软件分析技术 2018 课程大作业一

小组成员： 黄杨洋 张博洋 刘渊强

一、目标：实现一个Java上的指针分析系统

- 基于 **SOOT** 框架

二、代码目录及使用环境

- `src/main/java`为分析程序源代码目录
- `code/benchmark`中保存了评测类及被测类的实现
- `code/test`中保存了测试代码（被分析的代码）
- 请使用 **Intellij** 打开 project，并将`code`和`src/main/java`目录标为源代码目录
- 命令行参数：最后一个类名被视为`main()`方法所在类，修改该参数即可分析不同的测试代码；示例：`-w -cp out/production/pointerAnalysis -pp test.Hello`

三、算法主要设计思想及代码实现

1. `src/main/java`中基于 **SOOT** 框架实现了一个 **Anderson** 风格的指针分析算法，即基于约束(constraint-based)或基于子集(subset-based)的指针分析方法
2. 由于java中的指针均以引用形式存在，故而只存在基本约束（`a = new A()`)和简单约束（`a = b`）这两种约束类型
3. `src/main/java/RunPointerAnalysis.java`为分析入口，`src/main/java/WholeProgramTransformer.java`为基于SOOT框架实现的转换函数，`src/main/java/AnswerPrinter.java`作用为输出分析结果，`src/main/java/AndersonAnalysis.java`为实现的分析算法
4. 分析难点： 流敏感分析、域敏感分析、上下文敏感分析、数组分析、递归分析
5. 数据流分析框架：
    1. 正向分析
    2. 半格元素：一个字典集合，每个键代表一个变量，值为该变量可能指向的内存位置
    3. 交汇操作：并
    4. 变换函数：
        1. Benchmark.alloc(id): 获取id作为下一个new语句的内存位置
        2. New语句
        3. 赋值语句：存在两种情况，变量赋值`a = b`和域赋值`a.f = b.f`
            - kill 集合是左值对应的内存位置，gen 集合是右值对应的内存位置
            - 若右值是域 `base.field`，此时的 gen：先求出 base 所指向的内存位置（如 [1,2]），查询当前集合中是否有 1.f,2.f 的位置，若存在则加入 gen 中
            - 若左值是域`base.field`，此时的 kill：同样先求出 base 所指向的内存位置，查询当前集合中是否有该域的位置，若存在则加入 kill 中

4. 函数调用语句（过程间）：
    - 新建一个一个 AndersonAnalysis 类的实例，传入被调用函数和目前堆上的分析结果进行分析
    - 分析完毕后，把分析结果中堆上的结果和对返回值分析的结果取出，放入当前实例的分析结果中
    - 只分析了一次被调函数，无法处理递归的情况
5. 数组
6. 递归