Inheritance ---> Polymorphism ---> Case study for Inheritance + Polymorphism (Design pattern)

- Polymorphic (having many different shapes/forms)

Example#1:

```
UIControl (render) --- Checkbox (render)
|
Radiobutton (render)
```

Lessons:

- Interfaces can serve as data types in Java. If I use interface name as a data type for a given variable, then the only objects that can be referenced by that variable are the ones created from classes that implement Speaker interface.

```
Speaker ---> speak()
Philosopher implements Speaker
Politician implements Speaker

Speaker s = new Philosopher();
```

- Idea: generate random objects (shapes).

Example#2: RandomShapeGenerator

The version of the method that should be called is determined by the type of the object referenced by the variable used to call the method.

- Bigger example:

Example#3: YetAnotherPolymorphismExample

```
Speaker ---> speak()
Philosopher implements Speaker
Politician implements Speaker + bicker method

Speaker s = new Politician();
((Politician) s).bicker(); ?
```