**蚂蚁爬杆系统——静态分析报告**

10165101285 丁鹿新

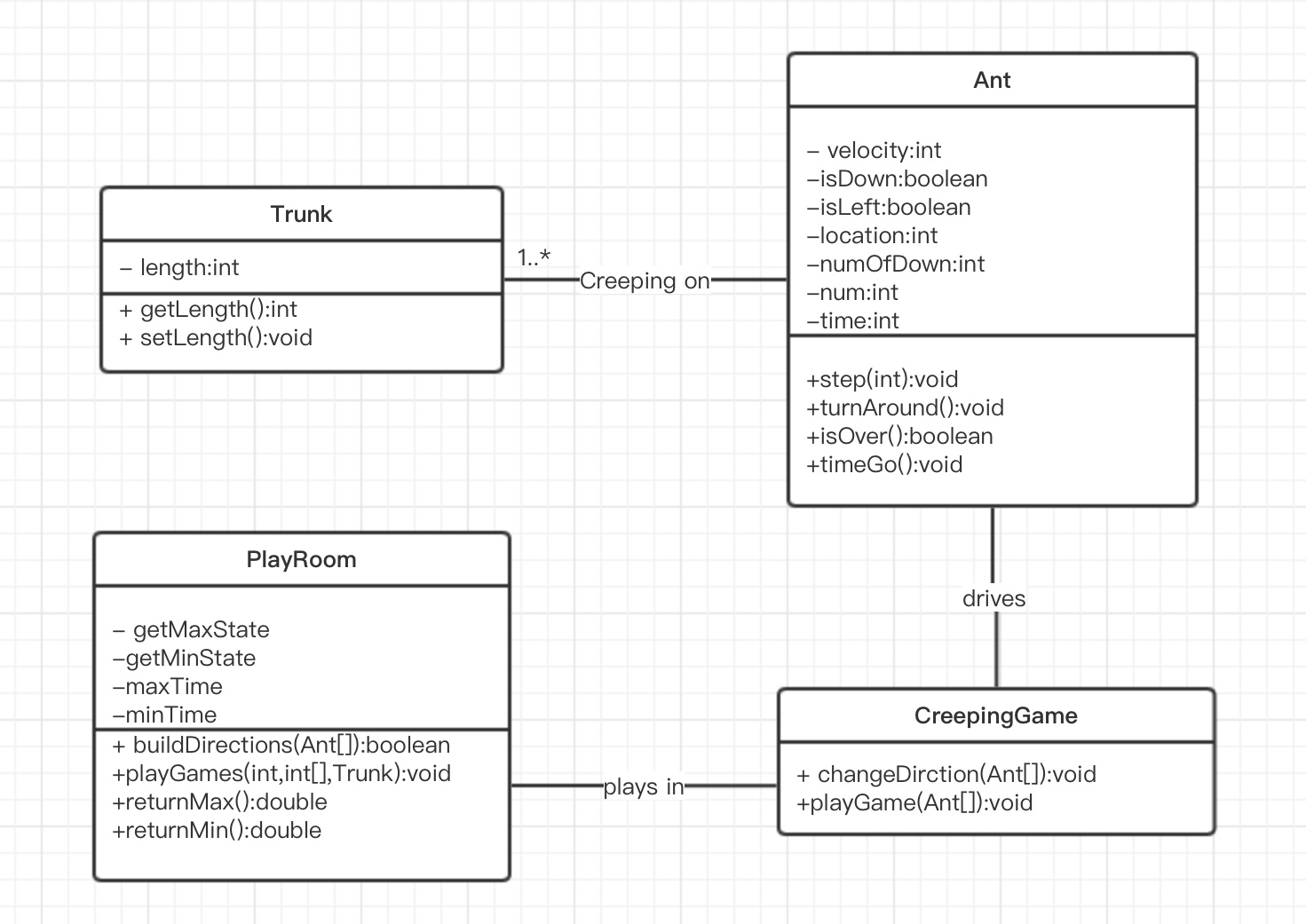
**系统描述：**

有一根300 厘米的细木杆，在第30 厘米、80 厘米、110 厘米、160 厘米、250厘米这**5个位置**上各有一只蚂蚁。木杆很细，不能同时通过两只蚂蚁。开始时，蚂蚁的头朝左还是朝右是任意的，它们只会朝前走或调头，但不会后退。当任意两只蚂蚁碰头时，两只蚂蚁会同时调头朝相反方向走。假设蚂蚁们每秒钟可以走5厘米的距离。

