拟机已经成为当今使用最为广泛的 JVM 之一。所以，对 ART 虚拟机进行研究有着非同寻常的意义。本书的出现一定程度上填补了这方面的空白。

本书的主要内容概述如下：

- ❑ 第 1 章介绍 ART 虚拟机学习前需要准备好的工具、环境等。
- ❑ 第 2 章介绍 Class 文件的格式及内容。
- ❑ 第 3 章介绍 Android 中 Dex 文件的格式。
- ❑ 第 4 章介绍 ELF 文件格式。
- ❑ 第 5 章介绍 C++11 相关的、能帮助读者阅读 ART 源码的必备知识。
- ❑ 第 6 章以编译原理为基础，介绍 ART 虚拟机编译相关的知识。
- ❑ 第 7 章以 ART Runtime 对象的创建为主线，介绍主要的模块及一些关键类、数据结构等知识。
- ❑ 第 8 章以 ART Runtime 的 Start 为主线进行分析，覆盖的内容包括相关模块的启动、类的解析、加载、链接、初始化等。
- ❑ 第 9 章介绍 dex 字节码转机器码的核心进程 dex2oat 以及 .oat 和 .art 文件格式。
- ❑ 第 10 章介绍虚拟机的解释执行和 JIT 部分以及异常的投递和处理的过程。
- ❑ 第 11 章介绍 JNI 在 ART 虚拟机的实现。
- ❑ 第 12 章介绍虚拟机 Java 线程执行相关的知识，包括线程暂停和恢复运行、synchronized、Object wait/notify 的实现、volatile 变量的读写处理等。
- ❑ 第 13 章介绍内存分配和释放相关的知识。包括 ART 虚拟机中的各种 Space 类型、new 指令的实现以及 ART 虚拟机中 Heap 模块的部分内容。

- ❑ 第 14 章介绍和垃圾回收有关的基础知识以及相关垃圾回收器，还有 Java Reference 的处理以及 Heap 模块的部分内容。

本书通过理论和代码相结合的方式进行讲解，旨在引领读者一步步了解 Android 系统中 JVM 的工作原理。

## 读者对象

- ❑ Android 系统开发工程师

系统开发工程师常常需要深入理解 Android 平台上各个系统的运转过程。本书所涉及的 Java 虚拟机是从事相关工作的读者在工作和学习中最想了解的。

- ❑ Android 应用开发工程师

Android 应用开发工程师所开发的程序是运行在 JVM 中的。如果能更深入地了解 JVM 的实现将极大帮助开发工程师写出更高质量的程序。

- ❑ 对 JVM 感兴趣的在校高年级本科生、研究生等研究人员

JVM 的理论书籍非常多，但很少有从分析源代码的角度来介绍其工作原理的。这本理论与代码实现深度结合的书籍一定可在该领域助相关研究人员一臂之力。

## 如何阅读本书

本书是一本有一定深度的书籍，所以读者在阅读时：

- 请务必首先阅读第 1 章。后续如果碰到阅读上的困难，可能还需时常回顾第 1 章。

本书的内容是经过笔者精心编排的，如果读者不是很有把握的话，建议严格按照顺序阅读。

本书的某些章节涉及了笔者在撰写它们时所参考的资料。这些资料较多，读者可根据它们开展进一步的研究工作。

另外，和笔者之前出版的《深入理解 Android》卷 I 以及卷 II 类似的是：本书在每章开头都把本章涉及的源码路径全部列出，而在具体分析源码时，则只列出该源码的文件名及所分析的函数或相关数据结构名。例如：

```
[AndroidRuntime.cpp->AndroidRuntime::start]
```

```
//这里是源码分析和一些注释
```

最后，本书在描述类之间的关系及函数调用流程上，使用了 UML 的静态类图及序列图。UML 是一个强大的工具，但它的建模规范过于繁琐，为更简单清晰地描述事情的本质，本书并未完全遵循 UML 的建模规范。这里仅举两例，如图 1 和图 2 所示。

