



**COMSATS UNIVERSITY ISLAMABAD,
ABBOTTABAD**

Design Pattern

Lab assignment # 02

Submitted by:

Laiba Binta Tahir FA21-BSE-019

Submitted to:

Mam Faiza Hameed

Contents

Facade Pattern	3
Code.....	3
Class diagram	6
Prototype pattern	7
Code.....	7
Class Diagram.....	10
Proxy pattern	11
Code.....	11
Class Diagram.....	13
Observer pattern	13
Code.....	13
Class Diagram.....	16

Facade Pattern

Example: home automation with multiple systems.

Code

```
package org.example;

public class TV {
    public void turnOn() { System.out.println("TV is ON."); }
    public void turnOff() { System.out.println("TV is OFF."); }
}
```

```
package org.example;

public class Lights {
    public void on() { System.out.println("Lights are ON."); }
    public void off() { System.out.println("Lights are OFF."); }
}
```

```
package org.example;

public class AC {
    public void start() { System.out.println("AC is running."); }
    public void stop() { System.out.println("AC is stopped."); }
}
```

```
package org.example;


class HomeAutomationFacade {
    private Lights lights;
    private AC ac;
    private TV tv;

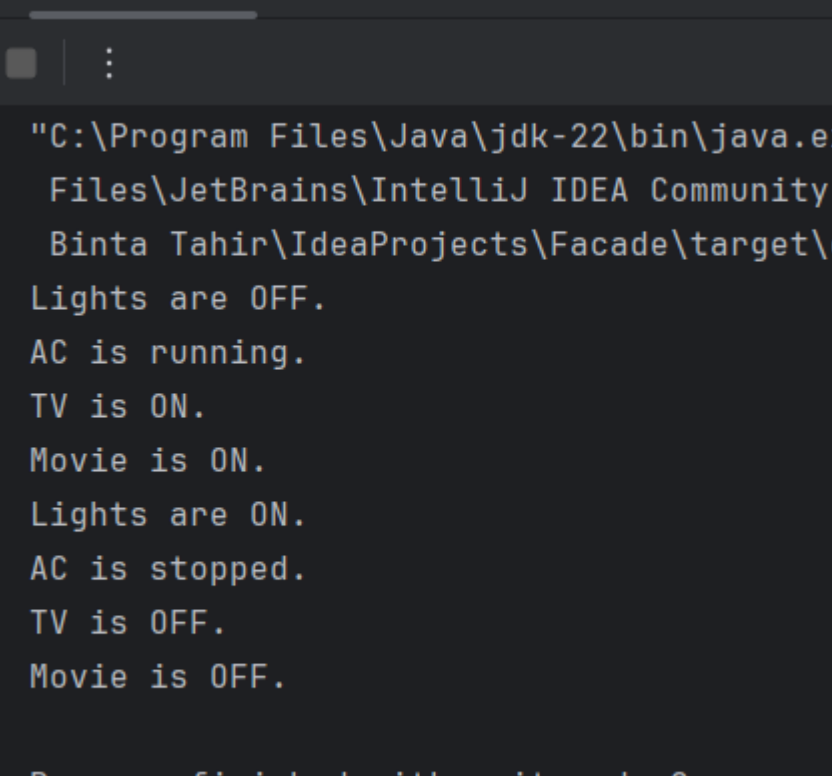
    public HomeAutomationFacade(Lights lights, AC ac, TV tv) {
        this.lights = lights;
        this.ac = ac;
        this.tv = tv;
    }

    public void startMovieMode() {
        lights.off();
        ac.start();
        tv.turnOn();
        System.out.println("Movie is ON.");
    }

    public void stopMovieMode() {
        lights.on();
        ac.stop();
        tv.turnOff();
        System.out.println("Movie is OFF.");
    }
}
```

