1. System.out.println("p1.equals(p2)? " + p1.equals(p2)); 🡪 False, because p1 is PersonWithJob and p2 is Person, not PersonWithJob, it will go to equals method on PersonWithJob class, and return false because of this code :

if(!(aPerson instanceof PersonWithJob)) return false;

System.out.println("p2.equals(p1)? " + p2.equals(p1)); 🡪 True, because p2 is Person and p1 is PersonWithJob (it is instanceof Person) and the name of p1 is equal with the name of p2. It doesn’t check salary, only check name.

1. …
2. Does it make sense to use inheritance here? Explain.

🡪 Yes, inheritance makes sense in this scenario because a Cylinder is a specialized type of Circle, it promotes code reuse, improves maintainability. Inheritance allows the Cylinder class to reuse the properties and methods of the Circle class. This is beneficial because a Cylinder can be seen as a Circle with an additional height dimension. By inheriting from Circle, Cylinder can utilize the radius property and computeArea method directly, reducing code duplication. And if changes are made to the Circle class, such as modifying the way the area is computed, these changes will automatically be reflected in the Cylinder class.