请编写一个程序，用面对对象的方法，计算各种可能情形中所有蚂蚁都离开木杆的最小时间和最大时间。

(最小时间为28，最大时间为54)

**类图：**



说明：

Ant为蚂蚁类，有基本的属性和方法；

Trunk为木杆类，有基本的属性和方法；

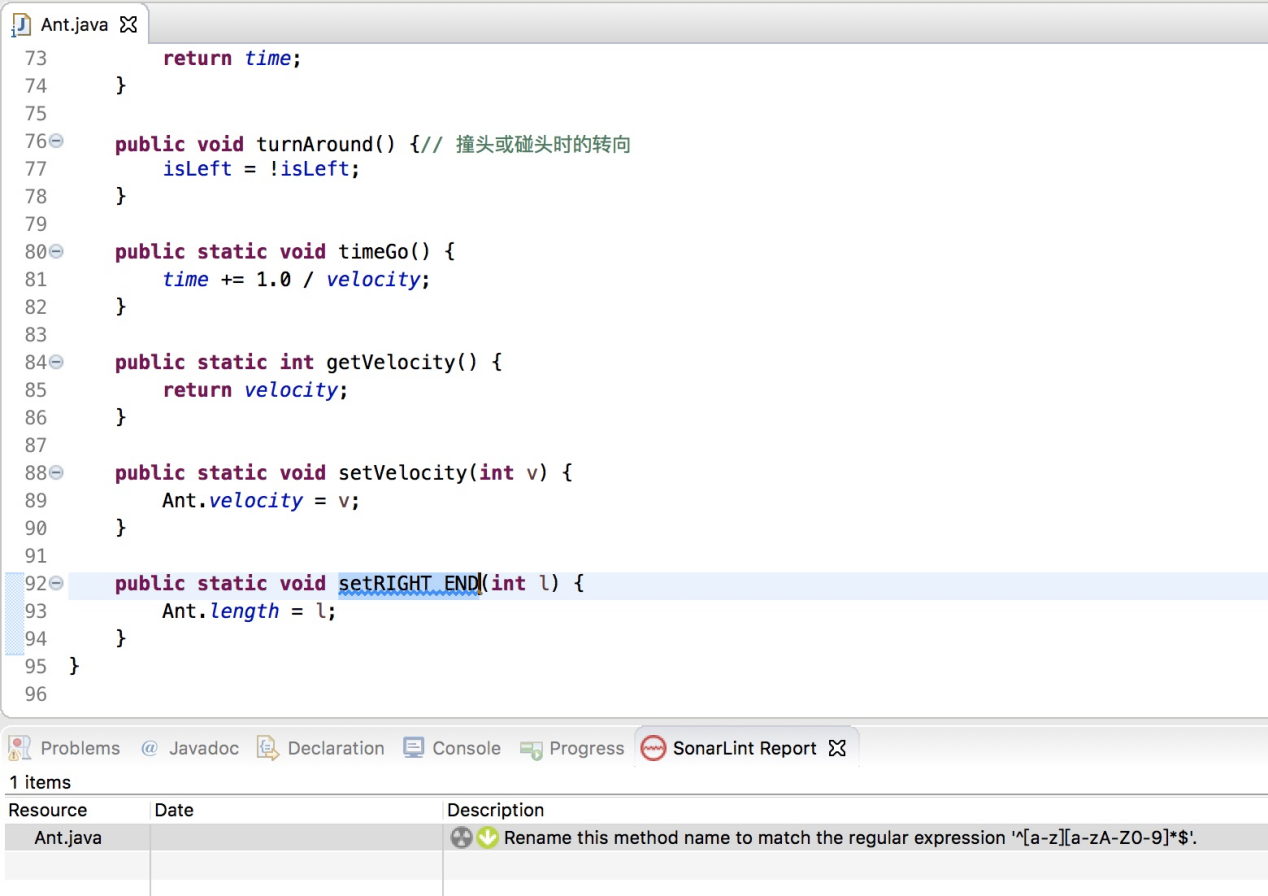
CreepingGame类来推进蚂蚁的爬行过程；

PlayRoom类来实现游戏不同的初始化设定以及程序的入口。

