CSE115 Lab 4 Spring 2015

Introduction

In this lab you will use the Java you have learned in lecture to write a small program. You will apply what you know about the composition relationship, methods, and the manipulation of variables. In particular you will practice creating a named package, creating a class, setting up a composition relationship, instantiating classes, using assignment statements to associate values with variables, calling methods, and swapping the values of variables.

Notice for those students who have some prior programming experience

> *There are many ways you could solve these lab exercises. They are designed to teach you particular problem-solving techniques. For this reason you may use only those language features which we have studied in class in your solution. Use of other approaches (including but not limited to primitive data types and conditional statements of any kind) may satisfy Web-CAT's automated grading, but will not be accepted by the graders on their review of submissions, and will result in a reduced grade, perhaps zero. You must demonstrate mastery of the skills/concepts taught to receive full credit.*

Preparatory tasks

1. Log in
2. Start Eclipse
3. Switch to the CVS Repository Exploring perspective
4. Check out the SP15-CSE115-Lab4 project from the Labs repository
5. Switch to the DrJava perspective

Lab Tasks

1. Define a class named lab4.EastFarm so that it is in a composition relationship with the example1.Terrarium class. Creating the class involves first creating a package named 'lab4', then creating a class named 'EastFarm' inside that package. Setting up the composition relationship involves three things:

   a. the *declaration* of an instance variable of the example1.Terrarium type (remember that instance variables use the '*private*' access control modifier),

   b. the *instantiation* of example1.Terrarium in the constructor of the lab4.EastFarm class, and

   c. and finally the *assignment* of the resulting reference to the instance variable.

If you are unsure how to do any of these things, REVIEW YOUR LECTURE NOTES and the posted LECTURE SLIDES. These are all things we have discussed and shown you how to do in lecture. Now it is your turn: apply what you have learned!

Once the lab4.EastFarm class has been created, define three methods as follows:

a. Define a void method named ‘addPig’ which adds a new example1.Pig to the composed Terrarium, and starts the Pig moving (by calling the start() method on the Pig object).

Again, if you are unsure how to do any of these things, REVIEW YOUR LECTURE NOTES and the posted LECTURE SLIDES.

b. Also define a void method named ‘addButterfly’ which adds a new example1.Butterfly to the composed Terrarium, and starts the Butterfly moving.

c. Define a void method named ‘addChicken’ which adds a new example1.Chicken to the composed Terrarium, and starts the Chicken moving.

2. Define a class named lab4.WestFarm so that it is in two composition relationships with the example1.Terrarium class. (*Hint: this means that there will be two instance variables of type example1.Terrarium in the lab4.WestFarm class.*)

   Define the methods ‘addPig’, ‘addButterfly’ and ‘addChicken’ as described above, but with the following modification. Define these methods so that each time any of these method is called, the Terrarium that the critter is added to is different: the first and second critters added must be added to the first and second Terrarium, respectively, after which adding cycles back to the first Terrarium. Therefore the third, fourth, fifth and sixth critters added must be added to the first, second, first and second Terrarium, respectively, and so on.

   The idea is that you’re a farmer and you need to ensure that no one Terrarium gets more crowded than any other, so as animals come in you will alternate which one you put the critters into.

   For example, if we label the Terrariums _t1 and _t2 the following code must result in three Butterflies, one Chicken and one Pig in _t1, and one Pig, two Butterflies and two chickens in _t2:

```
lab4.WestFarm et = new lab4.WestFarm();
et.addButterfly();          <- added to _t1
et.addPig();                <- added to _t2
et.addChicken();            <- added to _t1
et.addChicken();            <- added to _t2
et.addButterfly();          <- added to _t1
et.addButterfly();          <- added to _t2
et.addButterfly();      <- added to _t1
et.addButterfly();      <- added to _t2
et.addPig();            <- added to _t1
et.addChicken();        <- added to _t2
```

Your TA will discuss with you the technique to use to achieve this behavior. It involves interchanging (swapping) the values of variables.

Submitting your project to Web-CAT

Make sure you submit your work on time; due dates are listed on the Labs page of the course website. This lab will be automatically graded by Web-CAT but graders will review all submissions, make grade adjustments and give feedback as appropriate.