|  |  |
| --- | --- |
| Course | Advanced Software Design – CS525 |
| Assignment | Lab 7 |
| Week | 07 |
| Due | Mar 30, 2020 |
| Student name | Quan Hong Doan |
| Student ID | 986956 |

Online version:

<https://github.com/zithiat/asd/blob/master/labs%20%26%20assignments/Answers/Assignment_CS525_Week07_986956.docx>

**Problem 1**:

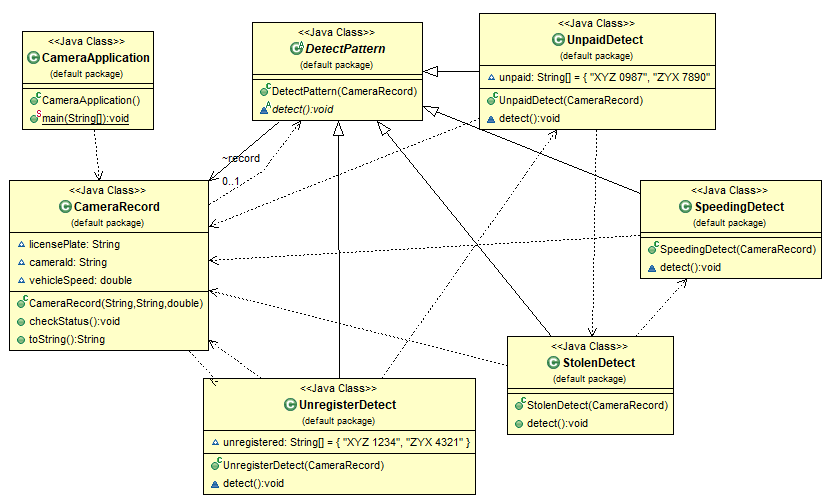
Problem 1 is the question a, b, c from the PDF file.

**Answer**:

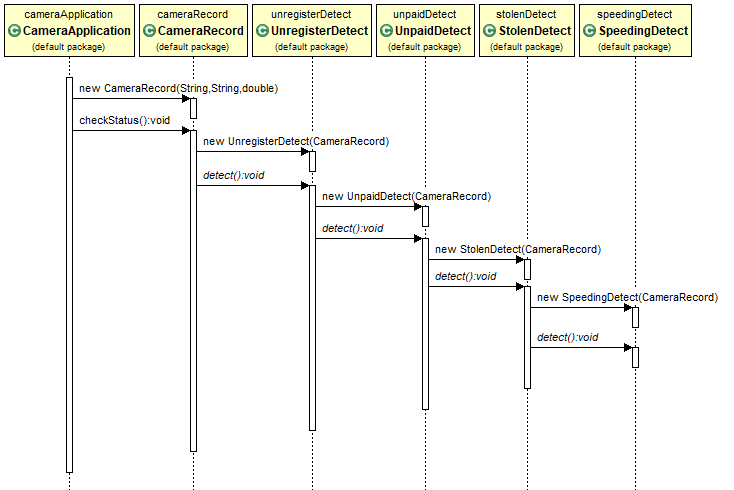
The source code on my GitHub

<https://github.com/zithiat/asd/tree/master/codes/code%20for%20labs/lab7/CameraApplication>

**Class diagram**



**Sequence diagram**



**CameraApplication**

**public** **class** CameraApplication {

**public** **static** **void** main(String[] args) {

String[] vecList = { "ABC 1234", // stolen

"XYZ 1234", // unregistered

"XYZ 0987", // unpaid

"ABC 6543", // speeding

"XZY 5678" // normal

};

String[] camIds = { "cam123", "cam234" };

CameraRecord rec1 = **new** CameraRecord(camIds[0], vecList[0], 54);

CameraRecord rec2 = **new** CameraRecord(camIds[1], vecList[1], 58);

CameraRecord rec3 = **new** CameraRecord(camIds[0], vecList[2], 45);

CameraRecord rec4 = **new** CameraRecord(camIds[1], vecList[3], 87);

CameraRecord rec5 = **new** CameraRecord(camIds[0], vecList[4], 64);

rec1.checkStatus();

rec2.checkStatus();

rec3.checkStatus();

rec4.checkStatus();

rec5.checkStatus();

}

}

**CameraRecord**

**public** **class** CameraRecord {

String licensePlate;

String cameraId;

**double** vehicleSpeed;

**public** CameraRecord(String cameraId, String licensePlate, **double** vehicleSpeed) {

**this**.licensePlate = licensePlate;

**this**.cameraId = cameraId;

**this**.vehicleSpeed = vehicleSpeed;

}

**public** **void** checkStatus() {

System.***out***.println(**this**.toString());

DetectPattern pattern = **new** UnregisterDetect(**this**);

pattern.detect();

System.***out***.println();

