# 《App研发录：架构设计、Cras》

## 图书基本信息

书名：《App研发录：架构设计、Crash分析和竞品技术分析》

13位ISBN编号：9787111516389

出版时间：2015-10-21

作者：包建强

页数：303

# 《App研发录：架构设计、Cras》

## 内容概要

本书是作者多年App开发的经验总结，从App架构的角度，重点总结了Android应用开发中常见的实用技巧和疑难问题解决方法，为打造高质量App提供有价值的实践指导，迅速提升应用开发能力和解决疑难问题的能力。本书涉及的问题有：Android基础建设、网络底层框架设计、缓存、网络流量优化、制定编程规范、模块化拆分、Crash异常的捕获与分析、持续集成、代码混淆、App竞品技术分析、项目管理和团队建设等。本书以“ 问题/解决方案 ”的形式给出疑难问题的解决方案，同时结合示例代码，深入剖析这些实用的编程技巧和模式，旨在帮助移动开发人员和管理人员提高编程效率，改进代码质量，打造高质量的App。

# 《App研发录：架构设计、Cras》

## 作者简介

包建强，毕业于复旦大学数学系。先后在多家互联网公司担任无线部门技术总监。在And roid、iOS、WP等多门无线技术中跋涉过，在App的项目管理上也有多年的实践经验。他是微软2008年MVP。曾经翻译出版《.NET探秘MSIL权威指南》，并有一个坚持写了6年的技术博客:http://jax.cnblOgs.com/。

# 《App研发录：架构设计、Cras》

## 书籍目录

# 《App研发录：架构设计、Cras》

## 精彩短评

1、这些经验很受用
2、APP竞品技术分析的部分值得产品好好研读~~
3、这本书为啥只有7.5？技术书籍不是非得显得牛逼才高分的啊，这样朴素的，作者自己的经验总结不好吗？你们喜欢那种傻了吧唧罗列代码，连他们自己都不知道自己在讲什么的书？不过作者似乎在系统方面比较薄弱。Android的UI太开放，导致现在市面上混乱的UI场面。
4、前两章还有点干货，后面的比较水
5、一本够我嚼两三年的书，大把大把的干货，感谢作者
6、作者的技术水平真是呵呵！！
7、总归会在开发中遇到
8、利用周末完结了，写的很好，见识了正规的团队研发，同时还介绍了一些重要但是平时自己没有用到的东西，谢谢作者。
9、网络基础架构，性能优化不错。竞品分析和crash都是难得的资料
10、一线技术人员的总结与分享, 可以学习与借鉴的很多. 特别是团队组织协调方面.
11、不是一本纯粹的技术书，内容有点杂。
12、实用
13、今年看过的最好的一本Android/移动开发相关的技术图书。
14、一般，书里不少内容有点过时了，而且作者说太多外话了，不太适合初级研发看
15、干货，涉及app崩溃收集、混淆技术、持续集成、团队管理
16、真正讲一个app的研发，而不是单纯的码代码，书很快看完了，但是里面的思想全部掌握并运用还是任重而道远啊。
17、技术牛逼的一本书
18、5天过完的一本书，其中第七章ProGuard的介绍还有点意思，其他章节不做评价。
19、非常不错的书，都是作者亲身开发经验总结出来的干货，三言两语点出关键。比起让人云里雾里不知所云的书不知道要高到哪里去了！虽然不是很厚，但是对于现阶段的初级Android程序员的自己太合适不过了。
20、还不错，相比技术对于移动开发的一些经验之谈值得借鉴
21、花了一周看完了，讲的比较系统。内容不很多，很多时候都是针对于作者自己项目中的问题讲解一番，不知是是否因为自己水平不够，并没有很多共鸣。
22、就一本安卓开发的书，怎么名字敢叫App研发录。靠，我是iOS开发者啊，看安卓有个毛用
23、干货不少，proguard那一章不错，毕竟以前没有哪本书系统地总结过。竞品分析那一章也不错，涨姿势了，知识面宽了。作者用的是Eclipse,没有对Android Studio和Gradle的讲解。
24、详尽地介绍了大公司的流程，还有一些最佳实践和坑等。
25、android
26、比较适合开发组长大概浏览，了解概念及一些技术点（插件化、本地测试、Native&H5交互、多渠道打包、开发后门），需要再自己后续研究。包括团队管理方向、新人文档、编码规范等，都是比较适合组长的知识点。Crash基本没看，看了也记不住，遇到时再根据堆栈信息去找bug就行，没必要现在把这些东西看一遍记下来
27、还不错，看这本书就像你和一个工作多年的技术经理在聊天，当然中间有一些作者的crash总结，可以没事拿来翻翻看看自己的app有没有同样的crash。看得出来作者有着非常丰富的带技术团队的经验，而且平时善于总结、归纳问题，确实对于我的开发工作有很多值得借鉴的，不过书中一些技巧、开源库不是最新最好的，或者不是最棒的解决办法，可能是作者很久之前写的吧
28、老司机经验之谈，新人可以管中窥豹，司机同行可以借机吐槽共鸣
29、android进阶很有用
30、1. 本书可以让你少趟一些坑。
2. 然而书中的“ 我觉得 ”有点多啊，看着信心不足。
3. 可见写博客是多么重要，那么多面试者总是说自己记了多少笔记，鬼才信呢~
31、“ 落地 ”，是阅读此书的最大感受，大到技术架构、竞品分析、发布流程，小到每个Crash、员工