在图 1 中：

- ❑ 外部类内部的方框用于表示内部类。另外，外部类 A、内部类 B 也用于表示内部类。
- ❑ 接口和普通类用同一种框图表示。

图 2 所示为本书描述数据结构时使用的 UML 图。

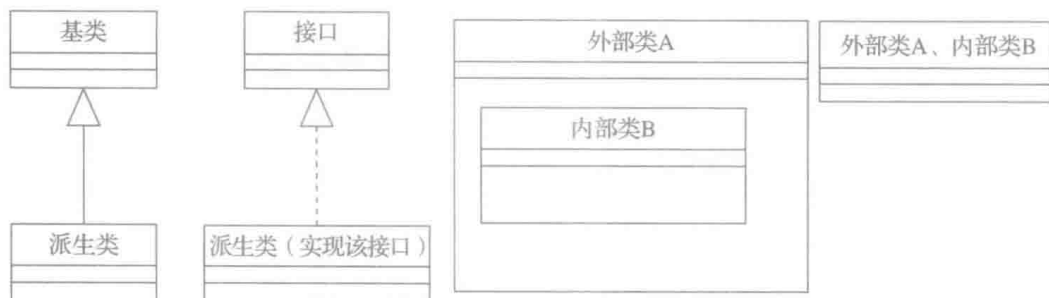


图 1 UML 示例图之一

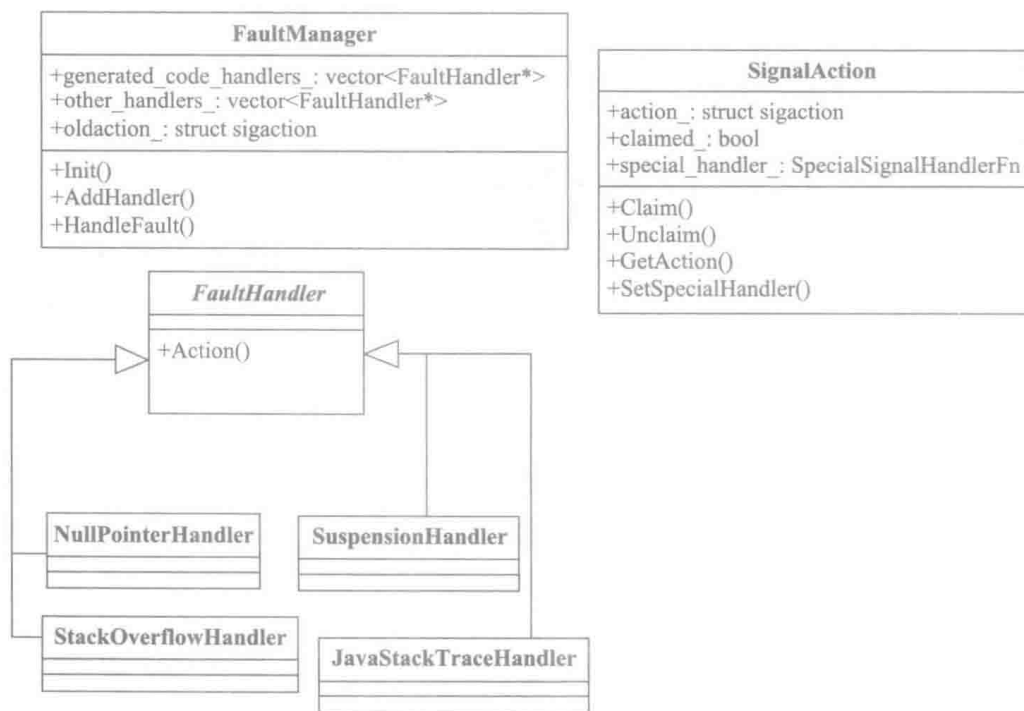


图 2 UML 示例图之二

图 2 为本书描述数据结构及成员时使用的 UML 图例。

**特别注意** 本书使用的 UML 图都比较简单，读者不必花费大量时间专门学习 UML。另外，出于方便考虑，本书所绘制的 UML 图没有严格遵守 UML 规范。这一点敬请读者谅解。

本书涉及的 Android 源码及一些开发工具的下载可通过笔者的博客 [blog.csdn.net/innost](http://blog.csdn.net/innost) 首页置顶文章“深入理解 Android 系列书籍资源分享更新”查看。关于它们的使用详情，请读者阅读本书第 1 章了解。



## 勘误和支持

由于作者的水平有限，加之编写时间仓促，书中难免会出现一些错误或不准确的地方，恳请读者不吝赐教。若有问题，可通过邮件或在博客上留言与笔者共同商讨。笔者的联系方式如下：

