/\*\*

\* Java编译器

\*

\* java编译器是一个不确定的术语，可指代javac、GCJ、JIT

\*

\* javac把用java编写的程序编译成为class文件

\* GCJ把用java编写的程序编译成为本地机器代码

\* JIT运行时把jvm指令翻译成本地机器代码，hotspot中提供两种实现（我们把它称为C1与C2

\*

\* hotspot中同时存在解释器与编译器（JIT）

\*

\* 解释器存在的理由：

\* 1、占用内存小，这在移动设备中很有用

\* 2、启动速度快（省去了编译为本地代码及优化本地代码的时间），这对对启动时间有要求的应用很有用

\* 3、编译器极短优化时的逃生后门（极短优化可能导致优化失败，这是退回去采用解释执行即可）

\* 4、为编译器收集信息

\* 编译器存在的理由：

\* 1、提高代码的执行速度（解释执行速度慢）

\*

\* 客户端模式与服务器端模式存在两种编译器的理由

\* 为了编译速度做了不同程度的优化

\* 客户端模式只做了简单的优化

\* 服务器端模式做了很多优化，但是耗时

\*

\* 编译对象

\* 1、调用次数较多的方法

\* 2、执行次数较多的循环体

\*

\* 热点探测：

\* 1、采样探测

\* 2、技术探测（hotspot中采用的方法）

\*

\* 编译触发条件：

\* 客户端模式：1500

\* 服务器端模式：10000

\*

\* 优化技术：

\* 一大堆的优化技术，标胶主要的CHA（类型层次关系分析）EA(逃逸分析)这两种技术是许多其他优化技术的基础

\*

\*

\*/