CIS351-Inheritence Lab

Submission Instructions

Submit a zip folder consisting of Flower.java and PlantArrayListExample.java in Blackboard

Download Materials

Download the Plant.java and PlantArrayListExample.java file before starting the lab

Background

In today's lab you will simulate a virtual garden through Inheritance and ArrayList. The Arraylist will act like the virtual garden that can hold two different types of objects. One of them is plant object and the other is flower object. Here, Plant.java is the `base/parent` class and Flower.java is the `derived/child` class.

Both Plant and Flower objects have some common properties like: name and cost. But Flower have some additional properties like: if_flower_is_annual and color

Your program should ask the user to enter plant's information as follows: `plant plantName plantCost`

Your program should also ask the user to enter flower's information as follows: `flower flowertName flowerCost isAnnual color`

The program should read plants or flowers from input (ending with -1), adding each plant or flower to the myGarden ArrayList and then printing them.

Here is an example Test Scenario:

If the input is:

plant Spirea 10
flower Hydrangea 30 false lilac
flower Rose 6 false white
plant Mint 4
-1

The output is:

Plant Information:
Plant name: Spirea
Cost: 10

Plant Information:
Plant name: Hydrengea
Cost: 30
Annual: false
Color of flowers: lilac

Plant Information:
Plant name: Rose
Cost: 6
Annual: false
Color of flowers: white

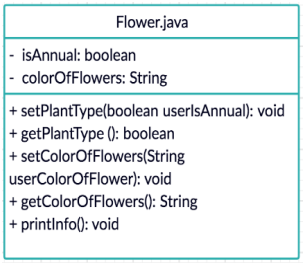
Plant Information:
Plant name: Mint
Cost: 4

Instructions

For this part of the lab, first you need to create a java file named Flower.java

Specificaton for *Flower.java*

1. Now, you need to implement a derived Flower class, given the Plant class is the base. Meaning, the Flower class will be the child class and Plant class will be the parent class. Make sure, both these classes are in the same folder.
2. Read the provided *Plant.java* class carefully to understand what is does. Without having good understandiong of this class, it may be difficult to complete the lab.
3. Then write the class correspondinf to the UML diagram fiven for *Flower.java* class.



Specificaton for *PlantArrayListExample.java*

1. Now open the PlantArrayListExample.java file, which you should have downloaded from Blackboard. This class will have the main method, which you will need to complete.
2. Complete the *main()* method. Start out by creating an ArrayList called myGarden that can store Plant objects. Note: Since Plant is the base class and Flower is the derived class, the Arraylist declared to store Plant will be able to refer to either Plant object or Flower object. See the slide deck for Inheritance and polymorphism if you need to revise these concepts.The ArrayList should be able to store objects that belong to the Plant class or the Flower class.
3. The program should read plants or flowers from input (ending with -1), adding each Plant or Flower to the myGarden ArrayList, and output each element in myGarden.
4. Finally, read the additional instructions provided in PlantArrayListExample.java. Carefullly follow those instructions to complete the main method.
5. Once you complete the main method you will be able to test the Flower class and Plant class based on the example test scenerio provided at the beginning of the handout.

Grading Criteria

Total points: 100 points

Flower.java - 50pt and PlantArrayListExample.java - 50pt

Farzana Rahman / *frahman@syr.edu*