}

@Override

**public** String toString() {

**return** "Camera=" + **this**.cameraId + "\tplate=" + **this**.licensePlate + "\tspeed=" + **this**.vehicleSpeed;

}

}

**DetectPattern**

**public** **abstract** **class** DetectPattern {

CameraRecord record;

**public** DetectPattern(CameraRecord record) {

**this**.record = record;

}

**abstract** **void** detect();

}

**UnregisteredDetect**

**public** **class** UnregisterDetect **extends** DetectPattern {

String[] unregistered = { "XYZ 1234", "ZYX 4321" };

**public** UnregisterDetect(CameraRecord record) {

**super**(record);

}

@Override

**void** detect() {

**if** (Arrays.*asList*(unregistered).contains(**this**.record.licensePlate)) {

System.***out***.println("Unregistered detected!");

} **else** {

DetectPattern nextHandle = **new** UnpaidDetect(**this**.record);

nextHandle.detect();

}

}

}

**UnpaidDetect**

**public** **class** UnpaidDetect **extends** DetectPattern {

String[] unpaid = { "XYZ 0987", "ZYX 7890" };

**public** UnpaidDetect(CameraRecord record) {

**super**(record);

}

@Override

**void** detect() {

**if** (Arrays.*asList*(unpaid).contains(**this**.record.licensePlate)) {

System.***out***.println("Unpaid ticket whose owner detected!");

} **else** {

DetectPattern nextHandle = **new** StolenDetect(**this**.record);

nextHandle.detect();

}

}

}

**StolenDetect**

**public** **class** StolenDetect **extends** DetectPattern {

**public** StolenDetect(CameraRecord record) {

**super**(record);

}

@Override

**public** **void** detect() {

**if** (record.licensePlate.contains("ABC 1234")) {

System.***out***.println("Stolen detected!");

} **else** {

DetectPattern nextHandle = **new** SpeedingDetect(record);

nextHandle.detect();

}

}

}

**SpeedingDetect**

**public** **class** SpeedingDetect **extends** DetectPattern {

**public** SpeedingDetect(CameraRecord record) {

**super**(record);

}

@Override

**void** detect() {

**if** (**this**.record.vehicleSpeed >= 65) {

System.***out***.println("Speeding detected!");

} **else** {

System.***out***.println("Normal");

}

}

}

Console output:

Camera=cam123 plate=ABC 1234 speed=54.0

Stolen detected!

Camera=cam234 plate=XYZ 1234 speed=58.0

Unregistered detected!

Camera=cam123 plate=XYZ 0987 speed=45.0

Unpaid ticket whose owner detected!

Camera=cam234 plate=ABC 6543 speed=87.0

Speeding detected!

Camera=cam123 plate=XZY 5678 speed=64.0

Normal

**Problem 2**:

Problem 2 is the question about the **Counter** modification from the PDF file.

**Answer**:

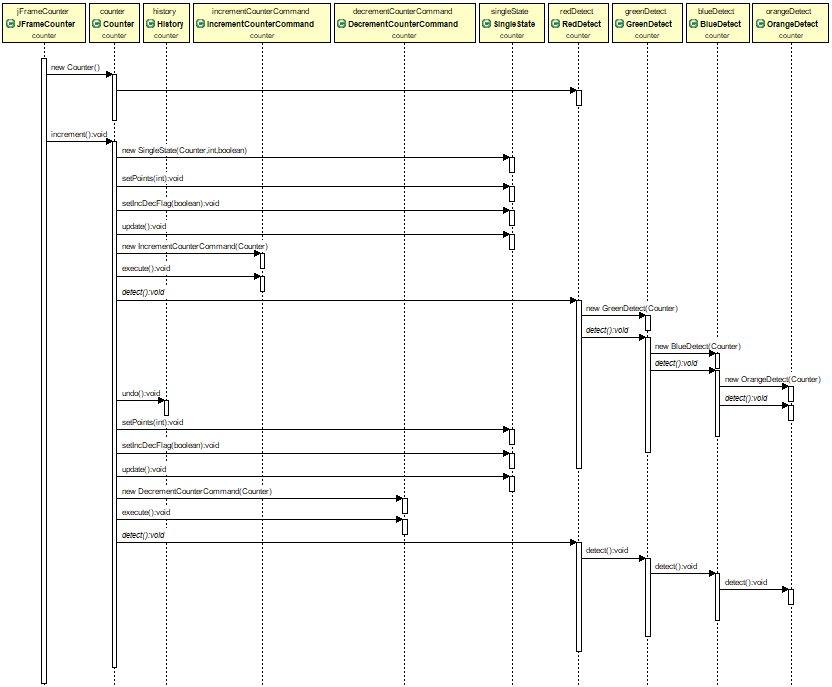
The source code on my GitHub

Class diagram

A close up of a map

Description automatically generated

Sequence diagram



Modification and new codes:

**Counter** class modification

**public** **void** increment() {

count++;