**静态分析：**

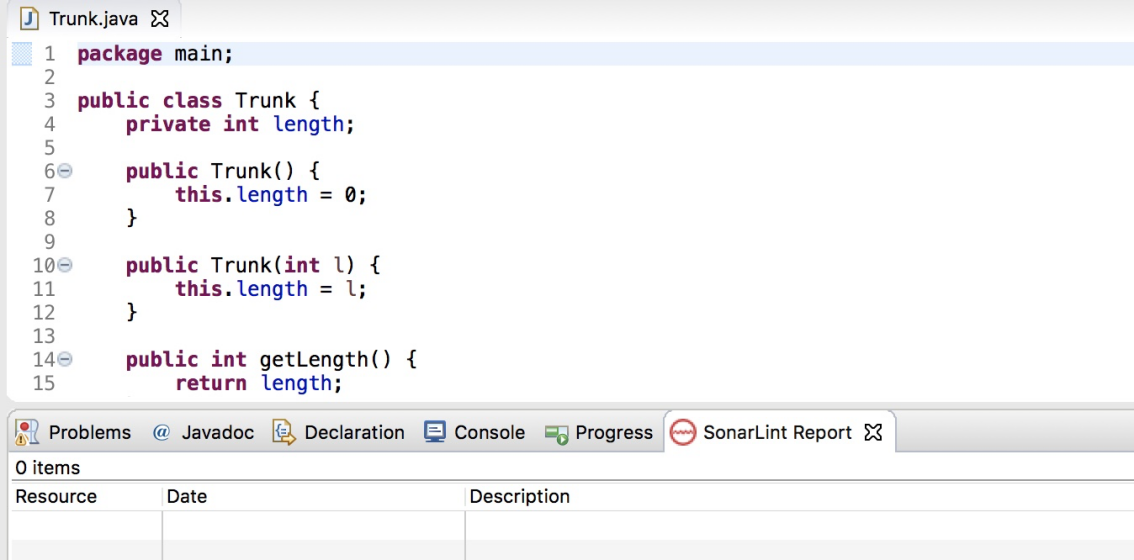
* Ant：

有一个命名不太规范，但无影响，无需改动。



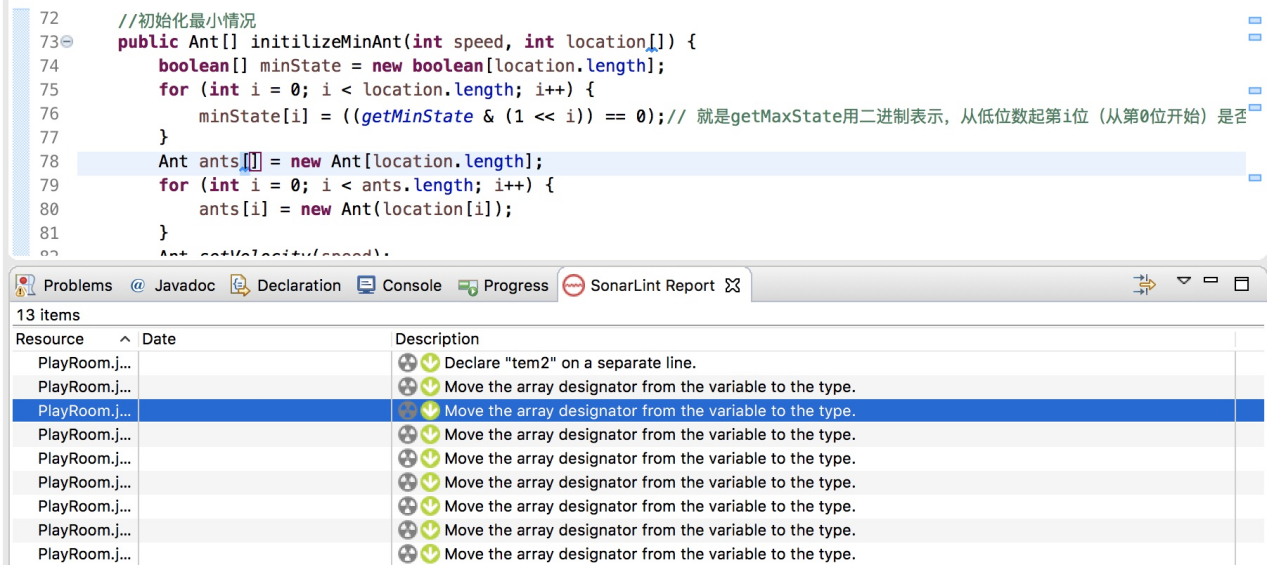
* Trunk：

这个类的较为简单，也没有需要改的地方。

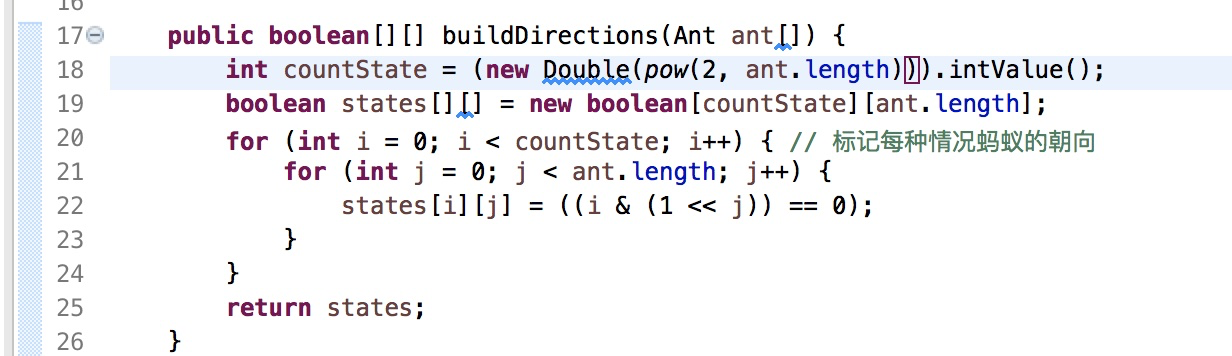


* PlayRoom：

1. 首先是一些小细节，根据提示，将所有的数组指示符从变量移动到类型。以及一些方法名称重新按照命名规范进行改正。

