



课程项目1

- 感谢唐浩、朱琪豪对开发包和评测平台的贡献
- 实现一个Java上的指针分析系统
- 排行榜给分：
 - 无法在测试程序上正常运行的不合格
 - 如：超时（3分钟），崩溃
 - 在测试程序上能输出结果，但结果不健壮(unsound)，1分
 - 结果健壮，根据精度分数在1-2之间
 - 得分为在所有测试用例上得分只和（数量由助教公布）
 - 代码提交作为评分参考
 - 提交一个自己编写的测试样例，包括代码和标准输出
- 最终给分：根据所有测试样例（含预定义的和大家提交的）上运行结果精度评分。
 - 公开的两个测试用例为60分，剩余测试用例为40分。
- 组队完成：
 - 2-3名同学一个小组
 - 组内贡献不均等的，请在提交的时候说明



程序样例

输入程序:

```
public static void main(String[] args) {  
    BenchmarkN.alloc(1); //标记分配点, 没有标记的默认编号为0  
    A a = new A();  
    BenchmarkN.alloc(2);  
    A b = new A();  
    BenchmarkN.alloc(3);  
    A c = new A();  
    if (args.length>1) a=b;  
    BenchmarkN.test(1, a); //标记测试点编号和被测变量  
    BenchmarkN.test(2, c);  
}
```

输出:

1: 1 2

2: 3

每行一个测试点, 以测试点编号开头。

冒号后面是可能的分配点, 多个分配点以空格分割



开发平台

- Java上常见静态分析平台（自学）：
 - Soot（推荐）
 - WALA
 - Chord
- 部分平台已经自带指针分析，要求
 - 不能直接调用平台的指针分析模块
 - 可以使用平台提供的其他支撑，比如数据流分析框架，控制流图构建，Java语言化简等



时间节点和提交内容

- 组队报给助教（11月7日）
- 代码提交（11月28日）
 - Readme.pdf: A4两页以内，描述算法的主要设计思想，小组成员姓名、学号和分工
 - 项目源代码（可直接 build）
 - pta-1.0-SNAPSHOT-jar-with-dependencies.jar: 编译好的 jar 文件
- 现场报告（11月30日）
 - 各组交流所采用的算法，预计每组10-15分钟左右



运行环境

- Java 1.8
- Maven \geq 3.6.0



测试方法

- `java -jar pta-1.0-SNAPSHOT-jar-with-dependencies.jar [src] [SomePackage.Main]`
 - `[src]`: 程序源码的根目录，同时包括java文件和class文件，并且包括JDK1.7版本的rt.jar和jce.jar（SOOT需要，开发包已经包括）
 - `[SomePackage.Main]`: 包含main函数的类名
- 输出写到result.txt
- 细节可参见readme.md



导出可执行的jar包

- Jar包需要包括所有dependency
- 可直接在项目根目录下 `mvn package`, 生成jar包为 `target/pta-1.0-SNAPSHOT-jar-with-dependencies.jar`



测试程序

- ~~Ubuntu16.04 JDK1.8~~ 实时给出程序运行结果
- 请使用队长的学号提交
- 测试一次程序所消耗的时间大概为3-4分钟，请不要一直连续提交评测任务。
- 提交网站： <https://162.105.88.22:8082>
- 时间：ddl提交样例测试程序之后，会更新排行榜上的对应排名，加入新编写的测试样例。



下周的提前准备

- 安装 java8, maven3
- 安装 java12, vscode, vscode插件Extension Pack for Java (本行可选, 用于演示vscode环境下的开发)