Java Programming: Lab 8

OOP Interfaces

Essentials

Problem 1:

*Check.*

Problem 2:

This code is for a class called Snake.

Paste your Java code here:

Problem 3:

A common design problem is writing a class that needs to know when another class has changed. For example, you have a GUI that displays stock prices from a stock exchange. It would be helpful if there could be a way to tell the GUI when the data changes, and the GUI can update.

Java provides an interface called Observer that can be implemented by a class that should be notified when an object changes.

HYPERLINK "http://docs.oracle.com/javase/7/docs/api/java/util/Observer.html"http://docs.oracle.com/javase/7/docs/api/java/util/Observer.html

The object that is to be observed should subclass the Java Observable class.

HYPERLINK "http://docs.oracle.com/javase/7/docs/api/java/util/Observable.html"http://docs.oracle.com/javase/7/docs/api/java/util/Observable.html

An Observable object can have one or more Observers. You’ll have to set which objects are observing your Observable. This Observable class, when it needs to tell its observers it has updated, can set a flag that means it has updated, and then call one method – notifyObservers - to notify all of its observers.

Here’s some code that demonstrates Observer and Observable.

Can you add comments to this code to explain what is going on? How does information flow through the program?

Can you add another class called EggCounter? EggCounter should also be an Observer. It will also observe your Chicken objects, and keep track of how many eggs each chicken has laid.

Your EggCounter should do ALL of the counting – don’t add a counter to your chicken class! Your Chickens don’t know and don’t care how many eggs they have laid.

Add some more Chickens to TestObserver and lay some more eggs to test your code.

import java.util.Observable;

import java.util.Observer;

public class TestObserver {

public static void main(String[] args) {

EggAlert eggMonitor = new EggAlert();

Chicken a = new Chicken("Mavis");

a.addObserver(eggMonitor);

Chicken b = new Chicken("Betty");

b.addObserver(eggMonitor);

a.layEgg();

b.layEgg();

a.layEgg();

}

}

class Chicken extends Observable {

String name;

boolean laidEgg = false;

Chicken(String name){

this.name = name;

}

public void layEgg(){

this.laidEgg = true;

this.setChanged();

this.notifyObservers();

this.laidEgg = false;

}

public String toString() {

return this.name;

}

public boolean laidEgg(){

return laidEgg;

}

}

class EggAlert implements Observer {

public void update(Observable chicken, Object arg) {

if (((Chicken)chicken).laidEgg() ){

System.out.println(chicken + " has laid an egg");

}

}

}

Problem 4:

Fork and clone the Snake game class project application from GitHub. Find where an interface is used. This code will be the basis for project 2.

HYPERLINK "https://github.com/minneapolis-edu/Snake" https://github.com/minneapolis-edu/Snake

Question: Which class(s) implements an interface? Why?

Advanced

Finish any incomplete Essentials problems first.

Spring break! No advanced section this week. Do remember to finish the Essentials section.

If you'd like something coding-related to do this week, I recommend you review some of your OO code and finish any incomplete problems from previous labs. Or, write some new code, or practice with GitHub, or practice some CodeBat problems, or start class project 2… or simply enjoy the break :)

Nothing to upload for this section.