Atitit 提升简洁代码的方法 总结

动态编译

静态导入。。。

易变业务使用脚本语言编写

* + 之前工作中有个这样的需求，10几种类型根据公式计算不同生失效时间，公式易变，当时想用js，找了好久都没找到好用的，后来放弃了；
  + 如果能找到适合的脚本engine，那就用吧；

使用构造代码块精炼程序

* 直接{}括起来的代码片段，会在每个构造函数中调用；
* 建议不要用；

让多重继承称为现实

* 外部类继承一个，内部类再继承另一个；

子列表只是原列表的一个视图

* subList返回子列表；
* 所有对subList产生对象的操作作用于原list；

不推荐使用binarySerach对列表进行检索

* 二分查找建立在基本有序的情况才行；
* 实现RandomAccess或数据量小于5K进行二分查找，否则顺序查找；

集合运算时使用更优雅的方式

* 并集addAll
* 交集retainAll
* 差集removeAll

使用suffle打乱列表

* collections.shuffle();

采用异常链传递异常

* + 异常需要封装和传递，不同视角展示不同提示；

受检异常尽可能转为非受检异常

编写高质量代码:改善Java程序的151个建议(第1章:JAVA开发中通用的方法和准则\_\_\_建议6~10)

阅读目录

建议6：覆写变长方法也循规蹈矩

建议7：警惕自增的陷阱

建议8：不要让旧语法困扰你

建议9：少用静态导入

建议10：不要在本类中覆盖静态导入的变量和方法

(378条消息) 编写高质量代码--改善Java程序的151个建议精要\_steveguoshao的专栏-CSDN博客

是把秦小波所著《编写高质量代码--改善Java程序的151个建议》一书的目录贴出来，相当于整个大纲，就好比武功心法，版权归属秦小波。

1. Java开发中通用的方法和准则

2. 基本类型

2.1 使用偶判断，不用奇判断

3. 类、对象及方法

4. 字符串

5. 数组和集合

6. 枚举和注解

7. 泛型和反射

8. 异常

8.1 提倡异常封装

8.2 采用异常链传递异常

8.3 受检异常尽可能转化为非受检异常

8.4 不要在finally块中处理返回值

8.5 使用Throwable获取栈信息

8.6 异常只为异常服务

8.7 多使用异常，把性能问题放一边

9. 多线程和并发

10. 性能和效率

11. 开源世界

11.1 大胆采用开源工具

11.2 推荐使用Guava扩展工具包

11.3 Apache拓展包

11.4 推荐使用Joda日期时间拓展包

11.5 可以选择多种Collections拓展

12. 思想为源

12.1 提倡良好的代码风格

12.2 不要完全依赖单元测试来发现问题

12.3 让注释正确、清晰、简洁