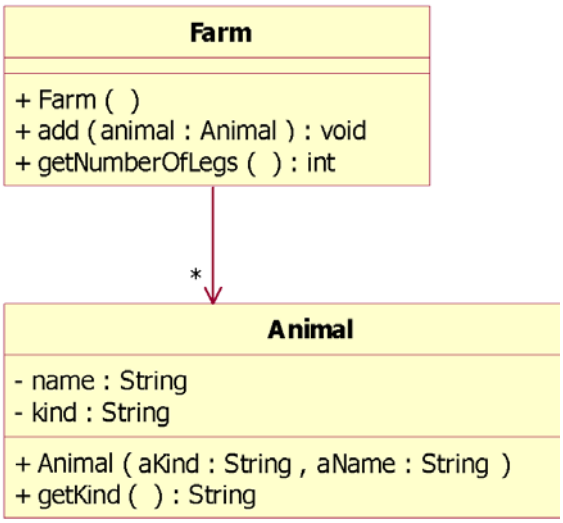
Q1 Consider the following system. Can you spot a bad smell in Farm.getNumLegs()? Can you fix it? Apply Larman's GRASP principles of **Information Expert** and **Polymorphism**. For your solution: draw a UML class diagram and write the code for Farm.getNumLegs (as well as any new code that you need to introduce).

```
public class FarmTest extends
…{

public void testGetNumLegs() {
 Farm f = new Farm();
 f.add(new Animal("Duck",
                  "Donald"));
 assertEquals(2,
f.getNumLegs());
 f.add(new Animal("Dog",
"Pluto"));
 assertEquals(6,
f.getNumLegs());
 }
}
```

Note: *while your refactoring might induce changes to the test case code, the assert lines must stay the same.*




```
public class Animal {

 private String name;
 private String kind;

 public Animal(String aKind,
                String
aName) {
  kind = aKind;
  name = aName;
 }

 public String getKind() {
  return kind;
 }
…
```

```
public class Farm {
  private List<Animal> animals =
new …;
  public int getNumLegs() {
   int result = 0;
   Iterator<Animal> it =
animals.iterator();
   while(it.hasNext()) {
        Animal a = it.next();

 if(a.getKind().equals("Duck")) {
             result += 2;
        } else
if(a.getKind().equals("Dog")) {
             result += 4;
        } else { /*?*/
        }
   }
   return result;
}
```