- 邮箱 fanping.deng@gmail.com
- 博客 blog.csdn.net/innost

## 致谢

本书的顺利出版首先要感谢杨福川编辑的大力支持。另外，要感谢张锡鹏编辑在审稿期间严谨负责的工作。

另外，笔者需要特别感谢现就职的民生银行总行信息科技部。这是笔者第一次供职于一家金融企业。在此工作的这段时间里，我深刻体会到了民生科技人勇于开拓、锐意创新的精神气质，同时也感受到“金融科技为银行创造价值”的深远意义和重大责任。在此，笔者借助本书对相关领导和同事表示衷心的感谢。他们是牛新庄、毛斌、李建兵、林冠峰、李彧、娄丽明、侯佳腾、常薇、王连诚、张梦涵、侯超、金西银、孙升芸、孟凡娇、文静、赖穆彬等。正是你们的鼓励、支持和信任才使我的业余研究成果得以成书。

当然，本书能快速出版，还需要感谢几位功力深厚并热心参与技术审稿的专家。他们是滴滴出行资深研发工程师孙鹏飞和赵旭阳、高通无线半导体技术有限公司资深工程师钟长庚。几位专家在各自领域所体现出来的专业素养和技术水平之高时刻提醒笔者应牢记“路漫漫其修远兮，吾将上下而求索”。另外，其他几位小伙伴罗迪、段启智、高建武、陈永志也对本书的编写提供了不小的帮助。在此一并感谢他们。

最后，一如既往地感谢家人和妻子。另外，特别感谢索菲娅小朋友，正是她不厌其烦地问“爸爸，你为什么看电脑呀”，才使得笔者不得不加快编写速度。最后，感谢所有花费宝贵时间和精力关注本书的读者以及所有在人生和职业道路上曾给予我指导的诸位师长。

邓凡平  
北京

# Contents 目 录

推荐序	
前言	
<b>第 1 章 本书必读</b> ..... 1	
1.1 概述.....1	
1.2 准备环境和工具.....2	
1.2.1 准备源代码.....2	
1.2.2 准备 Source Insight.....2	
1.2.3 准备模拟器和自制系统镜像.....5	
1.2.4 小结.....8	
1.3 本书的内容.....9	
1.4 本书资源下载说明.....12	
<b>第 2 章 深入理解 Class 文件格式</b> ..... 13	
2.1 Class 文件格式总览.....13	
2.2 常量池及相关内容.....14	
2.2.1 常量项的类型和关系.....14	
2.2.2 信息描述规则.....18	
2.2.3 常量池实例剖析.....19	
2.3 field_info 和 method_info.....19	
2.4 access_flags 介绍.....21	
2.5 属性介绍.....22	
2.5.1 属性概貌.....22	
2.5.2 Code 属性.....23	
2.5.3 LineNumberTable 属性.....25	
2.5.4 LocalVariableTable 属性.....26	
2.6 Java 指令码介绍.....27	
2.6.1 指令码和助记符.....27	
2.6.2 如何阅读规范.....28	
2.7 学习路线推荐.....30	
2.8 参考资料.....30	
<b>第 3 章 深入理解 Dex 文件格式</b> ..... 31	
3.1 Dex 文件格式总览.....31	
3.1.1 Dex 和 Class 文件格式的区别.....31	
3.1.2 Dex 文件格式的概貌.....35	
3.2 认识 Dex 文件.....36	
3.2.1 header_item.....36	
3.2.2 string_id_item 等.....37	
3.2.3 class_def.....38	
3.2.4 code_item.....40	
3.3 Dex 指令码介绍.....41	
3.3.1 insns 的组织形式.....41	
3.3.2 指令码描述规则.....42	

