Behavioral Design Pattern

By Raza Sikander
Project Engineer
CDAC Hyderabad

Behavioral Design Pattern

- Behavioral design patterns are concerned with the interaction and responsibility of objects.
- In these design patterns,the interaction between the objects should be in such a way that they can easily talk to each other and still should be loosely coupled.

Behavioral Design Pattern

- Template Pattern
- Mediator Pattern
- Observer Pattern

Template Pattern

just define the skeleton of a function in an operation, deferring some steps to its subclasses

Benefits:

- It is very common technique for reusing the code.This is only the main benefit of it.

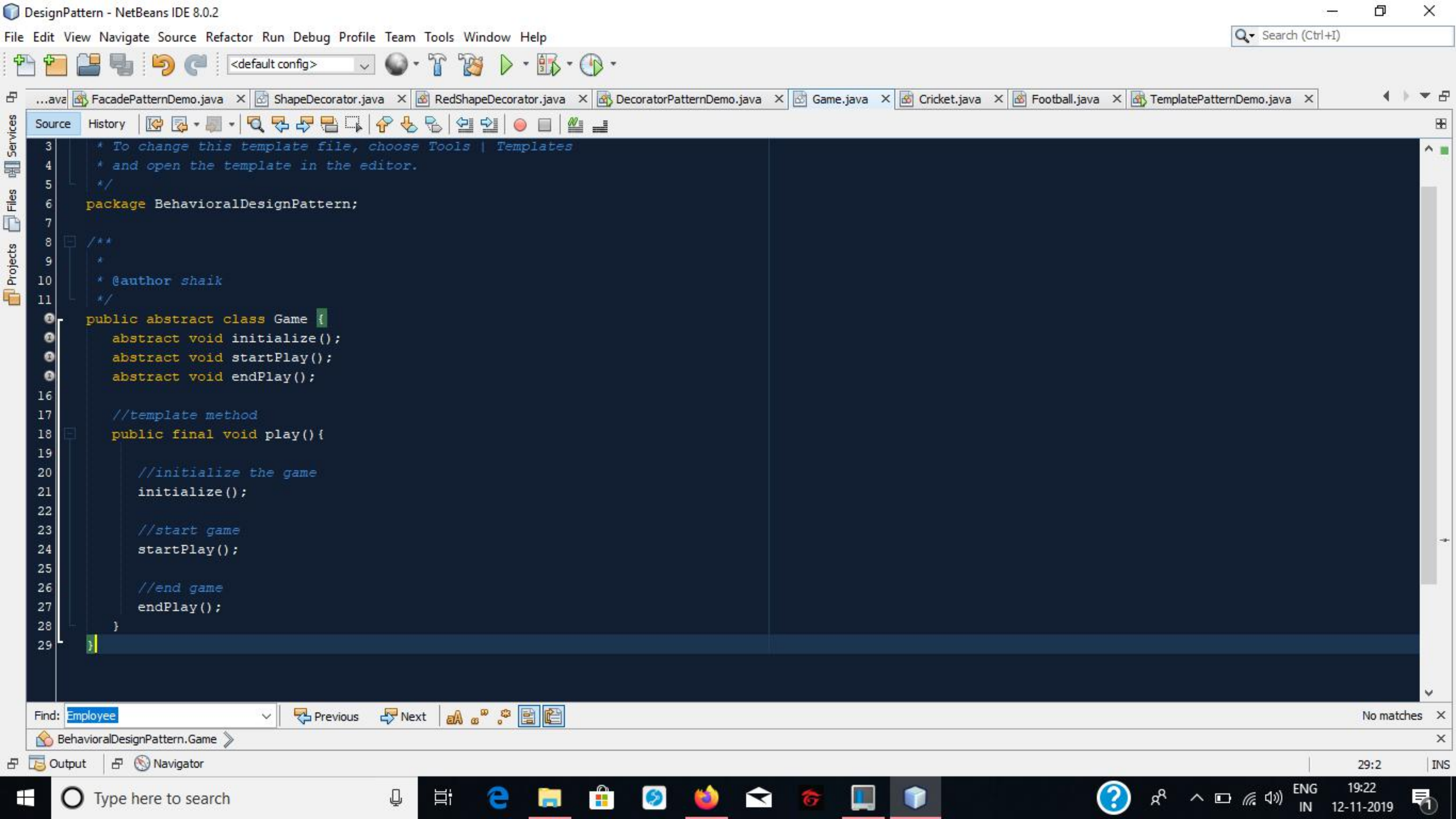
Template Pattern

Usage:

- It is used when the common behavior among sub-classes should be moved to a single common class by avoiding the duplication.

