SWEA Iteration 4: Code Quality and State Pattern

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

<<Date>>

1 Clean Code Method One

1.1 Before Code

[Include a screenshot/listing of your method(s) BEFORE cleaning up]

1.2 Analysis

[Insert the Clean Code Template here, fill in the 'is OK' column with 'yes/no/not applicable' and fill in the 'Explain' column for the BEFORE code - outlining shortly why the property is OK or not; OR a reference to a longer discussion below]

[Provide further analysis/argumentation in case the 'Explain' column is too small to provide a sufficient analysis]

1.3 After Code

[Include a screenshot/listing of your method AFTER cleaning up]

1.4 Conclusion

[Insert the Clean Code Template here, fill in the 'is OK' column with 'yes/no/not applicable' and fill in the 'Explain' column for the AFTER code]

[Provide further analysis/argumentation in case the 'Explain' column is too small to provide a sufficient analysis]

2 Clean Code Method Two

2.1 Before Code

[Include a screenshot/listing of your method(s) BEFORE cleaning up]

2.2 Analysis

[Insert the Clean Code Template here, fill in the 'is OK' column with 'yes/no/not applicable' and fill in the 'Explain' column for the BEFORE code]

[Provide further analysis/argumentation in case the 'Explain' column is too small to provide a sufficient analysis]

2.3 After Code

[Include a screenshot/listing of your method AFTER cleaning up]

2.4 Conclusion

[Insert the Clean Code Template here, fill in the 'is OK' column with 'yes/no/not applicable' and fill in the 'Explain' column for the AFTER code]

[Provide further analysis/argumentation in case the 'Explain' column is too small to provide a sufficient analysis]

3 ZetaStone

3.1 Design and UML

[Include a UML diagram that shows the design of the ZetaStone part of HotStone, with emphasis on the State pattern introduced and the existing winner strategies that are reused. (A good quality picture of a hand-drawn UML diagram is OK.)]

3.2 State Selection Code

[Include a screenshot of the (JaCoCo-painted[1]) state selection code in the Context implementation of the State pattern, showing that correct behaviour of ZetaStone has been tested.]

4 Backlog

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

- ...
- ...

[1] It would be fine if you present JaCoCo or IntelliJ Code Coverage painted code to show your tests cover both states.