3.4 学习路线推荐	44	5.4 操作符重载	106
3.5 参考资料	45	5.4.1 操作符重载的实现方式	107
<b>第4章 深入理解 ELF 文件格式</b>	<b>46</b>	5.4.2 输出和输入操作符重载	108
4.1 概述	46	5.4.3 -> 和 * 操作符重载	110
4.2 ELF 文件格式介绍	46	5.4.4 new 和 delete 操作符重载	111
4.2.1 ELF 文件头结构介绍	47	5.4.5 函数调用运算符重载	117
4.2.2 Linking View 下的 ELF	52	5.5 函数模板与类模板	118
4.2.3 Execution View 下的 ELF	61	5.5.1 函数模板	119
4.2.4 实例分析: 调用动态库中的 函数	65	5.5.2 类模板	122
4.2.5 ELF 总结	72	5.6 lambda 表达式	125
4.3 学习路线推荐	73	5.7 STL 介绍	127
4.4 参考资料	73	5.7.1 string 类	128
<b>第5章 认识 C++11</b>	<b>74</b>	5.7.2 容器类	129
5.1 数据类型	76	5.7.3 算法和函数对象介绍	134
5.1.1 基本内置数据类型介绍	76	5.7.4 智能指针类	138
5.1.2 指针、引用和 void 类型	77	5.7.5 探讨 STL 的学习	140
5.1.3 字符和字符串	81	5.8 其他常用知识	141
5.1.4 数组	82	5.8.1 initializer_list	141
5.2 C++ 源码构成及编译	83	5.8.2 带作用域的 enum	141
5.2.1 头文件示例	83	5.8.3 constexpr	142
5.2.2 源文件示例	85	5.8.4 static_assert	143
5.2.3 编译	86	5.9 参考资料	143
5.3 Class 介绍	88	<b>第6章 编译 dex 字节码为机器码</b>	<b>145</b>
5.3.1 构造、赋值和析构函数	89	6.1 编译器全貌介绍	147
5.3.2 类的派生和继承	97	6.2 编译器前端介绍	150
5.3.3 友元和类的前向声明	103	6.2.1 词法分析和 lex	151
5.3.4 explicit 构造函数	105	6.2.2 语法分析和 yacc	160
5.3.5 C++ 中的 struct	106	6.2.3 语义分析和 IR 生成介绍	171
		6.3 优化器介绍	175
		6.3.1 构造 CFG	176

6.3.2	分析和处理 CFG	181	介绍	302	
6.3.3	数据流分析与 SSA	191	7.4.2	FaultManager 介绍	307
6.3.4	IR 优化	204	7.5	Thread 介绍	311
6.4	ART 中的 IR—HInstruction	222	7.5.1	Startup 函数介绍	311
6.4.1	ART 中的 IR	222	7.5.2	Attach 函数介绍	312
6.4.2	IR 之间的关系	225	7.6	Heap 学习之一	325
6.4.3	ART IR 对象的初始化	231	7.6.1	初识 Heap 中的关键类	326
6.5	寄存器分配	233	7.6.2	Heap 构造函数第一部分	337
6.5.1	LSRA 介绍	235	7.7	JavaVMExt 和 JNIEnvExt	340
6.5.2	LSRA 相关代码介绍	247	7.7.1	JavaVMExt	341
6.6	机器码生成相关代码介绍	271	7.7.2	JNIEnvExt	343
6.6.1	GenerateFrameEntry	272	7.7.3	总结	344
6.6.2	VisitAdd 和 VisitInstance- FieldGet	273	7.8	ClassLinker	345
6.6.3	GenerateSlowPaths	275	7.8.1	关键类介绍	345
6.7	总结	277	7.8.2	ClassLinker 构造函数	352
6.8	参考资料	280	7.8.3	InitFromBootImage	353
			7.8.4	ClassLinker 总结	360
			7.9	总结和阅读指导	362
<b>第 7 章</b>	<b>虚拟机的创建</b>	283	<b>第 8 章</b>	<b>虚拟机的启动</b>	363
7.1	概述	284	8.1	Runtime Start	364
7.1.1	JniInvocation Init 函数介绍	286	8.2	初识 JNI	365
7.1.2	AndroidRuntime startVm 函数 介绍	287	8.2.1	JNI 中的数据类型	365
7.2	Runtime Create 介绍	288	8.2.2	ScopedObjectAccess 等 辅助类	367
7.2.1	Create 函数介绍	288	8.2.3	常用 JNI 函数介绍	369
7.2.2	Init 函数介绍	290	8.3	Jit LoadCompilerLibrary	373
7.3	MemMap 与 OatFileManager	293	8.4	Runtime InitNativeMethods	374
7.3.1	MemMap 介绍	293	8.4.1	JniConstants Init	374
7.3.2	OatFileManager 介绍	298	8.4.2	RegisterRuntimeNative Methods	375
7.4	FaultManager 介绍	302			
7.4.1	信号处理和 SignalAction				

