SWEA Iteration 5: Test Stub and Abstract Factory

<<Group name>>
Computer Science, University of Aarhus
8200 Århus N, Denmark
<<Names>>

<<Date>>

1 EpsilonStone

1.1 Design and UML

[Short argumentation for your EpsilonStone design: Refer to the UML diagram, and emphasize how indirect input are encapsulated; which delegates are the Test Doubles and the real production implementation.]

[INCLUDE UML DIAGRAM HERE - A GOOD PICTURE OF HANDRAWN IS OK.]

1.2 JUnit test cases

[Include screenshot/contents of the JUnit tests for EpsilonStone showing the used test doubles as well as the testing of behavior.]

[Include the filename of your Java test case source file above, so your TA can quickly find it in your source code.]

2 AbstractFactory

2.1 Design and UML

[Include a UML diagram that shows the design of HotStone with emphasis on the use of Abstract Factory. Please, do not draw all the association lines to all concrete Products/delegates as this will make the diagram a complete mess. (A good quality picture of a hand-drawn UML diagram is OK.)]

2.2 Configuring ZetaStone

[Include screenshot/contents of the ConcreteFactory that configures HotStone for the ZetaStone variant.]

[Include the filename of your Java ZetaStone factory source file above, so your TA can quickly find it in your source code.]

3 Backlog

The following features and requirements are still not implemented in our HotStone software:

- …
- …