座位安排、开展技术分享、简历筛选，作者总在寻求落地的解决方案，细细阅读，你不觉得这种方案有多么高大上，不觉得背后有多么华丽的理论支撑，但是你会觉得，一切都是踏踏实实、真真切切可落地实践的，为工作带来不少的实践指导。

建议移动开发从业者都阅读一下本书，如果你是一名程序员，有必要了解下你的leader背后要考虑的大局，以及做出某些决策的原因，这能帮助你更好的为团队分忧，承担更多的责任，获得更多的晋升空间。如果你是一名技术管理者，反思一下你的决策是否能落地，是否只是空谈，或者你的决策与你想达到的目标其实是两码事...完整如下
http://www.jianshu.com/p/ea48332c82af
32、书名是《app研发录》，不过感觉书的定位好像很模糊，既有普通Android工程师需要的技术内容，又有一些类似于产品经理所关心的开发流程之类的内容，除了前面讲的一个网络框架的设计让人有印象之外，还有就是一章专门用来讲crash，我也是醉了，所以感觉整本书并没有很多干货，有些内容也是浅尝即止，有人把它跟《Android开发艺术探索》相提并论，我觉得是不够资格的。
33、个人不推荐购买，我见过的书里面出奇的居然讲业务，到稍微想看的知识点往往几句话带过，完全是浪费时间
34、强烈推荐，一口气看完，如果之前看过这本书，项目会少踩很多坑
35、还OK

# 《App研发录：架构设计、Cras》

## 精彩书评

1、之所以上述所说，是因为看这本书的时候，总感觉怪怪的。因为在地铁上看完的，作者书中基本都是他自己工作中遇到的问题和坑，虽说这样会让人感觉找到了解决方案，可以再进行深入的研究，可是某些地方介绍的有点片面，仅仅是引用部分博客就以偏概全了。还有可能是涉及的内容大部分都是我自己已经踩过的坑，所以觉得学到的东西不太多。再说说值得一看的地方吧，首先也如前面提到的，书中内容基本都是作者工作之谈，所以有很实用的内容，推荐阅读章节：1. App竞品技术分析个人认为这是本书的精华，很少看到有人愿意这样详尽的介绍自己的“机密”经验，感谢作者的无私分享。2. 五、六、七章的异常和ProGuard介绍很详细，网上都是非常琐碎的介绍，推荐新人看看这部分，尤其是ProGuard，虽然现在第三方已经给出了完善的解决方案，帮我们做了这部分事情，不过了解最基本的原理才能学得更透彻。3. 项目、团队管理部分这部分是时间的积累才能收获的，提前学习了解以后我们必须经历的路没什么不好的。先大概就总结这么多吧。

# 《App研发录：架构设计、Cras》

## 章节试读

1、《App研发录：架构设计、Crash分析和竞品技术分析》的笔记-第71页

既然GlobalVariables是单例类，那就应该增加synchronized同步块防止多线程并发访问，原文没有考虑到这一点。

```
public class GlobalVariables implements Serializable, Cloneable {
  private static GlobalVariables instance;

  private GlobalVariables() {
  }

  public static GlobalVariables getInstance() {
    if (instance == null) {
      synchronized (GlobalVariables.this) {
        Object object = Utils.restoreObject(CACHE_DIR + TAG);
        if (object == null) {
          object = new GlobalVariables();
          Utils.saveObject(CACHE_DIR + TAG);
          instace = object;
        }
      }
    }
    return instance;
  }
}
```