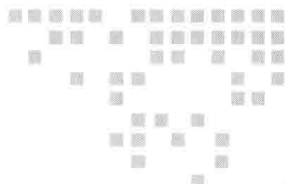
8.4.3	WellKnownClasses Init 和 LastInit .....	376	9.4.4	Setup 代码分析之四 .....	484
8.5	Thread 相关 .....	376	9.5	CompileImage .....	484
8.5.1	Runtime InitThreadGroups .....	377	9.5.1	Compile .....	485
8.5.2	Thread FinishSetup .....	377	9.5.2	ArtCompileDEX .....	496
8.5.3	Runtime StartDaemonThreads .....	380	9.5.3	OptimizingCompiler JniCompile .....	499
8.6	Runtime CreateSystemClassLoader .....	381	9.5.4	OptimizingCompiler Compile .....	527
8.7	类的加载、链接和初始化 .....	383	9.6	OAT 和 ART 文件格式介绍 .....	544
8.7.1	关键类介绍 .....	383	9.6.1	OAT 文件格式 .....	544
8.7.2	SetupClass .....	392	9.6.2	ART 文件格式 .....	550
8.7.3	LoadClass 相关函数 .....	393	9.6.3	oatdump 介绍 .....	554
8.7.4	LinkClass 相关函数 .....	398	9.7	总结 .....	561
8.7.5	DefineClass .....	414	<b>第 10 章 解释执行和 JIT</b> .....	<b>562</b>	
8.7.6	Verify 相关函数 .....	416	10.1	基础知识 .....	564
8.7.7	Initialize 相关函数 .....	424	10.1.1	LinkCode .....	564
8.7.8	ClassLinker 中其他常用函数 .....	426	10.1.2	Runtime ArtMethod .....	566
8.7.9	ClassLoader 介绍 .....	437	10.1.3	栈和参数传递 .....	572
8.8	虚拟机创建和启动关键内容梳理 .....	445	10.2	解释执行 .....	580
<b>第 9 章 深入理解 dex2oat</b> .....	<b>447</b>		10.2.1	art_quick_to_interpreter_	
9.1	概述 .....	448	bridge .....	580	
9.2	ParseArgs 介绍 .....	452	10.2.2	artQuickToInterpreter-	
9.2.1	CompilerOptions 类介绍 .....	453	Bridge .....	582	
9.2.2	ProcessOptions 函数介绍 .....	454	10.2.3	EnterInterpreterFromEntry-	
9.2.3	InsertCompileOptions 函数		Point .....	584	
	介绍 .....	455	10.2.4	调用栈的管理和遍历 .....	593
9.3	OpenFile 介绍 .....	456	10.3	ART 中的 JIT .....	599
9.4	Setup 介绍 .....	458	10.3.1	Jit、JitCodeCache 等 .....	600
9.4.1	Setup 代码分析之一 .....	458	10.3.2	JIT 阈值控制与处理 .....	609
9.4.2	Setup 代码分析之二 .....	464	10.3.3	OSR 的处理 .....	612
9.4.3	Setup 代码分析之三 .....	474	10.4	HDeoptimize 的处理 .....	615
			10.4.1	VisitDeoptimize 相关 .....	616