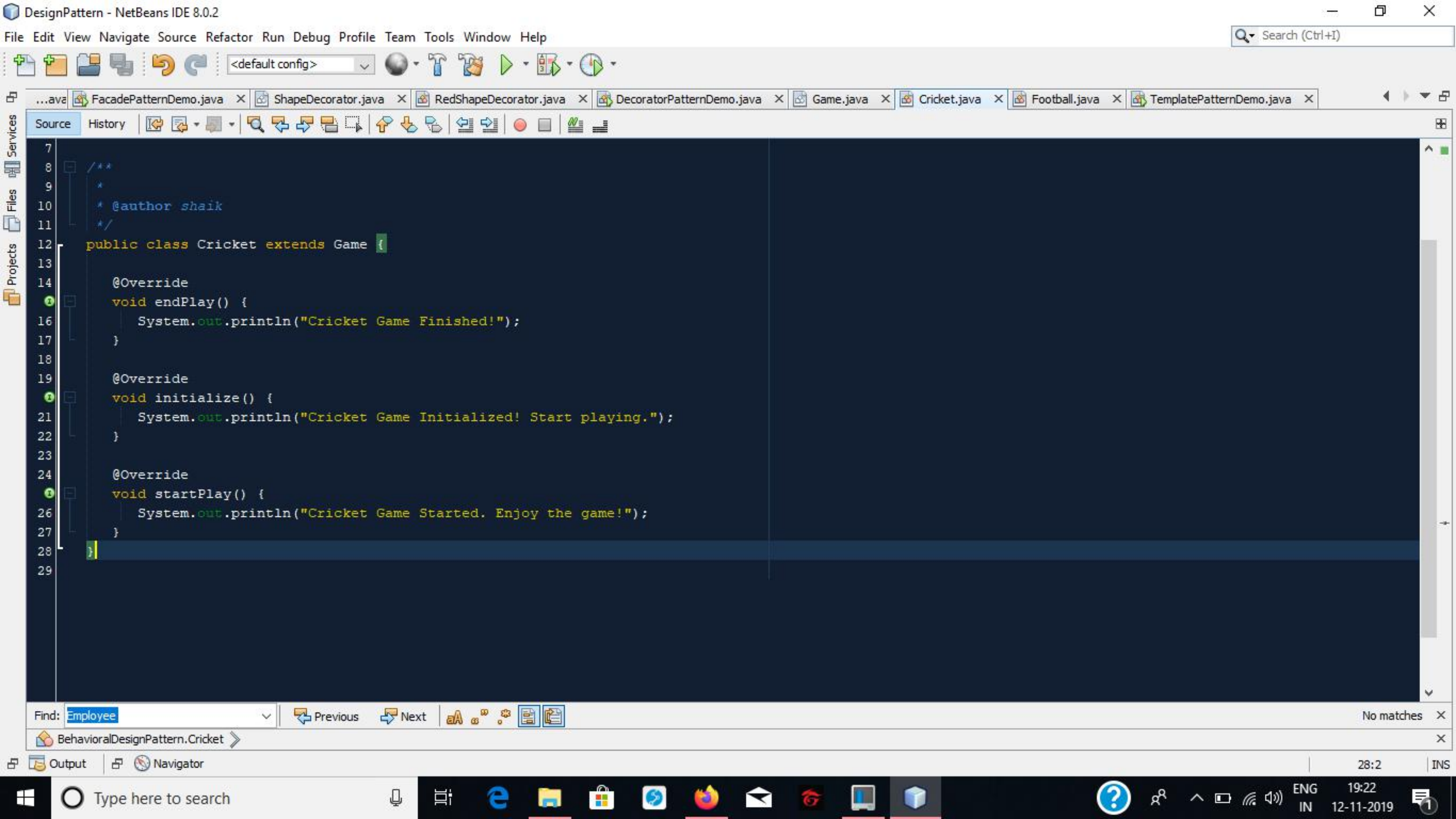
Template Pattern

Implementation of Template Design Pattern

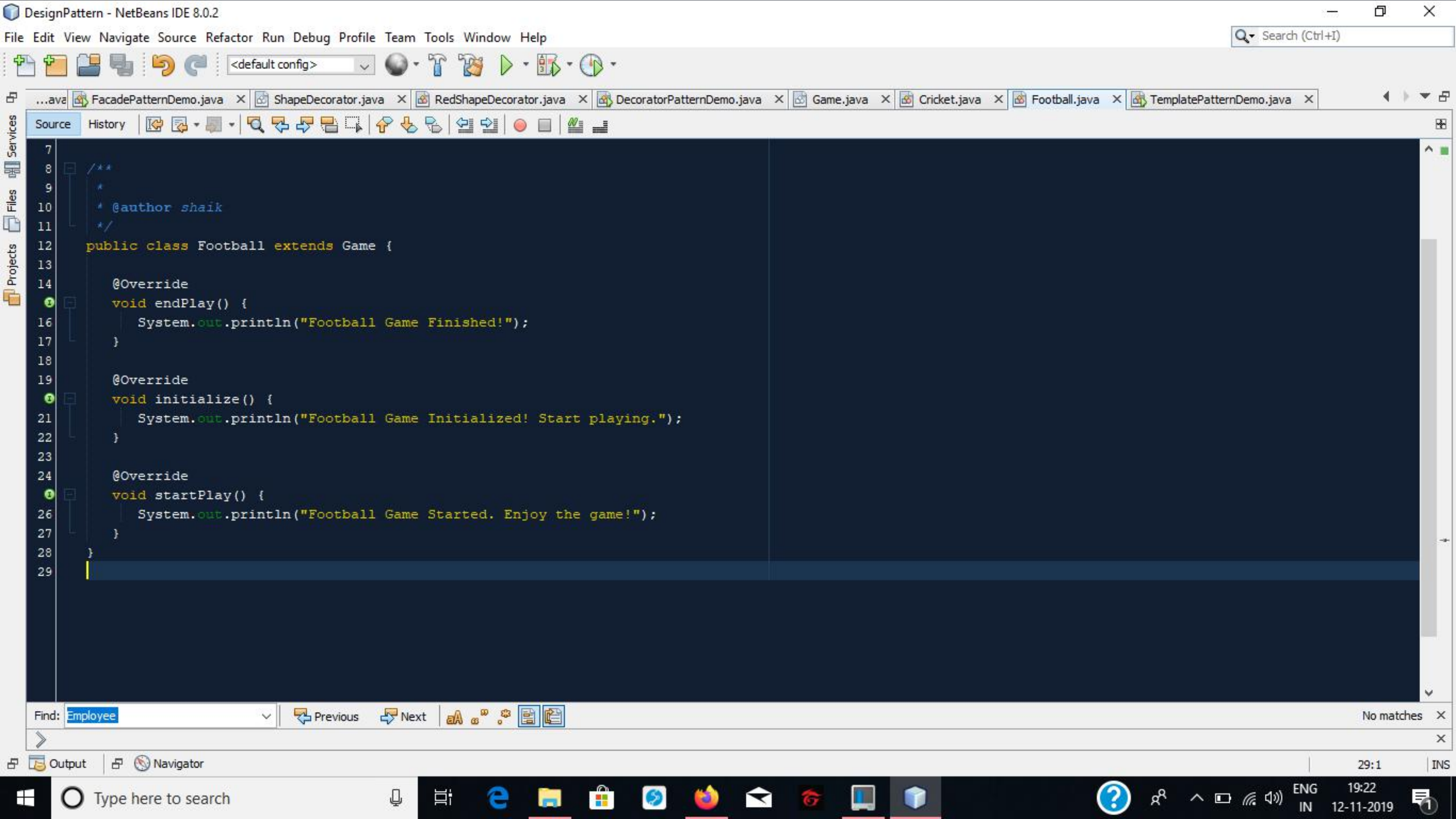

DesignPattern - NetBeans IDE 8.0.2

```java
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package BehavioralDesignPattern;

/**
 *
 * @author shaik
 */
public abstract class Game {
    abstract void initialize();
    abstract void startPlay();
    abstract void endPlay();

    //template method
    public final void play(){

        //initialize the game
        initialize();

        //start game
        startPlay();

        //end game
        endPlay();
    }
}
```

