**Lesson3**

**Name:** Yadir Hernandez Batista **StudentID:** 610545

**Problem #1:**

Person p1 = new PersonWithJob("Joe", 30000);

Person p2 = new Person("Joe");

//As PersonsWithJobs, p1 should be equal to p2

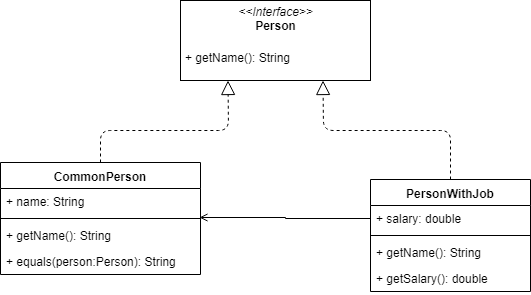
System.out.println("p1.equals(p2)? " + p1.equals(p2));

System.out.println("p2.equals(p1)? " + p2.equals(p1));

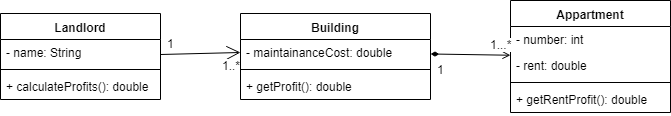
The implementation shows false at the first time because the PersonWithJob instance (p1) is checking that the argument the equal method receives is a PersonWithJob instance and p2 is a concrete Person object, not a PersonWithJob. By the contrary when p1 checks is compared with p2 as argument, the implementation returns true because PersonWithJob class is an inherited class of Person, so, by definition p1 is a Person and the comparison fills the requirements: the instance is not null, the instance is a Person and their names are equals.

**Solution**: Decorator pattern

By using a composition you can define PersonWithJob as a class that contains a person and extends instance’s features. Created a Person interface to maintain the concepts.

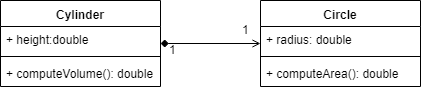


**Problem #2:**



**Problem 3:**

Don’t make sense because by definition a Cylinder is not a Circle, it can contain one as base and the height value.



**Problem 4:**