**this**.state.setIncDecFlag(**true**);

**this**.state.setPoints(count);

**this**.state.update();

**this**.observerList.stream().forEach(e -> e.update(count));

**this**.detect.detect();

Logger.*writeLog*("Incrementing " + count);

}

**public** **void** decrement() {

count--;

**this**.state.setIncDecFlag(**false**);

**this**.state.setPoints(count);

**this**.state.update();

**this**.observerList.stream().forEach(e -> e.update(count));

detect.detect();

Logger.*writeLog*("Decrementing " + count);

}

**DetectPattern**

**public** **abstract** **class** DetectPattern {

**public** Counter counter;

**public** DetectPattern(Counter counter) {

**this**.counter = counter;

}

**abstract** **void** detect();

}

**RedDetect**

**public** **class** RedDetect **extends** DetectPattern {

// 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"

**public** RedDetect(Counter counter) {

**super**(counter);

}

@Override

**void** detect() {

**int** c = counter.getCount();

**if** ((c < 10 & c % 2 == 0) | c == 12 | c == 13) {

System.***out***.println("Red: even, less than 10, or equal 12 or 13");

} **else** {

DetectPattern green = **new** GreenDetect(counter);

green.detect();

}

}

}

**GreenDetect**

**public** **class** GreenDetect **extends** DetectPattern {

// If the counter value is even and the counter value >= 10 but not 12 or 13 then we

// print in the console : "Green"

**public** GreenDetect(Counter counter) {

**super**(counter);

}

@Override

**void** detect() {

**int** c = counter.getCount();

**if** (c >= 10 & c % 2 == 0 & (c != 12 | c != 13)) {

System.***out***.println("Green: even, greater than 9 and not equal 12, nor 13");

} **else** {

DetectPattern blue = **new** BlueDetect(counter);

blue.detect();

}

}

}

**BlueDetect**

**public** **class** BlueDetect **extends** DetectPattern {

// 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”

**public** BlueDetect(Counter counter) {

**super**(counter);

}

@Override

**void** detect() {

**int** c = counter.getCount();

**if** ((c % 2 != 0 & c < 15 & (c != 12 | c != 13)) | c == 17 | c == 19) {

System.***out***.println("Blue: odd, less than 15, or equal 17 or 19");

} **else** {

DetectPattern orange = **new** OrangeDetect(counter);

orange.detect();

}

}

}

**OrangeDetect**

**public** **class** OrangeDetect **extends** DetectPattern {

// If the counter value is odd and the counter value >= 15 but not 17 or 19 then we

// print in the console : "Orange"

**public** OrangeDetect(Counter counter) {

**super**(counter);

}

@Override

**void** detect() {

**int** c = counter.getCount();

**if** (c >= 15 & c % 2 != 0 & (c != 17 | c != 19 | c != 12 | c != 13)) {

System.***out***.println("Orange: Odd, more than 14, not equal 17, nor 19");

}

}

}

**Logger** (could be implementing as Observer PULL or PUSH mechanism)

**public** **class** Logger {

**public** **static** **void** writeLog(String msg) {

System.***out***.println("Logger: " + msg);

}

}

**Console output**

SingleState updated points: 2

Red: even, less than 10, or equal 12 or 13

Logger: Incrementing 2

SingleState updated points: 4

Red: even, less than 10, or equal 12 or 13

Logger: Incrementing 4

SingleState updated points: 6

Red: even, less than 10, or equal 12 or 13

Logger: Incrementing 6

SingleState updated points: 8

Red: even, less than 10, or equal 12 or 13

Logger: Incrementing 8

SingleState updated points: 10

Green: even, greater than 9 and not equal 12, nor 13

Logger: Incrementing 10

DoubleState Received points: 13

Red: even, less than 10, or equal 12 or 13

Logger: Incrementing 13

DoubleState Received points: 16

Green: even, greater than 9 and not equal 12, nor 13

Logger: Incrementing 16

DoubleState Received points: 19

Blue: odd, less than 15, or equal 17 or 19

Logger: Incrementing 19

DoubleState Received points: 22

Green: even, greater than 9 and not equal 12, nor 13

Logger: Incrementing 22

DoubleState Received points: 25

Orange: Odd, more than 14, not equal 17, nor 19

Logger: Incrementing 25

DoubleState Received points: 28

Green: even, greater than 9 and not equal 12, nor 13

Logger: Incrementing 28

DoubleState Received points: 31

Orange: Odd, more than 14, not equal 17, nor 19

Logger: Incrementing 31

DoubleState Received points: 34

Green: even, greater than 9 and not equal 12, nor 13

Logger: Incrementing 34

DoubleState Received points: 37

Orange: Odd, more than 14, not equal 17, nor 19

Logger: Incrementing 37

DoubleState Received points: 34

Green: even, greater than 9 and not equal 12, nor 13

Logger: Decrementing 34

DoubleState Received points: 31

Orange: Odd, more than 14, not equal 17, nor 19

Logger: Decrementing 31