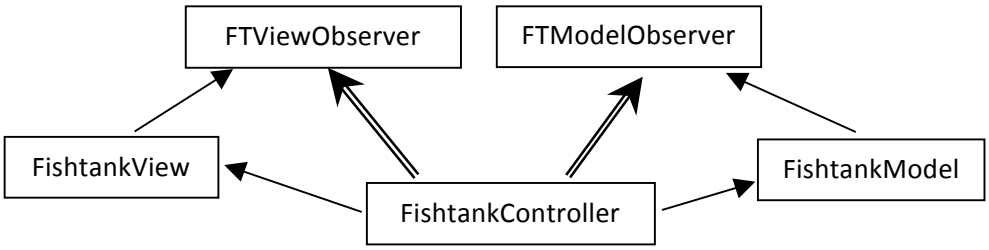
CSCI 3353 Object Oriented Design
Homework Assignment 9, part 2
Due Friday, December 2

In the first part of this assignment, you re-designed the fish tank code from HW 6 to follow the MVC pattern. Your task in this part of the assignment is to implement this design.

To ensure some consistency, I would like everyone to download my solution to part 1, and use that design to guide your implementation. I also would also like your implementation to include the class and interface names from the following class diagram, which was given in part 1:



Your implementation will almost certainly need additional classes. It is your responsibility to figure out what you need. If a class has similar functionality to a class from HW 6, please use the same class name. Otherwise, you are free to use whatever class names you like. You should discover that you don't need to write a lot of new code – most of your code should come from my solution to HW 6.

Once you get the fish tank working, I want you do implement an additional view. This view is a swing frame that displays how many fish are in the tank. For example, the screenshot below shows what my view looks like:

Just to make things interesting, you should implement the view so that the count (in the above screenshot, the text field containing the number 7) flashes on and off. The way to implement the flashing is to use a timer. Each time the timer goes off, the component's *setVisible* method gets called – first with the argument *true*, then with the argument *false*, etc.

My solution to part 1 says that the *Fish* class should be divided into two parts: one part that contains the images, and one part that contains the position, direction, and movement strategy. This is a good idea, but turns out to be impractical in this case. So I encourage you to **not** change the *Fish* class from how it is in HW 6. Instead, both the model and the view will hold the current collection of fish.

My solution to part 1 also says that the model should contain the size of the fish tank. A user can change size of the tank by clicking on an edge of the window and dragging. The view therefore needs to be able to tell the model when the tank changes its size. The best way to do this is to create a component listener for the view. Look up the interface *ComponentListener* and class *ComponentAdapter* in the JavaDoc documentation. Your view should have code that looks something like this:

```
addComponentListener(new ComponentAdapter() {
   public void componentResized(ComponentEvent e){
      // code to execute when the view is resized
   }
});
```

WHAT TO SUBMIT: Please strip off all package declarations from your Java files, and compress them into a single zip file. You do not need to include the fish image files. Then submit the zip file to Canvas.