13.3	BumpPointerSpace 和 RegionSpace.....	725	14.1.1	Mark-Sweep Collection 原理 介绍.....	815	
13.3.1	BumpPointerSpace.....	726	14.1.2	Copying Collection 原理 介绍.....	817	
13.3.2	RegionSpace.....	733	14.1.3	Mark-Compact Collection 原理介绍.....	818	
13.4	DIMallocSpace 和 RosAlloc- Space.....	740	14.1.4	其他概念.....	819	
13.4.1	DIMallocSpace.....	741	14.2	Runtime VisitRoots.....	819	
13.4.2	RosAllocSpace.....	745	14.2.1	关键数据结构.....	821	
13.4.3	rosalloc 介绍.....	748	14.2.2	Thread VisitRoots.....	824	
13.5	LargeObjectMapSpace.....	760	14.3	ART GC 概览.....	827	
13.6	new-instance/array 指令的处理.....	762	14.3.1	关键数据结构.....	827	
13.6.1	设置内存分配器.....	762	14.3.2	ART GC 选项.....	830	
13.6.2	解释执行模式下的处理.....	767	14.3.3	创建回收器和设置回收 策略.....	832	
13.6.3	机器码执行模式下的处理.....	770	14.4	MarkSweep.....	835	
13.6.4	Heap AllocObjectWith- Allocator.....	773	14.4.1	Heap 相关成员变量取值 情况.....	835	
13.7	细观 Space.....	779	14.4.2	MarkSweep 概貌.....	837	
13.7.1	Space 类.....	779	14.4.3	MarkingPhase.....	840	
13.7.2	ContinuousSpace 和 Discon- tinuousSpace 类.....	781	14.4.4	PausePhase.....	848	
13.7.3	MemMapSpace 和 Continuous MemMapAllocSpace 类.....	782	14.4.5	ReclaimPhase.....	851	
13.7.4	MallocSpace 类.....	783	14.4.6	FinishPhase.....	857	
13.8	Heap 学习之二.....	784	14.4.7	PartialMarkSweep.....	857	
13.8.1	Heap 构造函数.....	784	14.4.8	StickyMarkSweep.....	858	
13.8.2	关键类介绍.....	792	14.4.9	Concurrent MarkSweep.....	864	
13.8.3	ObjectVisitReferences.....	806	14.4.10	Parallel GC.....	868	
13.9	总结.....	812	14.4.11	MarkSweep 小结.....	869	
<b>第 14 章 ART 中的 GC.....</b>			813	14.5	ConcurrentCopying.....	870
14.1	GC 基础知识.....	814	14.5.1	InitializePhase.....	871	
			14.5.2	FlipThreadRoots.....	873	

14.5.3	MarkingPhase	881	14.8.2	MarkSweep 中 Reference 对象的处理	903
14.5.4	ReclaimPhase	883	14.8.3	ReferenceProcessor	904
14.5.5	ConcurrentCopying 小结	885	14.8.4	PhantomReference 的处理	912
14.6	MarkCompact	885	14.8.5	finalize 函数的调用	913
14.6.1	MarkingPhase	886	14.8.6	Reference 处理小结	917
14.6.2	ReclaimPhase	889	14.9	Heap 学习之三	917
14.6.3	MarkCompact 小结	891	14.9.1	Heap Trim	917
14.7	SemiSpace	892	14.9.2	CollectGarbageInternal	919
14.7.1	InitializePhase	893	14.9.3	PreZygoteFork	924
14.7.2	MarkingPhase	894	14.9.4	内存碎片的解决	926
14.7.3	SemiSpace 小结	898	14.10	总结	927
14.8	Java Reference 对象的处理	899	14.11	参考资料	928
14.8.1	基础知识	899			





# 本书必读

## 1.1 概述

笔者写书向来是最后才写第一章。此时，全书的主体内容已完全确定，笔者在学习 ART 虚拟机以及编撰本书的过程中所遇到的问题、总结的经验和教训等才可以完整地汇总并分享给各位读者。所以，本章是全书的点睛之笔，为必读章节。并且，我相信随着读者阅读的深入，还会时常回顾本章。

总体来说，本书**并不简单**。其实，从本书的目标——Java 虚拟机也可以想得到，对 Java 应用程序来说，虚拟机就算是操作系统了。哪一本讲操作系统的书会简单呢？

具体到 Android ART 虚拟机来说<sup>⊖</sup>，本书以 Android 7.0 为参考，绝大部分待分析的源代码位于 art 目录中。

- ❑ 包含 C++ 代码 1071 个文件。其中，.cc 文件中包含 236 744 有效代码行（即不算注释及空行），.h 文件中包含 74 710 有效代码行。
- ❑ 包含汇编文件 1704 个文件，覆盖 x86、arm、mips 的 32 位和 64 位 6 个 CPU 平台，有效代码共 19 955 行。

也就是说，我们的 ART 虚拟机是一个有着 30 多万行代码的庞然大物。针对这样一个复杂的系统，要想从一个对它略知一二的初学者成长为一个能品头论足甚至指点江山的熟练者，这一路的学习历程必然不会轻松。

接下来，笔者将介绍阅读本书时必须准备的工具。磨刀不误砍柴工，建议读者先把这些工具准备好之后再开始后面的学习。

---

⊖ 此处的统计以笔者下载的 Android 7.0 源码为目标。