1. Task – 6 : For this task, briefly describe if this mission is an oversight on the part of Eclipse’s refactoring operation and why or why not. Additionally, briefly describe how (or if) this operation is any different than a simple find all and replace.
   1. This mission is an oversight on the part of Eclipse’s refactoring operation. The reason is that refactoring is a central part of software maintenance to keep code from decaying to such a point that it is not maintainable. We could solve the problem with less work with Eclipse
   2. A simple find all and replace method is useful when we work on a very small chunk of code , we cannot apply it on a big project , there are too many things that need to be changed
2. Task 7 : For this task, briefly describe your experience with this task and for which design smells pushing down or pulling up a class’s field(s) and/or associated methods could help make the code more maintainable and why
   1. We won’t need to change many line of code when we just need to push an attribute down to the sub class in the hierarchy tree
   2. Less manual work equals to less silly typing error
3. Task – 8: For this task, briefly describe your experience with this task and for which design smells extracting an interface could help make the code more maintainable and why. In your description, be sure to include a description of which methods you extracted into the interface and what new files were created in this operation.
   1. The idea is that, in addition to owning squares (i.e., properties), players can own other things, something not in the original Monopoly game.
   2. It means that we have to control the owner, the price and the action on the property,
   3. All method should be in the interface except toString() for it belongs to Object
   4. Extracting an interface could help make the code more maintainable when the programmer defined many functions which have same interface and do same tasks in many separated classes.
4. Task – 9 : For this task, briefly describe your experience with this task including the method signature you extracted and why you chose this one
   1. Method signature:

**public** **int** calculateMonopoliesRent(**int** rentToCharge, String [] monopolies)

* 1. This method should be public because it doesn’t use any private data of the class, we could reuse it within other class. It works just like a supporting function

1. Task – 10 : For this task, briefly describe your experience with this task and for which design smells creating a local variable from repeated code could help make the code more maintainable and why. In addition, comment on whether it is always OK to do this to a function call and whether it could affect the correctness of a program.
   1. This local variable helps to make the code more maintainable. We reduce the number of function calls to get the same value, we won’t need to change the value more than one time when we get rid of repeated code.
   2. Creating a local variable from repeated code doesn’t affect the correctness of a program. It also helps to reduce the running time of the application
2. Task – 11 : 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.
   1. This change will help the user know whether the function call is successful or not by comparing the input and the output
   2. We also need to change the interface of setAvailable() in IOwnable