Main.java × HomeAutomationFacade.java TV.java Lights.java AC.java

```
1 package org.example;
2
3
4 To Run code, press Shift F10 or click the  icon in the gutter.
5 public class Main {
6     public static void main(String[] args) {
7         Lights lights = new Lights();
8         AC ac = new AC();
9         TV tv = new TV();
10
11         HomeAutomationFacade facade = new HomeAutomationFacade(lights, ac, tv);
12         facade.startMovieMode();
13         facade.stopMovieMode();
14     }
15 }
```



```
Run Main x
"C:\Program Files\Java\jdk-22\bin\java.exe" "-
Files\JetBrains\IntelliJ IDEA Community Editi
Binta Tahir\IdeaProjects\Facade\target\classe
Lights are OFF.
AC is running.
TV is ON.
Movie is ON.
Lights are ON.
AC is stopped.
TV is OFF.
Movie is OFF.

Process finished with exit code 0
```

Class diagram

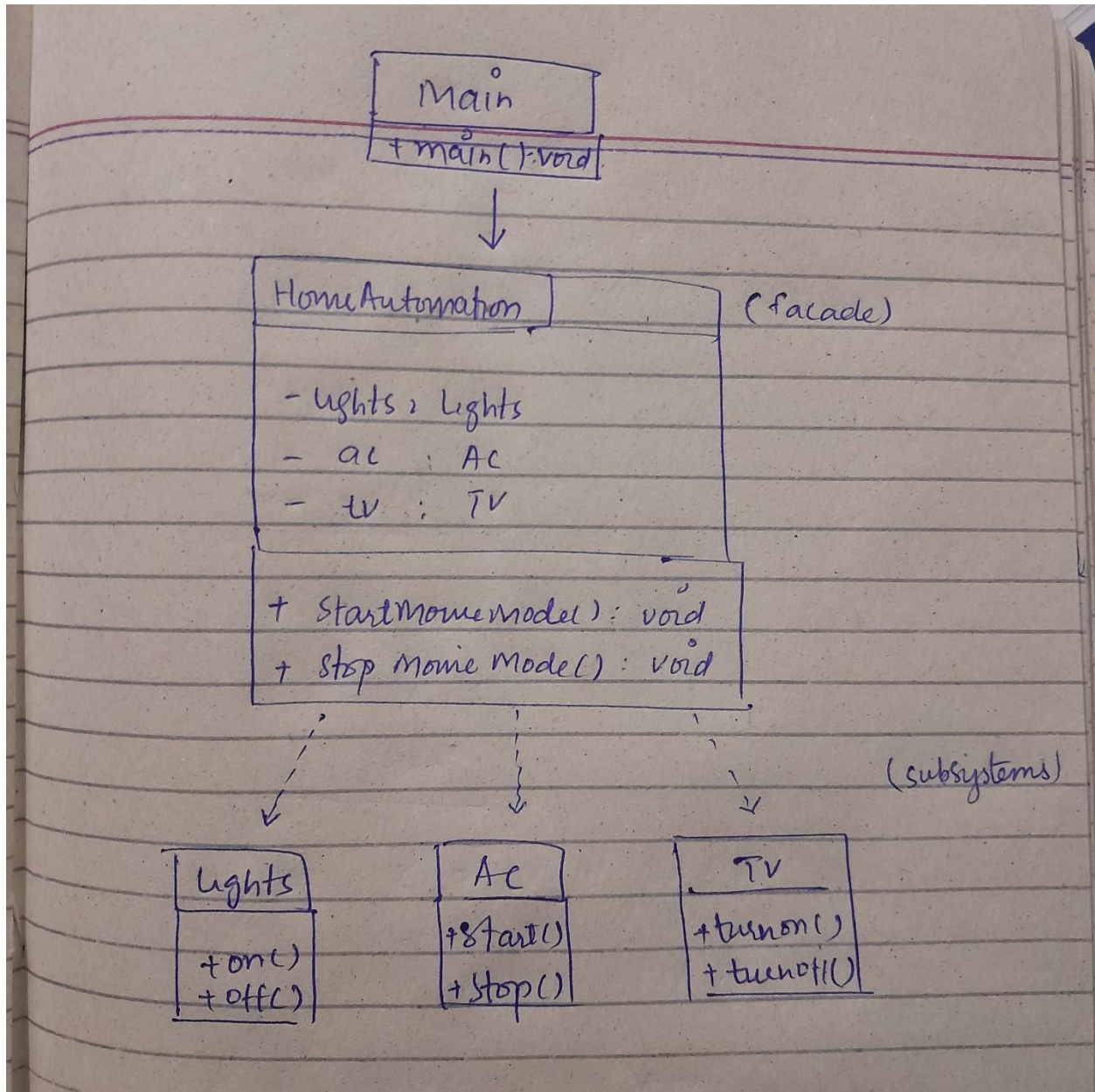
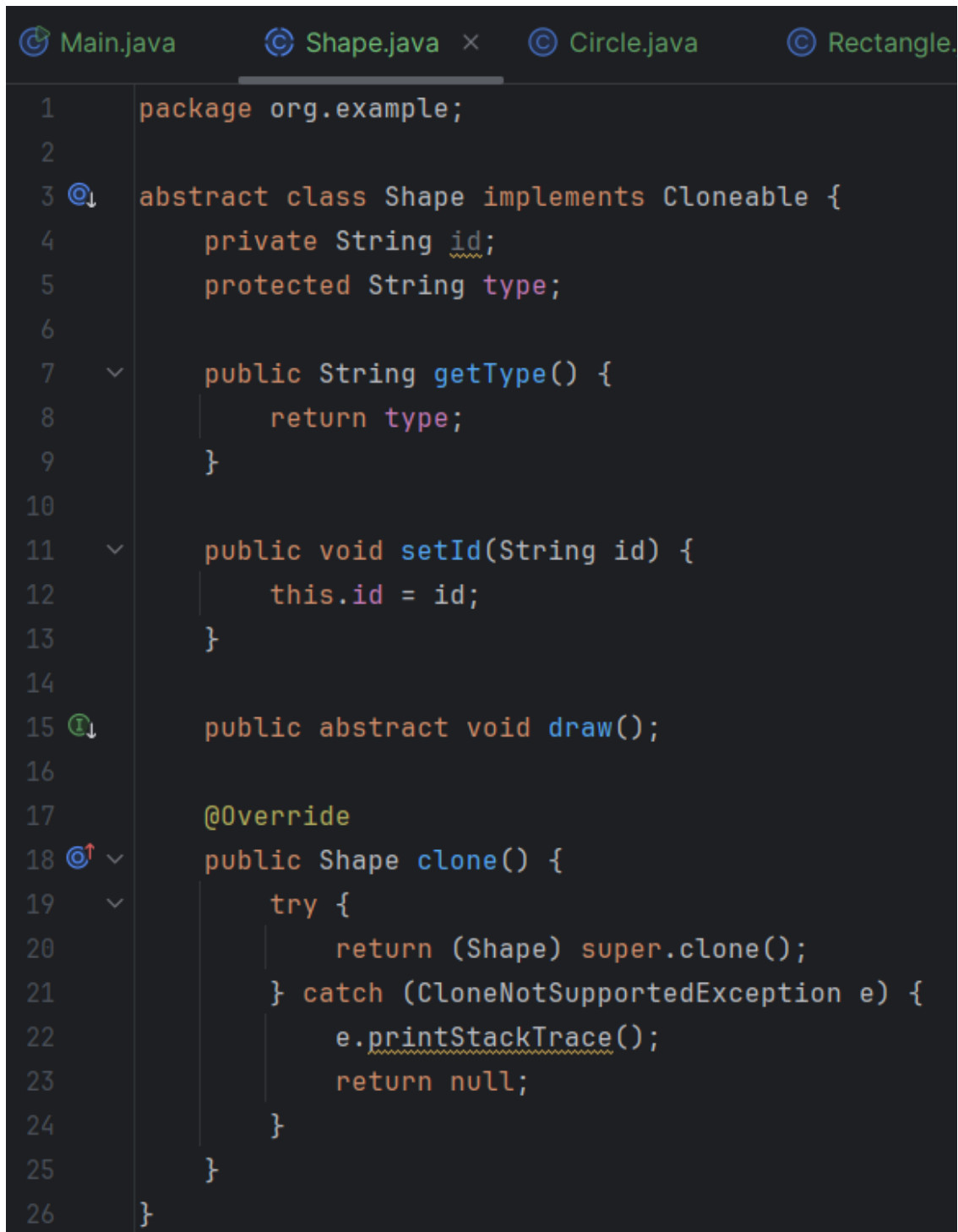


Figure 1 Facade pattern

Prototype pattern

Example: A Shape class where cloning is required.

