**Create a class hierarchy, that reflects to the class diagram below. Use ArgoUML to draw the diagram and create the Java code skeleton.**

We have two types of rooms: kitchen, living room

The living room has an owner, but the kitchen doesn’t.

Later, we want to add window(s) and door(s) to the rooms.

The diagram below shows the *decorator design pattern*. The idea of the decorator is that it stores a reference to the room, and the additional property, which is used to decorate the room. The decoration now is just a simple message printing to the console (it prints out the decorations: doors, windows). Because the decorator is also a kind of room, we can continue this decoration as long as we want. The only thing we have to take care of is that we have to pass the last decorated room to the new decorator. In order to make the draw() of the decorator work properly, its implementation should call first the draw() method of the decorated room, and then print out the decoration.

