Exercise 1. Write an aspect to record (probably in a map) all the objects that have a field that is read in the course of void Drawable.draw(...). You should also take into account field reads that happen in methods that are called (possibly indirectly) by draw. Also define advice that detects any writes to fields of these objects, and calls draw again (on the correct instance of Drawable) whenever that happens. Is your solution likely to work in practice?

Exercise 2. Class P implements a complex data structure. There is an invariant that must always be satisfied, and there is a method to check the invariant in P called boolean checkP(). Write an aspect to invoke checkP at the end of each public method, and print the name of such a method if a violation is detected. (Hint: you need to use thisJoinPoint). Carefully consider the possibility of non-termination (for instance when checkP calls other public methods), and how to avoid it.

Exercise 3. Given is a class C that executes commands, and each command is a method in C whose name starts with the characters cmd. C furthermore has a reset method that resets the internal state of C. Show how to augment C with a replay facility: whenever replay() is called on an instance of C, it resets C, and it replays all the commands that happened since the previous reset.