
# CSE443 Object Oriented Analysis and Design

## Erchan Aptoula

## Homework 03

**Seyfullah Becerikli**

**121044028**

# PART 1

In first I created six abstract class which are for components. We have three different market type and they should have different component. So that I created three packages for markets and they contain six classes inside each of them. Every class extends from previous abstract classes which is relation them (For example: ***TRBattery*** class extends ***Battery*** class.). After that the turn for component factory. I created an interface which is ***PhoneComponentFactory***. It has six functions for creating each component. There are three factories that implements that interface and each functions return new component according to its market type.

We have three phone models and they have components which are different depend on their market. I provide that creating Phone abstract class and ***PhoneModelEnum***. So when we want to add any kind of model we can modify that abstract class and enum.

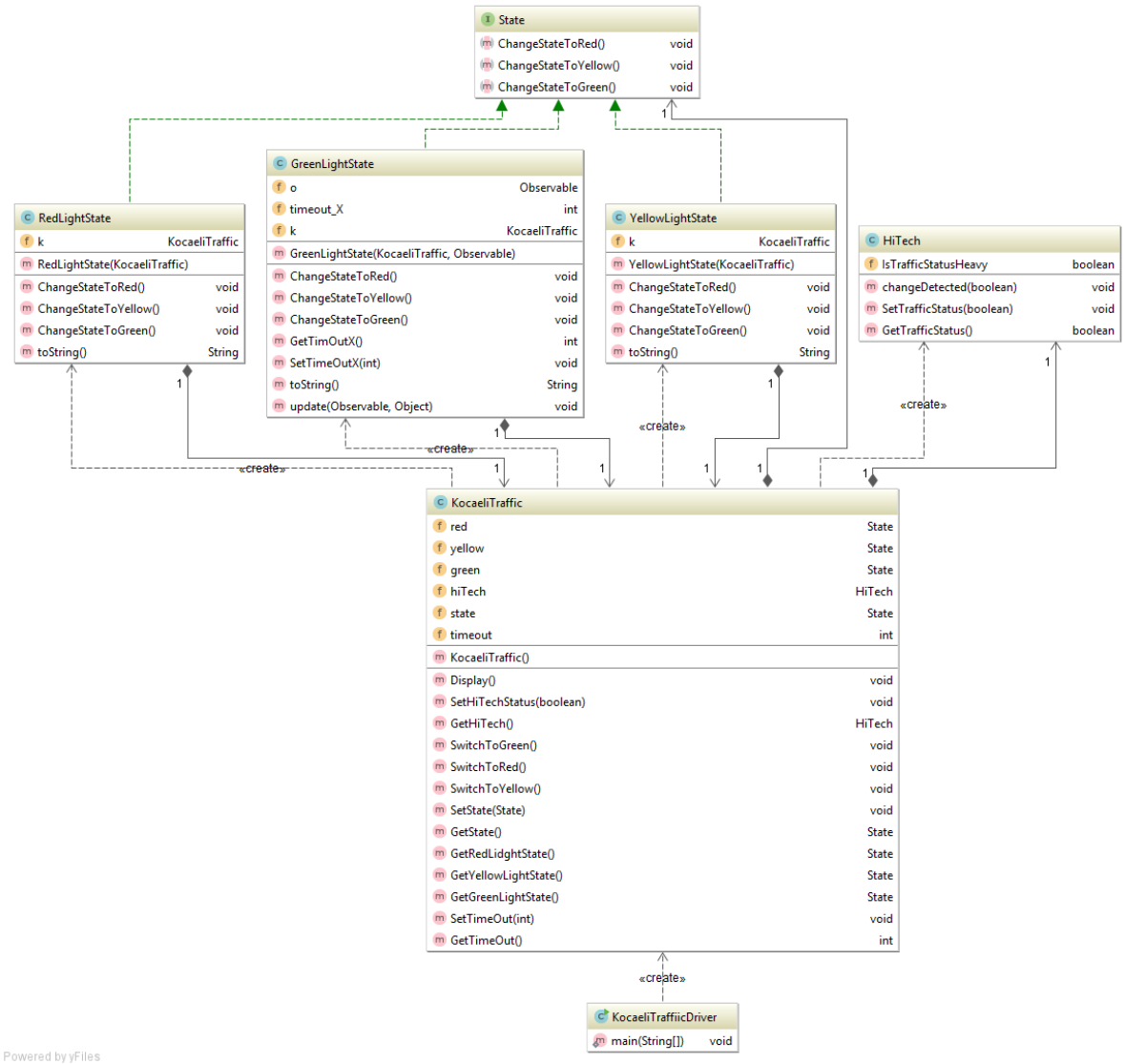
Finally, phone should be creating and take place of store. For this I have an abstract class. This class creates a phone with abstract ***CreatePhone*** function. After that we can order that phone with ***OrderAPhone*** function. Other three phone store extends from this abstract class. They decide which phone model be created inside **CreatePhone** function.

Class diagram is so big. It is inside folder.

# PART 2

I have a state interface. There is three function for changing light state. Other three class implements that interface. But they implement just one function each of them depend on the state. I used Java's Observable pattern for green state. Because I thought that only green light should changes traffic status. I implement Observer's update function inside it. There is also HiTech class and it extends from Java's Observable class. When it detects heavy traffic it notify the observers.

The class diagram is below.


State
ChangeStateToRed() void
ChangeStateToYellow() void
ChangeStateToGreen() void
GreenLightState
o Observable
timeout_X int
k KocaeliTraffic
GreenLightState(KocaeliTraffic, Observable)
ChangeStateToRed() void
ChangeStateToYellow() void
ChangeStateToGreen() void
GetTimOutX() int
SetTimeOutX(int) void
toString() String
update(Observable, Object) void
RedLightState
k KocaeliTraffic
RedLightState(KocaeliTraffic)
ChangeStateToRed() void
ChangeStateToYellow() void
ChangeStateToGreen() void
toString() String
YellowLightState
k KocaeliTraffic
YellowLightState(KocaeliTraffic)
ChangeStateToRed() void
ChangeStateToYellow() void
ChangeStateToGreen() void
toString() String
HiTech
IsTrafficStatusHeavy boolean
changeDetected(boolean) void
SetTrafficStatus(boolean) void
GetTrafficStatus() boolean
KocaeliTraffic
red State
yellow State
green State
hiTech HiTech
state State
timeout int
KocaeliTraffic()
Display() void
SetHiTechStatus(boolean) void
GetHiTech() HiTech
SwitchToGreen() void
SwitchToRed() void
SwitchToYellow() void
SetState(State) void
GetState() State
GetRedLidghtState() State
GetYellowLightState() State
GetGreenLightState() State
SetTimeOut(int) void
GetTimeOut() int
KocaeliTraffiicDriver
main(String[]) void
«create»
1
Powered by yFiles