

DMS Comparator

Implement the Comparator `DMSComparator`, which compares `Animal` instances. An `Animal` instance is greater than another `Animal` instance if its **dynamic type** is more *specific*. See the examples to the right below.

In the second and third blanks in the `compare` method, **you may only use the integer variables predefined** (`first`, `second`, etc), **relational/equality operators** (`==`, `>`, etc), **boolean operators** (`&&` and `||`), **integers**, and **parentheses**.

As a *challenge*, use equality operators (`==` or `!=`) and no relational operators (`>`, `<=`, etc). There may be more than one solution.

```
class Animal {
    int speak(Dog a) { return 1; }
    int speak(Animal a) { return 2; }
}
class Dog extends Animal {
    int speak(Animal a) { return 3; }
}
class Poodle extends Dog {
    int speak(Dog a) { return 4; }
}
```

Examples:

```
Animal animal = new Animal();
Animal dog = new Dog();
Animal poodle = new Poodle();

compare(animal, dog) // negative number
compare(dog, dog) // zero
compare(poodle, dog) // positive number
```

```
1 public class DMSComparator implements _____ {
2
3     @Override
4     public int compare(Animal o1, Animal o2) {
5         int first = o1.speak(new Animal());
6         int second = o2.speak(new Animal());
7         int third = o1.speak(new Dog());
8         int fourth = o2.speak(new Dog());
9
10        if (_____ ) {
11            return 0;
12
13        } else if (_____ ) {
14            return 1;
15        } else {
16            return -1;
17        }
18    }
19 }
```