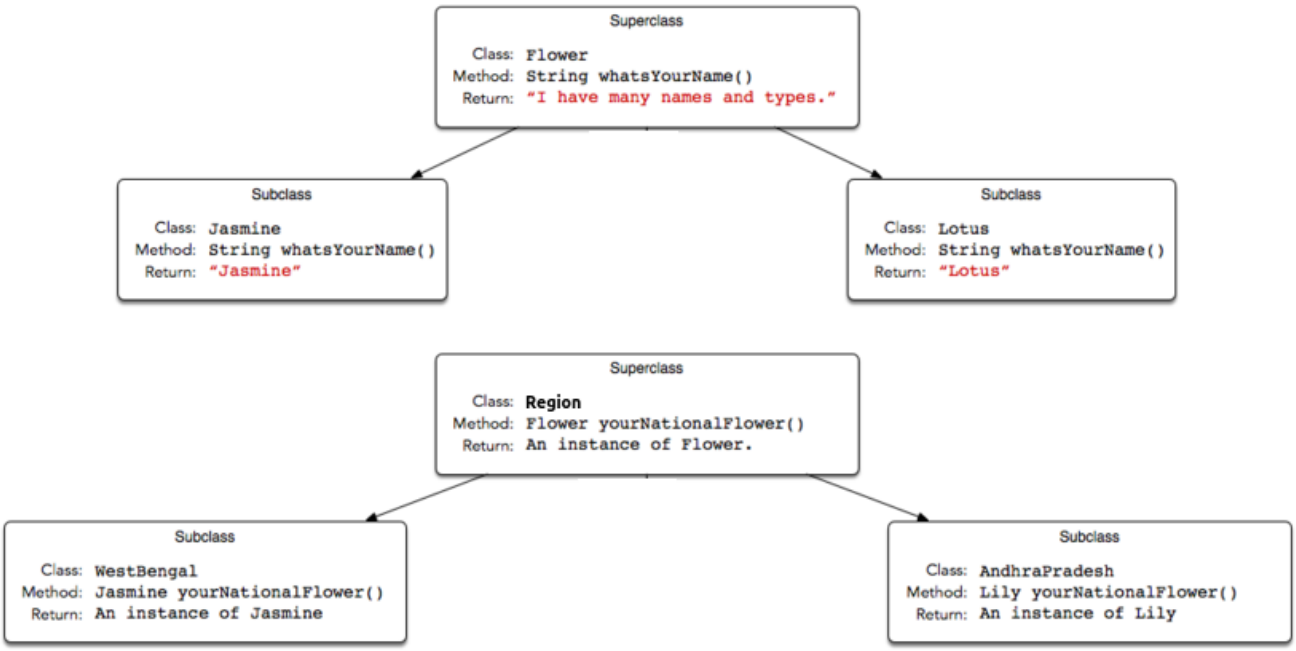
Covariant Return Types

Java allows for Covariant Return Types, which means you can vary your return type as long you are returning a subclass of your specified return type.

Method Overriding allows a subclass to *override* the behavior of an existing superclass method and specify a return type that is some subclass of the original return type. It is best practice to use the `@Override` annotation when overriding a superclass method.

Implement the classes and methods detailed in the diagram below:



You will be given a partially completed code in the editor where the *main* method takes the name of a state (i.e., `WestBengal`, or `AndhraPradesh`) and prints the national flower of that state using the classes and methods written by you.

Note: *Do not* use access modifiers in your class declarations.

Resources
Covariant Return Type
Java Covariant Type

Input Format

The locked code reads a single string denoting the name of a subclass of *State* (i.e., `WestBengal`, `Karnataka`, or `AndhraPradesh`), then tests the methods associated with that subclass. You are not responsible for reading any input from stdin.

Output Format

Output is handled for you by the locked code, which creates the object corresponding to the input string's class name and then prints the name returned by that class' national flower's *whatsYourName* method. You are not responsible for printing anything to stdout.

Sample Input 0

```
AndhraPradesh
```

Sample Output 0

```
Lily
```

Explanation 0

An *AndhraPradesh* object's *yourNationalFlower* method returns an instance of the *Lily* class, and the *Lily* class' *whatsYourName* method returns `Lily`, which is printed by the hidden code checker.