Code



```
1 package org.example;
2
3 @abstract class Shape implements Cloneable {
4     private String id;
5     protected String type;
6
7     public String getType() {
8         return type;
9     }
10
11     public void setId(String id) {
12         this.id = id;
13     }
14
15     public abstract void draw();
16
17     @Override
18     public Shape clone() {
19         try {
20             return (Shape) super.clone();
21         } catch (CloneNotSupportedException e) {
22             e.printStackTrace();
23             return null;
24         }
25     }
26 }
```

```
Main.java x Shape.java Circle.java Rectangle.java pom.xml (Prototype)

1 package org.example;
2
3 | To Run code, press Shift F10 or click the ▶ icon in the gutter.
4
5 ▶ public class Main {
6 ▶     public static void main(String[] args) {
7         Circle circle = new Circle();
8         circle.setId("1");
9
10        Circle clonedCircle = (Circle) circle.clone();
11        System.out.println("Cloned Shape: " + clonedCircle.getType());
12        clonedCircle.draw();
13
14        Rectangle rectangle = new Rectangle();
15        rectangle.setId("2");
16
17        Rectangle clonedRectangle = (Rectangle) rectangle.clone();
18        System.out.println("Cloned Shape: " + clonedRectangle.getType());
19        clonedRectangle.draw();
20    }
21 }
```

```
Main.java Shape.java Circle.java Rectangle.java
1 package org.example;
2
3 // Concrete Prototype 1
4 class Circle extends Shape {
5     public Circle() {
6         type = "Circle";
7     }
8
9     @Override
10    public void draw() {
11        System.out.println("Drawing a Circle.");
12    }
13 }
```



```
Main.java Shape.java Circle.java Rectangle.java x
1 package org.example;
2
3 // Concrete Prototype 2
4 class Rectangle extends Shape {
5     public Rectangle() {
6         type = "Rectangle";
7     }
8
9     @Override
10    public void draw() {
11        System.out.println("Drawing a Rectangle.");
12    }
13 }
```

```
↑ "C:\Program Files\Java\jdk-22\bin\java.exe"
↓ Files\JetBrains\IntelliJ IDEA Community Edi
  Binta Tahir\IdeaProjects\Prototype\target\c
  Cloned Shape: Circle
  Drawing a Circle.
  Cloned Shape: Rectangle
  Drawing a Rectangle.
  Process finished with exit code 0
```

