Zoya kayani

Fa22-bse-042

Lab task

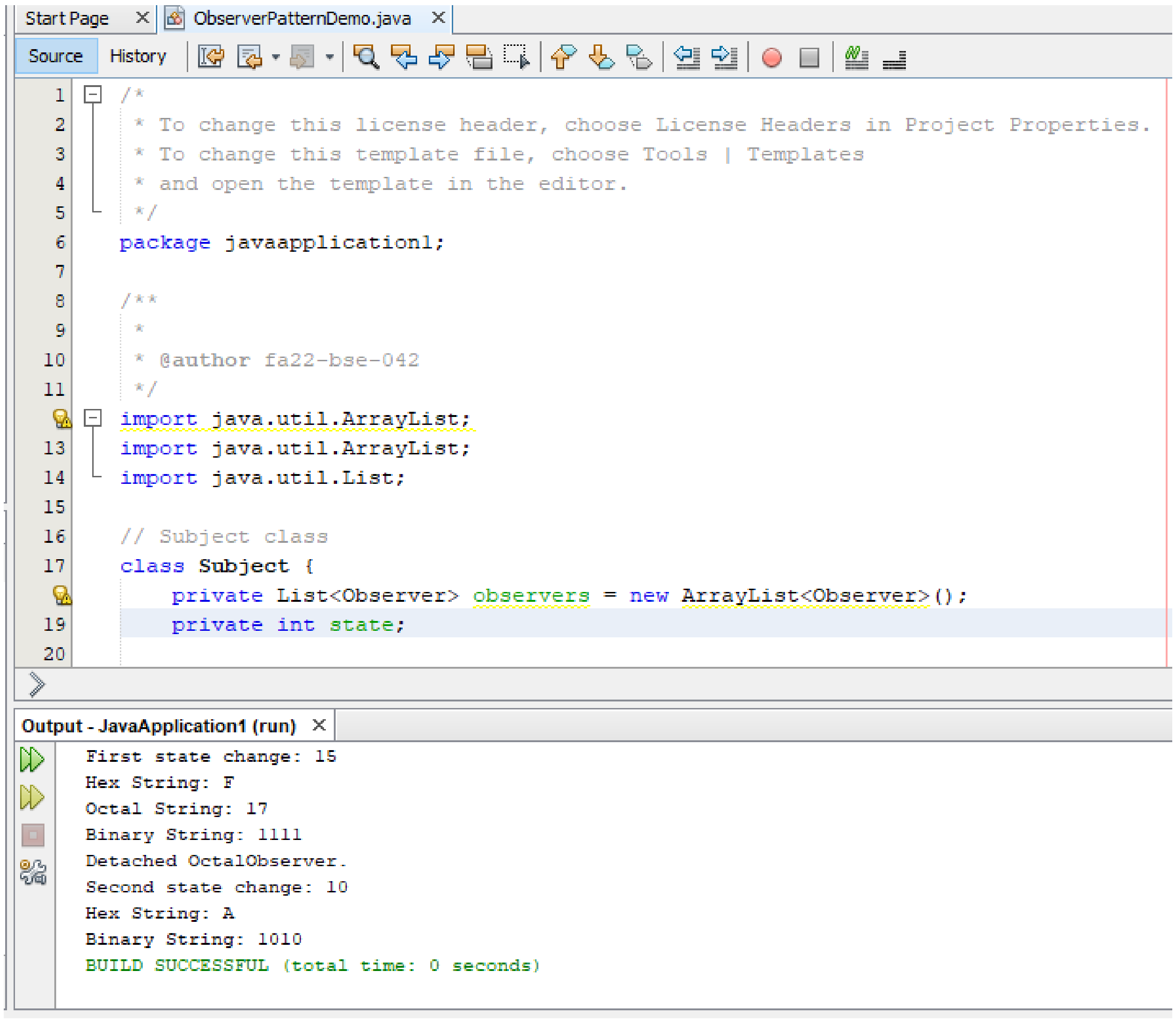
Question 1:

1. add detach observer 0 method to the subject

2. modify pattern use in observerpatterndemo such that you change observer and then update the state only active observers are notified

```
package javaapplication1;


/**
 *
 * @author fa22-bse-042
 */
import java.util.ArrayList;
import java.util.ArrayList;
import java.util.List;


// Subject class
class Subject {
   private List<Observer> observers = new ArrayList<Observer>();
   private int state;


   public int getState() {
      return state;
   }


   public void setState(int state) {
        this.state = state;
        notifyAllObservers();
    }

    public void attach(Observer observer) {
        observers.add(observer);
    }

    public void detach(Observer observer) {
        observers.remove(observer);
    }

    public void notifyAllObservers() {
        for (Observer observer : observers) {
            observer.update();
        }
    }
}

// Observer class
abstract class Observer {
    protected Subject subject;
    public abstract void update();
}

// Concrete observer classes
class BinaryObserver extends Observer {
    public BinaryObserver(Subject subject) {
        this.subject = subject;
```

```
      this.subject.attach(this);
   }

   @Override
   public void update() {
      System.out.println("Binary String: " + Integer.toBinaryString(subject.getState()));
   }
}

class OctalObserver extends Observer {
   public OctalObserver(Subject subject) {
      this.subject = subject;
      this.subject.attach(this);
   }

   @Override
   public void update() {
      System.out.println("Octal String: " + Integer.toOctalString(subject.getState()));
   }
}

class HexaObserver extends Observer {
   public HexaObserver(Subject subject) {
      this.subject = subject;
      this.subject.attach(this);
   }

   @Override
   public void update() {
      System.out.println("Hex String: " + Integer.toHexString(subject.getState()).toUpperCase());
   }
}

// Demo class
public class ObserverPatternDemo {
   public static void main(String[] args) {
      Subject subject = new Subject();

      HexaObserver hexaObserver = new HexaObserver(subject);
      OctalObserver octalObserver = new OctalObserver(subject);
      BinaryObserver binaryObserver = new BinaryObserver(subject);

      System.out.println("First state change: 15");
      subject.setState(15);

      // Detach the octal observer
      subject.detach(octalObserver);
      System.out.println("Detached OctalObserver.");

      System.out.println("Second state change: 10");
      subject.setState(10);  // Only HexaObserver and BinaryObserver should be notified
   }
}
```



```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package javaapplication1;

/**
 *
 * @author fa22-bse-042
 */
import java.util.ArrayList;
import java.util.ArrayList;
import java.util.List;

// Subject class
class Subject {
    private List<Observer> observers = new ArrayList<Observer>();
    private int state;

```

```
First state change: 15
Hex String: F
Octal String: 17
Binary String: 1111
Detached OctalObserver.
Second state change: 10
Hex String: A
Binary String: 1010
BUILD SUCCESSFUL (total time: 0 seconds)
```

Question 2

in the same project create another package obsever.eventsmanagement and add another smssupport listener that when notified will check if default sms length is more than 160 character in the class then give a warning to define valid default sms othervise just send a sms to the phone number

code:

import java.util.ArrayList;

import java.util.List;

