

## Lab 7

Write an application that handles camera information we receive from different camera's above or next to interstates and highways. We receive camera records with the attributes license plate, speed and camera id. Every camera has its own id. The cameras have built-in software to recognize license plates and to measure the speed of the cars. Our application has to handle these camera records in the following way:

We need to detect every car that is stolen. If we receive a camera record of a stolen car, we need to notify the police (using a `system.out.println`)

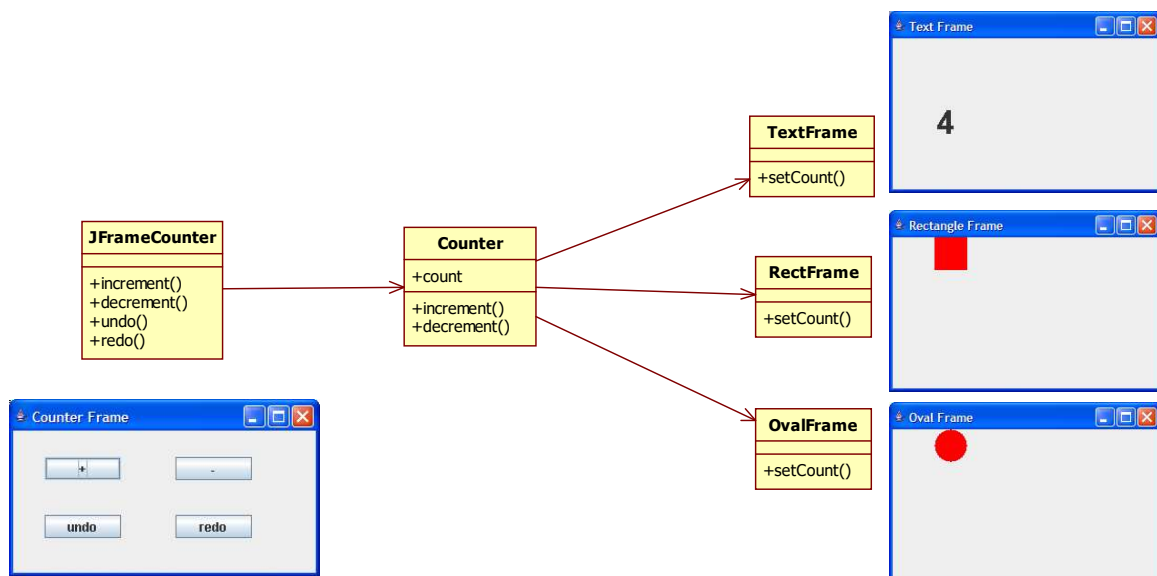
We need to detect every car that is speeding. If we receive a camera record of a car that drives too fast, we need to send the owner a speeding ticket (using a `system.out.println`)

We need to detect every car that is currently not registered. If we receive a camera record of a car that is currently not registered, we need to send the owner a ticket (using a `system.out.println`)

We also need to detect every car whose owner has one or more unpaid tickets. If we receive a camera record of a car whose owner has one or more unpaid tickets, we need to notify the police (using a `system.out.println`)

It should be easy to add other detection logic based on the camera records we receive in the system

- Draw the UML class diagram of your system
  - Draw the sequence diagram that show how your design works
  - Implement your design in Java. In the `main()` method of the Application you can simulate that we receive different camera records
- d. In lab 6 we applied different pattern to the counter application:



Now we get the following additional requirements:

- Whenever the counter value changes we want to do the following:
  - If the counter value is even and the counter value  $< 10$  or if the counter value is even and the counter value  $= 12$  or  $13$  then we print in the console : **“Red”**
  - If the counter value is even and the counter value  $\geq 10$  but not  $12$  or  $13$  then we print in the console : **“Green”**
  - If the counter value is odd and the counter value  $< 15$  or if the counter value is odd and the counter value  $= 17$  or  $19$  then we print in the console : **“Blue”**
  - If the counter value is odd and the counter value  $\geq 15$  but not  $17$  or  $19$  then we print in the console : **“Orange”**
- The counter value needs to be stored in the database. You don't need to do actual database access, but you can simulate that with a `System.out.println()`
- Anytime the counter value changes, we want to write that to a log file (you can simulate this with a `System.out.println()`)
- We want a clear separation between UI, domain and database access

Draw the class diagram based on the solution of the previous lab. Your class diagram also needs to show the patterns we used in the previous labs.

- e. Draw the sequence diagram that shows the following scenario:
  - a. The user clicks the increment button
  - b. The user clicks undo
- f. Implement your new design in Java. Your starting code should be the solution of lab 6. So the code should also implement the patterns of the previous exercise.