In this lab, you will use inheritance. You will use an abstract parent class that has instance data from a Vocabulary class. The Vocabulary class uses the PartOfSpeech class you wrote in an earlier lab.

As you examine the Vocabulary class, you can see that the relationship between Vocabulary and PartOfSpeech is a HAS-A relationship. There is no inheritance here. The class and a tester are provided for you to see how the two classes work together.

Download these files and put them in a project/Lab6 folder PartOfSpeech.java, Vocabulary.java, VocabularyTest1.java. All the word files are in word files.zip. Unzip them and place them in your project, or download them one by one.

Once you study and grasp the Vocabulary and PartOfSpeech classes and relationship, you can move on to study and review the Robot parent class, and its relationship to the Robot children. The Robot parent HAS-A Vocabulary object as instance data. The Robot children inherit from the Robot parent, so each Robot child uses the parent Vocabulary in a different way. The Robot children are Robots, or we can say RedRobot is a Robot; YellowRobot is a Robot, etc.

**Download these files:**

Robot.java

RedRobot.java, YellowRobot.java

PartOfSpeech.java

Vocabulary.java

word files.zip

Construct a set of classes according to this diagram. Be sure to use class, variable, and method names as noted.

***A PoetryGenerator1 class with a main method that generates phrases according to the rules below is provided for you.***  RedRobot and YellowRobot classes already follow these rules. Model your additional Robot children (GreenRobot, BlueRobot, PurpleRobot) on the RedRobot and YellowRobot provided, using the rules below. Add a small main method at the bottom of each child for a simple test to make sure the robot children generate the phrases according to the rules.

Rules for phrase generation:

* + The red robots understand adjectives and nouns, and produce one of the following types of phrases
    - 1 adjective followed by 1 noun (i.e., “happy campers”)
    - 2 adjectives followed by 1 noun (i.e., “clean calm bird”)
  + The yellow robots understand nouns, verbs, adjectives, and adverbs and produce one of the following types of phrases
    - 2 words of the same type (any type is possible for the pair) joined by “or” (i.e., “runs or hides”)
    - 2 words of the same type (any type is possible for the pair) joined by “and” (i.e., “cow and chicken”)
  + The blue robots understand verbs and adverbs and produce one of the following types of phrases
    - 1 verb followed by 1 adverb (i.e., “eating rapidly”)
    - 1 verb followed by 2 adverbs joined by “and” (i.e., “reading astutely and barbarically”)
  + The purple robots understand prepositions, articles, adjectives, and nouns, and produce one of the following types of phrases
    - combine 1 preposition, 1 article, and 1 noun (“under the rock”)
    - combine 1 preposition, 1 article, 1 adjective, and 1 noun (“under the rock”)
  + The green robots understand articles, adjectives, nouns, and verbs and produce one of the following types of phrases
    - combine 1 article, 1 noun, and 1 verb (i.e., “the sun burns”)
    - combine 1 article, 1 adjective, 1 noun, and 1 verb (i.e., “the happy sun burns”)

The Robot class is the parent/super class for all the color robots.

Here is a diagram of the relationship of parent class Robot with the child robots. The abstract class Robot and child classes RedRobot, GreenRobot, PurpleRobot, BlueRobot, and YellowRobot are in the following hierarchy

Note that # indicates protected and italics indicate abstract



Note the PoetryGenerator1.java file will not compile until you complete the missing Robots. Feel free to modify it but don’t delete any of the robots.

For this lab, **submit** your three additional **robots** with output from the small main at the end of each program file pasted into a comment. Also **submit** your **PoetryGenerator1**.java file with its output pasted into a comment at the end of the file.