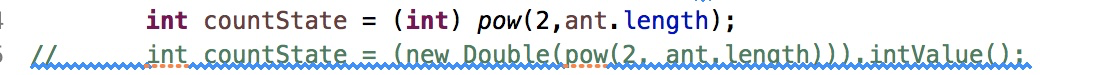


1. 提示去除这个double结构：





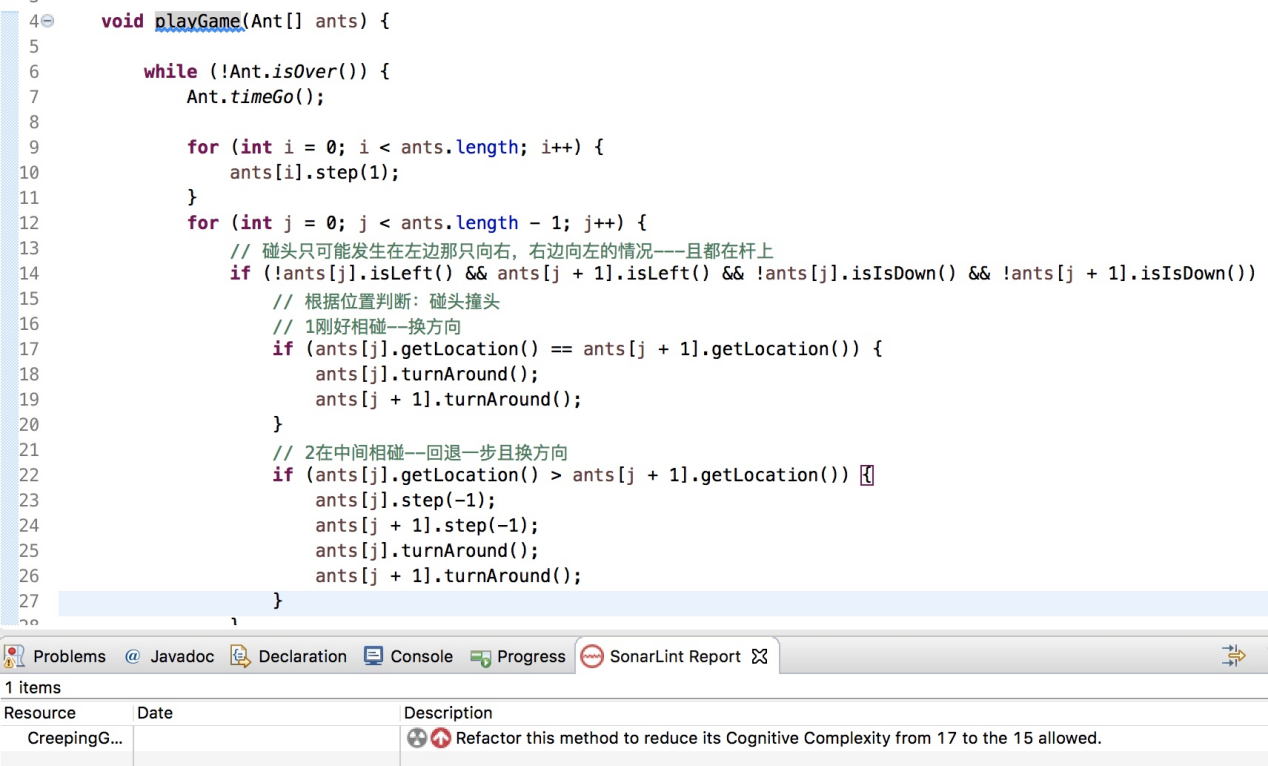
更改这一语句为更简单的写法，改为：



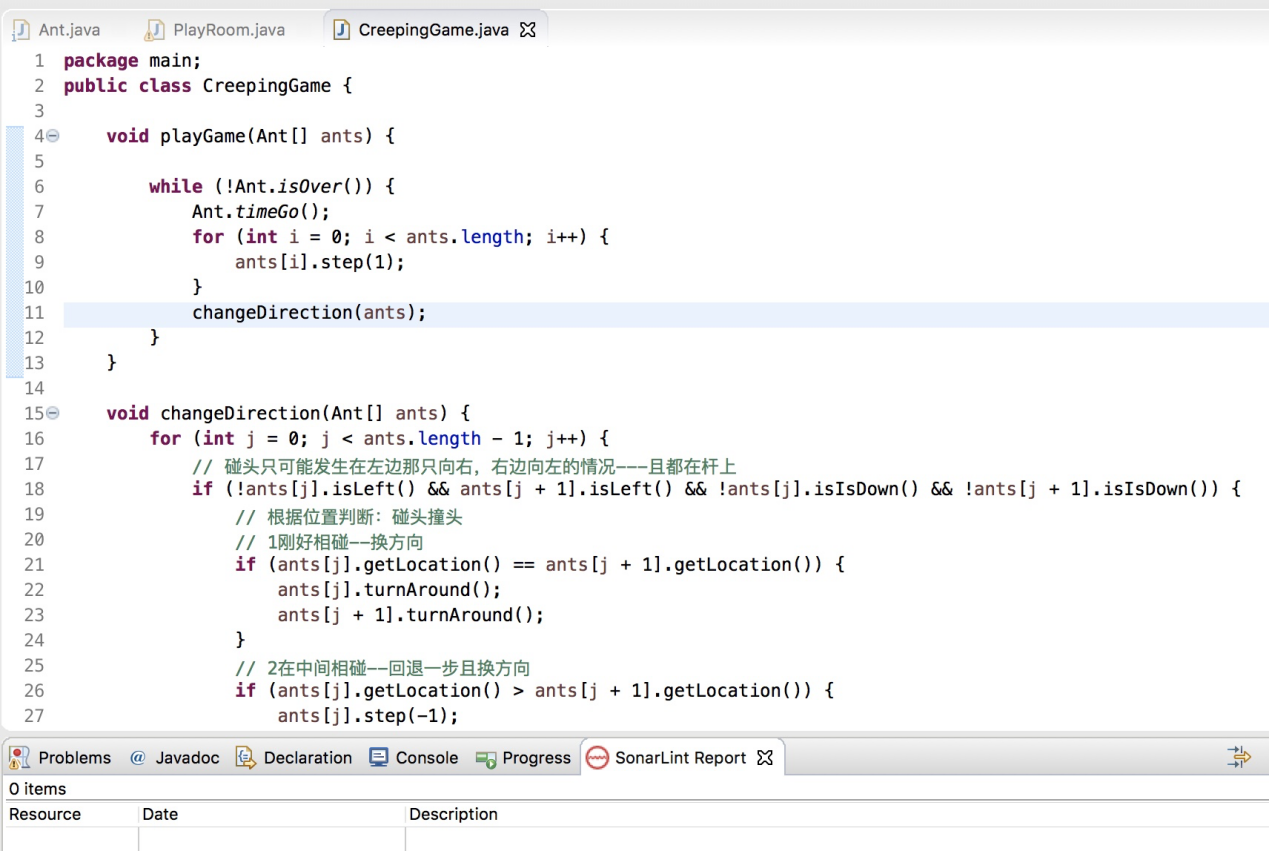
至此，这个类的所有问题均已修正。

* CreepingGame:

这是蚂蚁爬行直至离开杆（如果中途相撞则掉头）的核心逻辑代码，分析指出该代码认知复杂度过大，即不容易理解。



因此将判断是否碰撞改变方向的代码单独拿出来，写成一个函数。修改后：



**测试代码：**

该系统是要计算出5只蚂蚁，在所有的朝向情况下的用时，找出最大最小时间。而我们通过人工计算得最大时间为54，最小时间为28。

因此，我们针对PlayRoom中的returnMax() 和returnMin() 进行测试，判断得出的结果是否正确即可，代码和运行结果如下：

