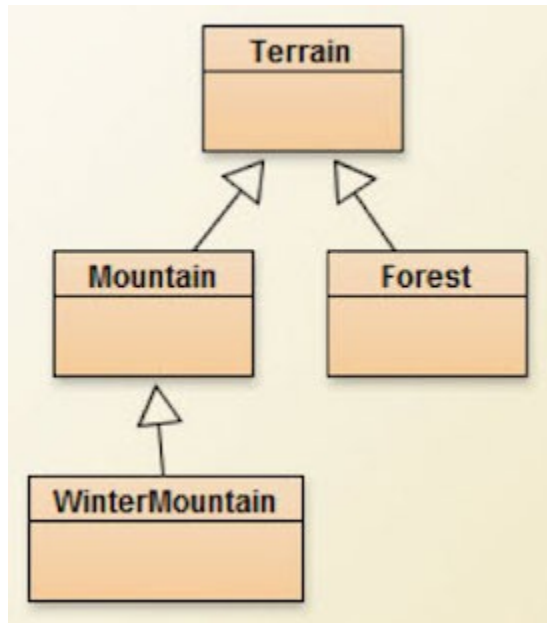


09.02 Assignment Instructions

Instructions: Create a class hierarchy of simple terrains that might be used in a graphical video game.

Background: This project will give you a feel for how programmers create terrains in which to place characters within a graphical video game. The terrains will be very simplistic representations containing details specific to each. When creating a game, many terrains are often used. Instead of starting from scratch each time, a programmer can simply start with a previous one that closely matches it and build it up from there. This saves lots of time and allows developers to reuse their resources wisely. The key to completing this assignment is to carefully manage the relationships between the super classes and the subclasses.

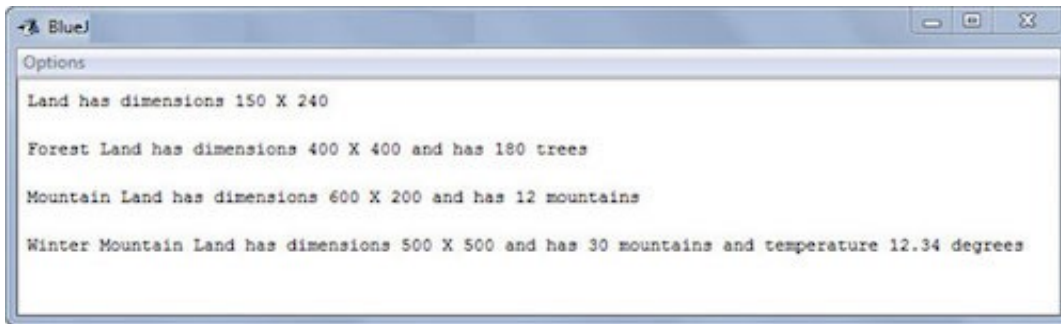
1. Create a new project called **09.02 Assignment** in your Module 09 Assignments folder.
2. Download the **Terrain.java** file to the new folder.
3. For this project, create each of the remaining classes for the hierarchy shown in the diagram.



4. A `Mountain` class has an attribute for the number of mountains.
5. A class `WinterMountain` has all the attributes of a mountain plus the temperature.
6. A class `Forest` has an attribute for the number of trees.
7. Create a program to test your terrains. Name it `GameTerrainTester` in your assignment folder.
Be sure to instantiate at least one instance of every terrain. When the program is run, each terrain and the values of its attributes need to be neatly displayed.

8. Add two more terrains to your project. Only one class can be a direct subclass of `Terrain`. Each new class needs an attribute that is unique to its terrain.

Expected Output: Your output should look similar to the image below but may vary based on the values used.



```
Options
Land has dimensions 150 X 240
Forest Land has dimensions 400 X 400 and has 180 trees
Mountain Land has dimensions 600 X 200 and has 12 mountains
Winter Mountain Land has dimensions 500 X 500 and has 30 mountains and temperature 12.34 degrees
```



 Print