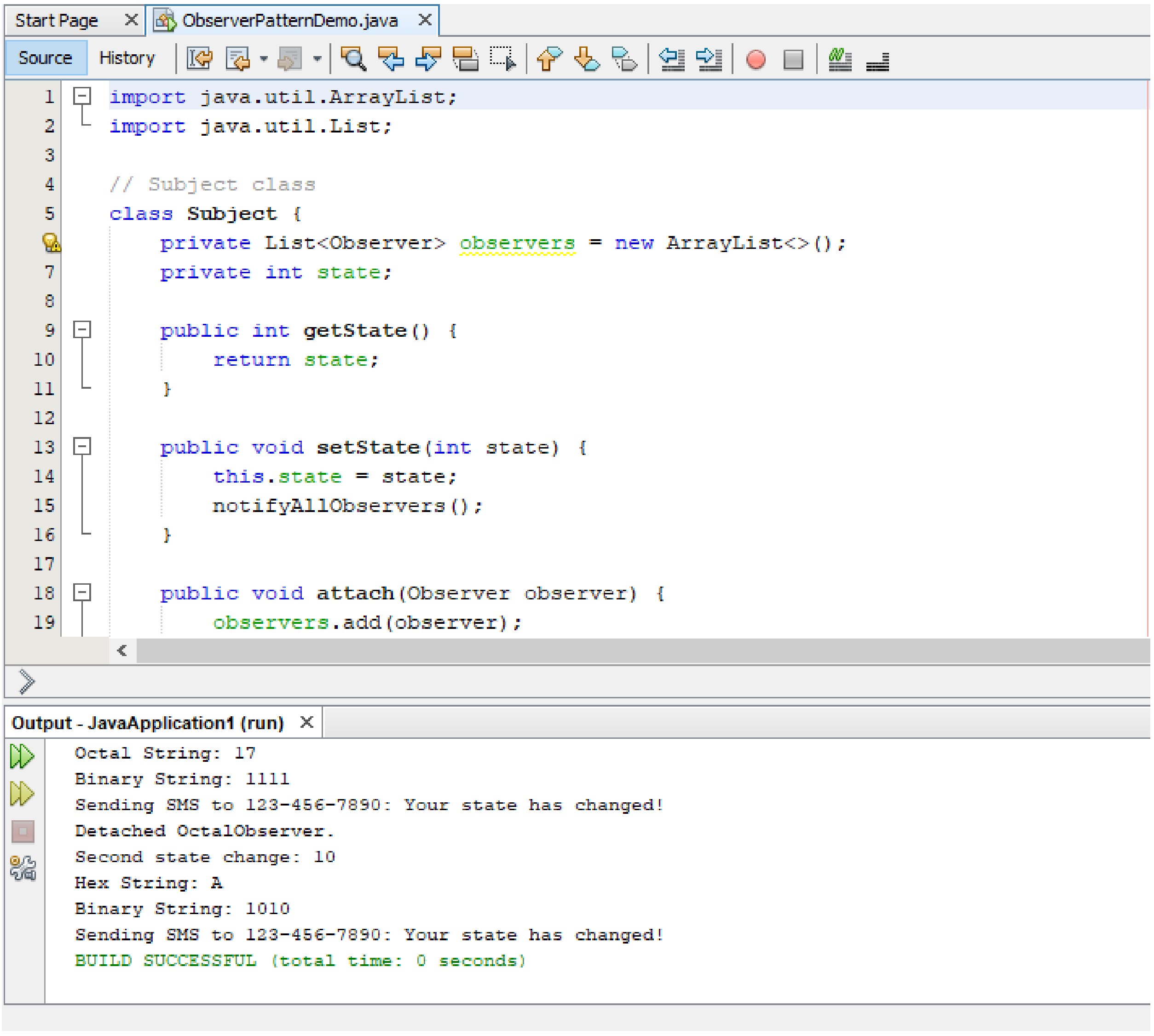
// Subject class

```
class Subject {
    private List<Observer> observers = new ArrayList<>();
    private int state;

    public int getState() {
        return state;
    }

    public void setState(int state) {
        this.state = state;
        notifyAllObservers();
    }

    public void attach(Observer observer) {
        observers.add(observer);
    }

    public void detach(Observer observer) {
        observers.remove(observer);
    }

    public void notifyAllObservers() {
        for (Observer observer : observers) {
            observer.update();
        }
    }
}

// Observer class
```

```
abstract class Observer {
   protected Subject subject;
   public abstract void update();
}

// Concrete observer classes
class BinaryObserver extends Observer {
   public BinaryObserver(Subject subject) {
      this.subject = subject;
      this.subject.attach(this);
   }

   @Override
   public void update() {
      System.out.println("Binary String: " + Integer.toBinaryString(subject.getState()));
   }
}

class OctalObserver extends Observer {
   public OctalObserver(Subject subject) {
      this.subject = subject;
      this.subject.attach(this);
   }

   @Override
   public void update() {
      System.out.println("Octal String: " + Integer.toOctalString(subject.getState()));
   }
}
```

```
class HexaObserver extends Observer {
    public HexaObserver(Subject subject) {
        this.subject = subject;
        this.subject.attach(this);
    }

    @Override
    public void update() {
        System.out.println("Hex String: " + Integer.toHexString(subject.getState()).toUpperCase());
    }
}

// SMS Support Listener
class SmsSupportListener extends Observer {
    private static final int MAX_SMS_LENGTH = 160;
    private String phoneNumber = "123-456-7890"; // Example phone number
    private String defaultSms = "Your state has changed!"; // Example SMS content

    public SmsSupportListener(Subject subject) {
        this.subject = subject;
        this.subject.attach(this); // Attach this listener to the subject
    }

    @Override
    public void update() {
        if (defaultSms.length() > MAX_SMS_LENGTH) {
            System.out.println("Warning: Default SMS length exceeds 160 characters. Please define a valid
default SMS.");
```

```
        } else {
            sendSms(phoneNumber, defaultSms);
        }
    }

    private void sendSms(String phoneNumber, String message) {
        System.out.println("Sending SMS to " + phoneNumber + ": " + message);
    }
}

// Demo class
public class ObserverPatternDemo {
    public static void main(String[] args) {
        Subject subject = new Subject();

        HexaObserver hexaObserver = new HexaObserver(subject);
        OctalObserver octalObserver = new OctalObserver(subject);
        BinaryObserver binaryObserver = new BinaryObserver(subject);
        SmsSupportListener smsListener = new SmsSupportListener(subject); // Create and attach SMS
listener

        System.out.println("First state change: 15");
        subject.setState(15);

        // Detach the octal observer
        subject.detach(octalObserver);
        System.out.println("Detached OctalObserver.");

        System.out.println("Second state change: 10");
        subject.setState(10);  // Only HexaObserver, BinaryObserver, and SmsSupportListener should be
notified

    }

}
```



```
import java.util.ArrayList;
import java.util.List;

// Subject class
class Subject {
    private List<Observer> observers = new ArrayList<>();
    private int state;

    public int getState() {
        return state;
    }

    public void setState(int state) {
        this.state = state;
        notifyAllObservers();
    }

    public void attach(Observer observer) {
        observers.add(observer);
```

```
Octal String: 17
Binary String: 1111
Sending SMS to 123-456-7890: Your state has changed!
Detached OctalObserver.
Second state change: 10
Hex String: A
Binary String: 1010
Sending SMS to 123-456-7890: Your state has changed!
BUILD SUCCESSFUL (total time: 0 seconds)
```