COSC 603

Software Maintenance & Testing

Spring 2016

**Project #2 – Refactoring and Design Smells**

Name: Marlene Encinas

**Task 11 – Refactoring: Changing a Method’s Signature.** ***For this task, briefly describe your experience with this task and for which design smells changing a method’s signature could help make the code more maintainable and why. In addition, comment on why things are changing in other class than just Cell.java and how this affected the definitions of any other classes besides Cell.java.***

Answer:

I was able to change the method signature of abstract playAction() to Boolean return type and its corresponding parameter. I could fix the compiler errors by adding return true to the methods that needed to return a Boolean value and completed the refactoring. The Junit test ran satisfactorily after the refactoring.

Changing a method’s signature can help when:

* There poorly thought out abstractions and boundaries
* Gathering up little pieces into subparts
* Small changes in the code could force changes in different classes and their methods
* Too many parameters used
* Method is too long
* Data clumps

This change impacted other classes because these other classes implement playAction() method, thus they need to be updated as well.