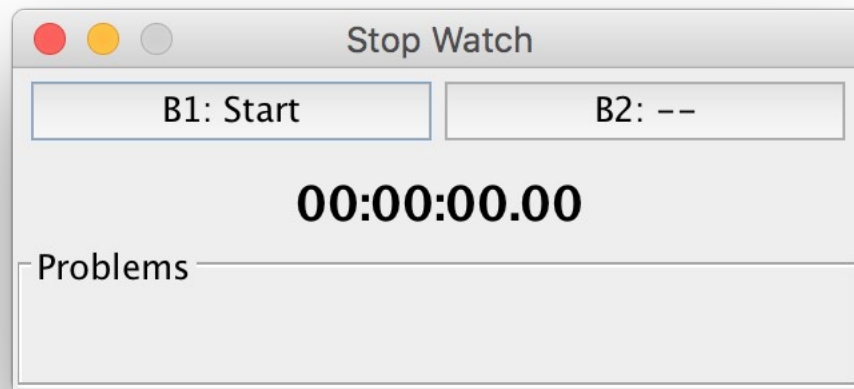


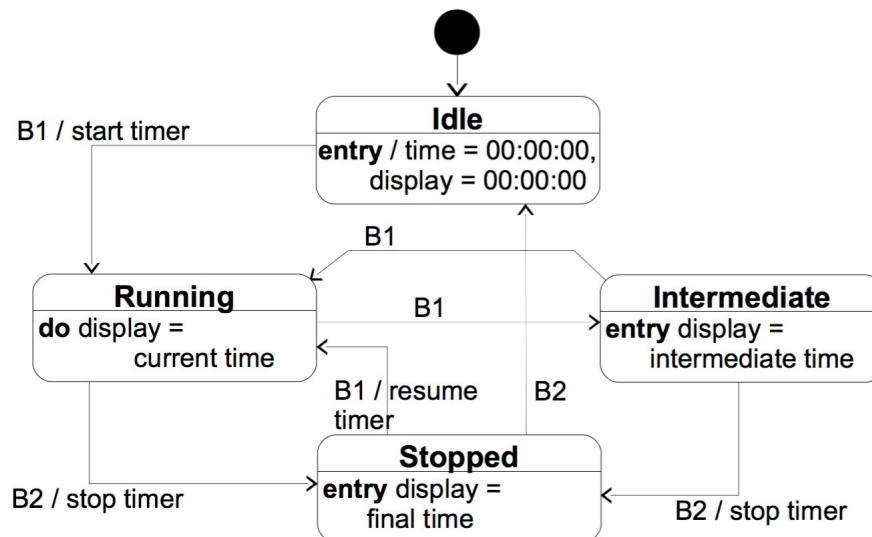
Exercise on State

Introduction

State-based behavior must be implemented in many applications. Consider for example this simulation of a (mechanical) hand-hold stop watch:



This application shall simulate a stop watch. The state behavior of the stop watch is given in the following UML state diagramm:

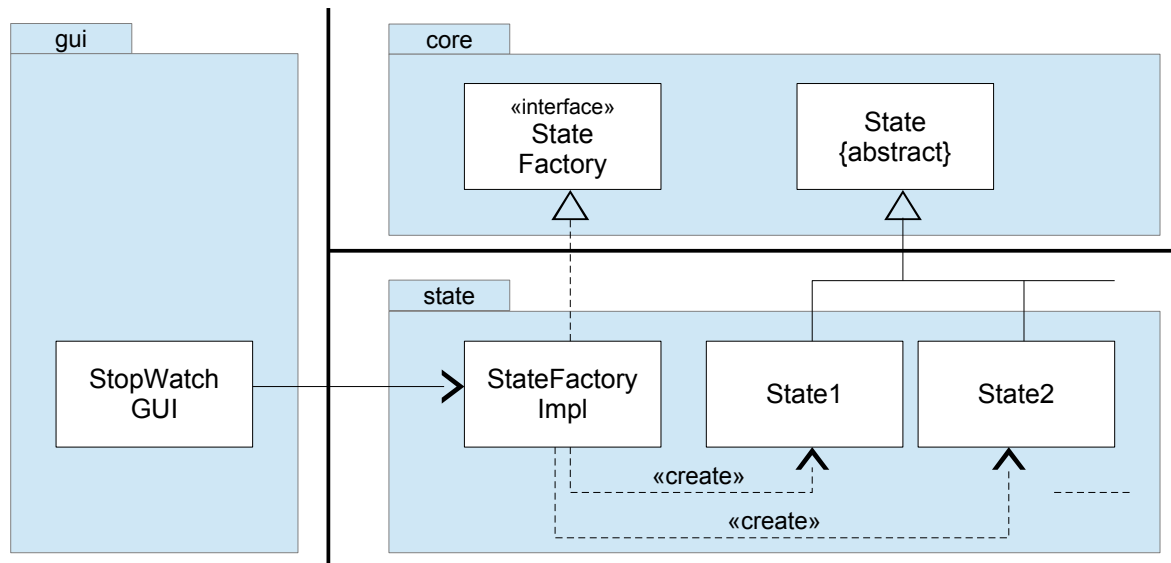


Sketch of Classes of an Application

A small GUI demo application is provided which uses a simplified framework for (simplified) stop watches. The sketch of the classes is given below. Your task is to complete the state classes.

Make sure that your classes can be instantiated by the given factory class `StateFactoryImpl`. Make also sure that the Java property file `statefactory.properties` contains the fully-qualified class names of your State classes.

The structure of the complete application is (only most relevant Java classes, interfaces, and packages are shown):



Tasks

Provide your concrete State classes. Replace class `DummyState` (not shown above) with your classes. Think of how you could test your classes in isolation, e.g., without any GUI classes?

Caveat: No JUnit test classes are provided. Provide your own!

Notice: The type of this project is Maven, i.e., this is a Maven project. Thus, import it as a Maven project into your IDE (Eclipse, IntelliJ, ...).

Biel, December 13, 2015