Class Diagram

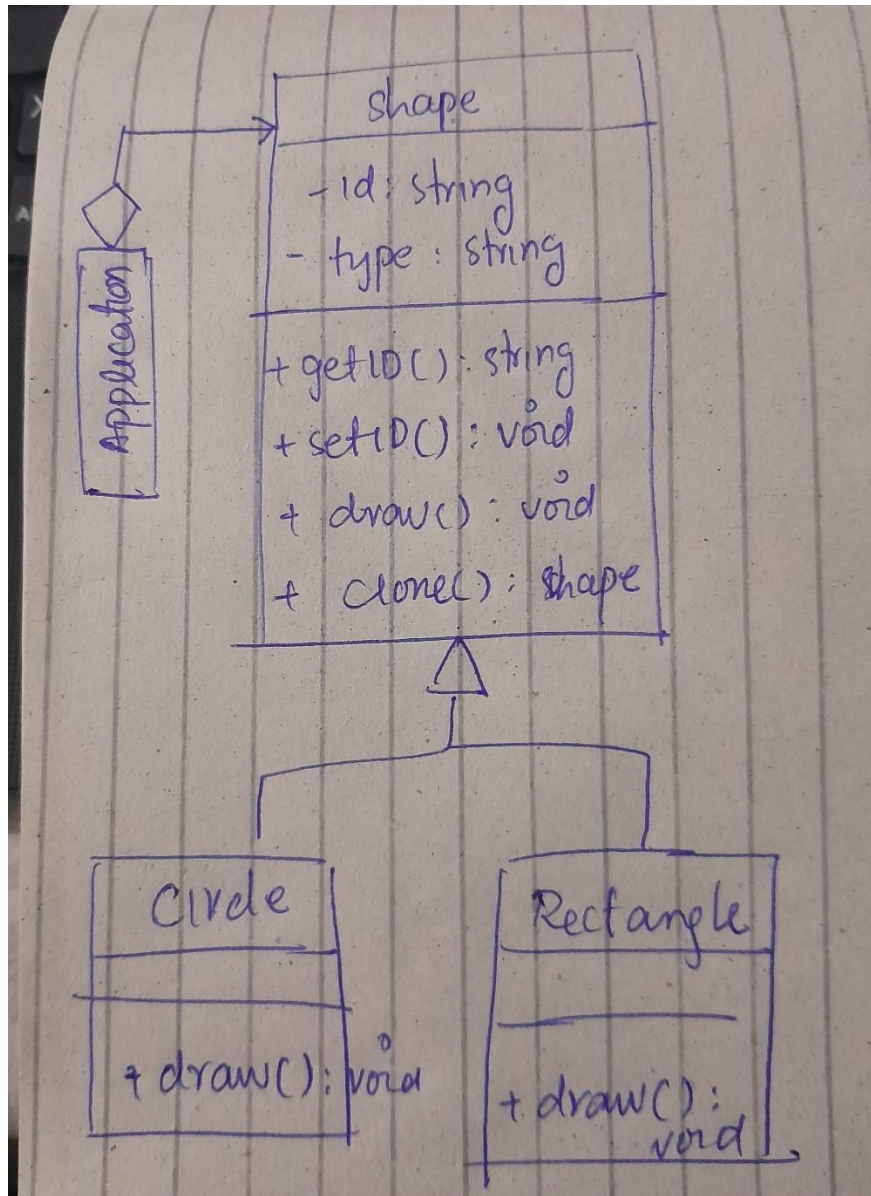


Figure 2 Prototype pattern

Proxy pattern

Example: A Proxy for accessing a Remote Server.

Code

```
Main.java pom.xml (Proxy_pattern)

1 package org.example;
2
3 public interface Server {
4     void request();
5 }
```

```
package org.example;


class RealServer implements Server {
    @Override
    public void request() {
        System.out.println("Processing request on Real Server...");
    }
}
```

```
package org.example;

class ProxyServer implements Server {
    private RealServer realServer;




    @Override
    public void request() {
        if (realServer == null) {
            realServer = new RealServer();
        }
        System.out.println("Proxy handling some checks before passing to Real Server...");
        realServer.request();
    }
}
```

```
1 package org.example;
2
3
4
5
6
7
8
9
10
```

To Run code, press **Shift** **F10** or click the  icon in the gutter.

```
11 public class Main {
12     public static void main(String[] args) {
13         Server proxy = new ProxyServer();
14         proxy.request();
15     }
16 }
```

Run Main x

↑ "C:\Program Files\Java\jdk-22\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.1.2\bin" -Dfile.encoding=Binta Tahir\IdeaProjects\Proxy_pattern\target\classes" org.example.Main

↓

⇌ Proxy handling some checks before passing to Real Server...

⇓ Processing request on Real Server...