这样写才是真正的单例类。

2、《App研发录：架构设计、Crash分析和竞品技术分析》的笔记-第96页

5.2和5.3章节就是作者自己在自high，这种分析数据库crash信息的基础sql语言毫无意义，也没技术含量，还浪费两个章节的篇幅，无语。

也就是5.1节大概提了一下，在Application的onCreate函数里注册UncaughtExceptionHandler还有些意义。

3、《App研发录：架构设计、Crash分析和竞品技术分析》的笔记-第159页

作者提到：对于多进程App而言，还是需要ContentProvider。其实，ContentProvider也是CS架构，只是Binder封装了SQLiteOpenHelper的一系列操作，将db封装成单例，那ContentProvider操作数据库也是单例的。

4、《App研发录：架构设计、Crash分析和竞品技术分析》的笔记-第114页

指出两处错误：

1. Comparator：我认为应该是基于快排的产物。
2. compare方法里面，p1和p2应该是d1和d2

还有，其实比较算法可以简化啊，这种简单的升序或者降序，直接：
return d1-d2;
不就可以了。

5、《App研发录：架构设计、Crash分析和竞品技术分析》的笔记-第232页

不太清楚竞品分析里干嘛揉入大量IOS的介绍，而且还都是最浅显的基本使用，浪费篇幅，看起来也没啥兴趣。

6、《App研发录：架构设计、Crash分析和竞品技术分析》的笔记-第89页

12条：为了节省内存，请使用ArrayList<自定义实体>,而不是HashMap.理由是：HashMap占用的内存多。

不太清除作者为什么会有这种结论，或者作者是否研究过HashMap的源码。

HashMap采用的数据结构是：数组+链表。之所以HashMap的查找时间复杂度是O(1)，是因为数组的下标是key的hash值，所以查找速率快。但是，不意味着数据多，HashMap分配的内存就要大。

7、《App研发录：架构设计、Crash分析和竞品技术分析》的笔记-第174页

指出一个ProGuard使用的错误，保留Parcelable序列化的类不被混淆,应该使用
-keepclassmembers class * implements android.os.Parcelable {
static android.os.Parcelable$Creator CREATOR;
}
而不是用-keep

8、《App研发录：架构设计、Crash分析和竞品技术分析》的笔记-第94页

1. 比较无语，第六章已经介绍了不能在子线程中更新UI，作者还自己搞了个线程,在线程里显示了一个Toast，真bug。

2. 首先实现一个UncaughtExceptionHandler,应该是单例类,示例代码如下：
```
public class CrashHandler implements UncaughtExceptionHandler {
  /** CrashHandler实例 */
  private static CrashHanlder mInstance;
  /** 系统默认的UncaughtException处理类 */
  private Thread.UncaughtExceptionHandler mDefaultHandler;

  private CrashHandler(Context context) {
    // 保留系统默认处理类,便于自己不处理时进行调用
    mDefaultHandler = Thread.getDefaultUncaughtExceptionHandler();
    // 设置自己为当前线程的未捕获异常处理类
    Thread.setDefaultUncaughtExceptionHandler(this);
  }
  public static CrashHandler getInstace(Context context) {
    if (mInstace == null) {
```

```
        synchronized(CrashHandler.class) {
          if (mInstace == null)
            mInstace = new CrashHandler(context);
        }
      }
    }

    @Override
    public void uncaughtException(Thread thread, Throwable ex) {
    }
}
```

在Application中注册:

```
public class App extends Application{
      @Override
      public void onCreate() {
        super.onCreate();
        //注册crashHandler
        CrashHandler crashHandler = CrashHandler.getInstance(getApplicationContext());
      }
}
```

9、《App研发录：架构设计、Crash分析和竞品技术分析》的笔记-第151页

作者说这一章花费了巨大的精力，但是其实很多Exception都是一笔带过，并没有深入讲解。

例如APK动态注册BroadcastReceiver，为什么会出现SecurityException?

10、《App研发录：架构设计、Crash分析和竞品技术分析》的笔记-开篇题记

APP研发录读书学习，感谢豆瓣读书笔记开发者.

# 《App研发录：架构设计、Cras》

## 版权说明