🖨 Process finished with exit code 0

🗑

Class Diagram

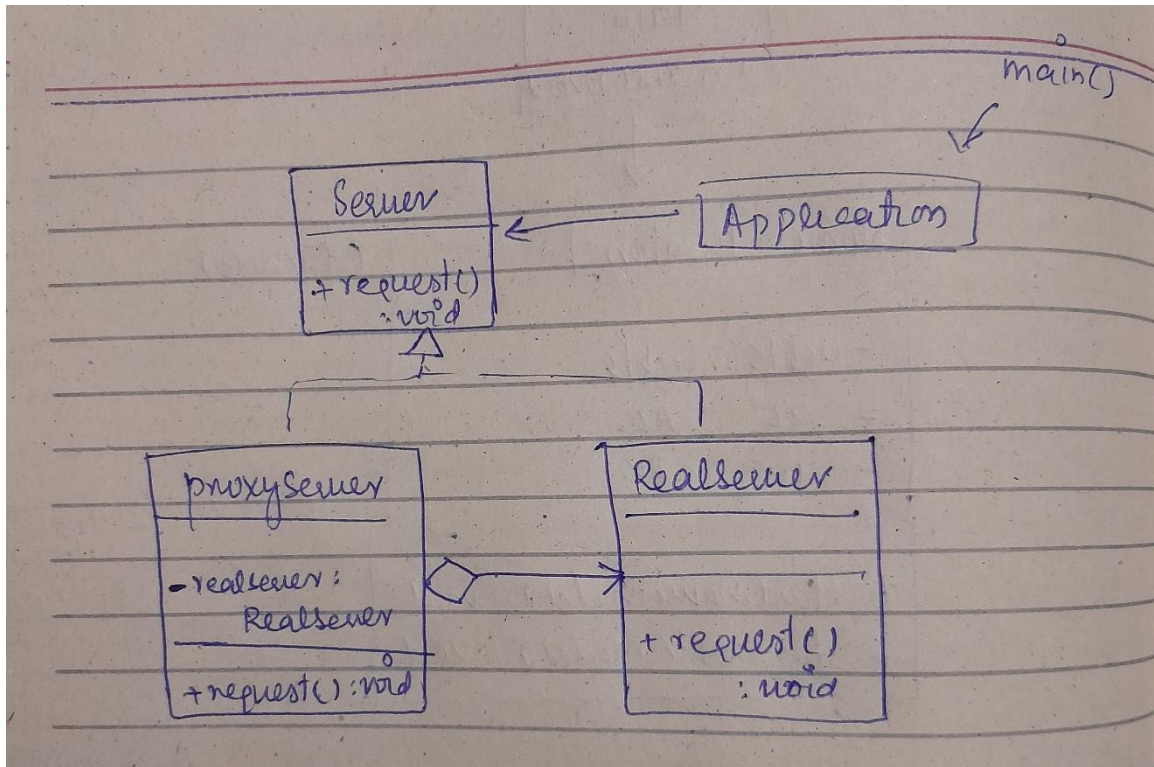


Figure 3 proxy pattern

Observer pattern

Example: A News Publisher (subject) notifies Subscribers (observers) when there's a new article.

Code

```
1 package org.example;  
2 interface Subscriber {  
3     void update(String news);  
4 }  
5
```

Main.java NewsPublisher.java × Subscriber.java EmailSubscriber.java

```
1 package org.example;
2 import java.util.ArrayList;
3 import java.util.List;
4
5 class NewsPublisher {
6     private List<Subscriber> subscribers = new ArrayList<>();
7     private String latestNews;
8
9     public void subscribe(Subscriber subscriber) {
10         subscribers.add(subscriber);
11     }
12
13     public void unsubscribe(Subscriber subscriber) {
14         subscribers.remove(subscriber);
15     }
16
17     public void setNews(String news) {
18         this.latestNews = news;
19         notifySubscribers();
20     }
21
22     private void notifySubscribers() {
23         for (Subscriber subscriber : subscribers) {
24             subscriber.update(latestNews);
25         }
26     }
27 }
```

```
package org.example;

// Concrete Observer
class EmailSubscriber implements Subscriber {
    private String email;

    public EmailSubscriber(String email) {
        this.email = email;
    }

    @Override
    public void update(String news) {
        System.out.println("Email sent to " + email + ": " + news);
    }
}
```

```
package org.example;
public class Main {
    public static void main(String[] args) {
        NewsPublisher publisher = new NewsPublisher();

        Subscriber sub1 = new EmailSubscriber("laiba@gmail.com");
        Subscriber sub2 = new EmailSubscriber("warda@gmail.com");

        publisher.subscribe(sub1);
        publisher.subscribe(sub2);

        publisher.setNews("Observer Pattern Implemented!");
    }
}
```



```
Run Main x
C:\Program Files\Java\jdk-22\bin\java.exe "-javaagent:C:\Program Files\JetBrains\JetBrains\IntelliJ IDEA Community Edition 2024.1.2\bin" -Dfile.encoding=UTF-8
Binta Tahir\IdeaProjects\ObserverPattern\target\classes" org.example.Main
Email sent to laiba@gmail.com: Observer Pattern Implemented!
Email sent to warda@gmail.com: Observer Pattern Implemented!
Process finished with exit code 0
```

Class Diagram