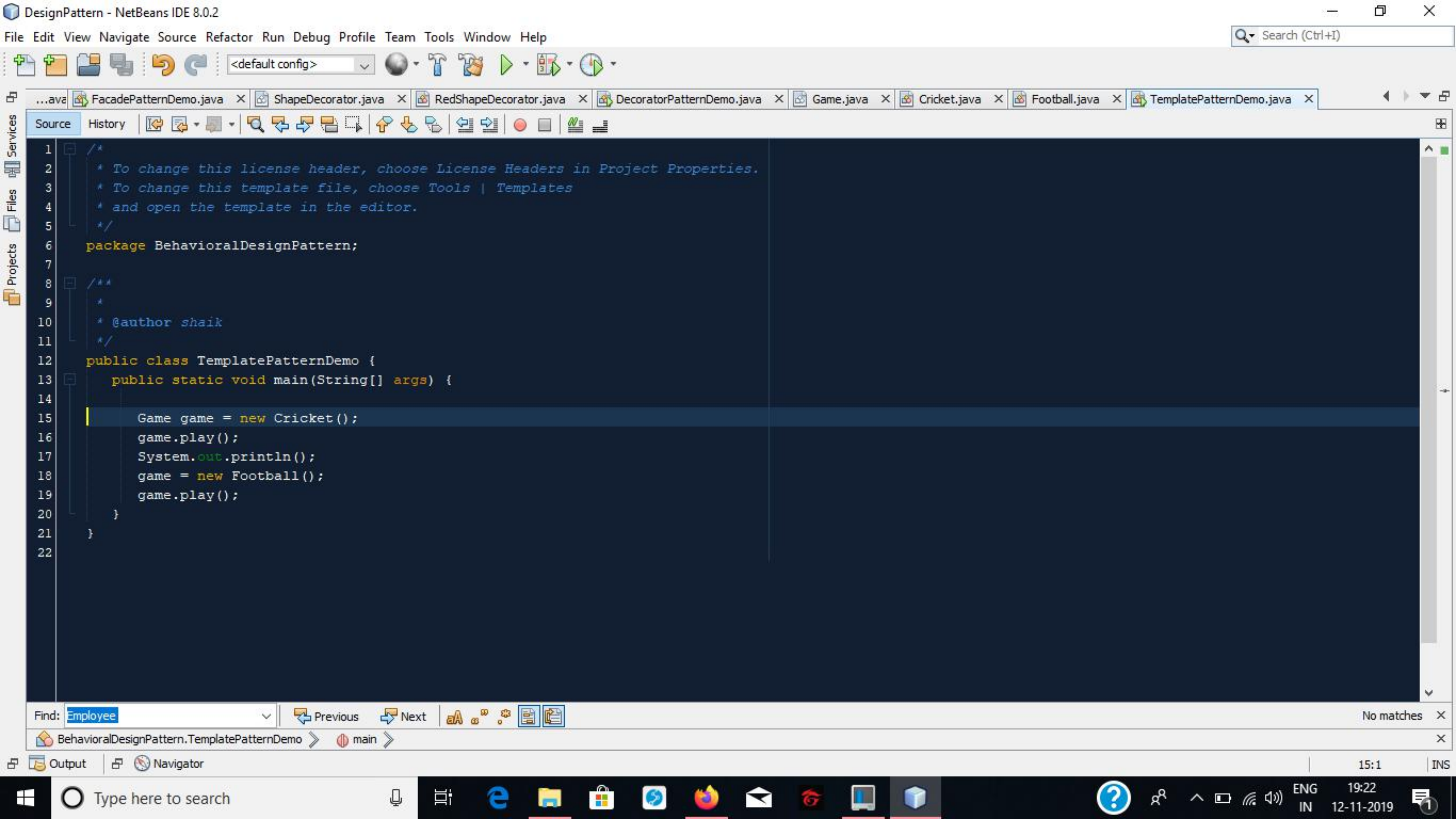
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
...ava FacadePatternDemo.java ShapeDecorator.java RedShapeDecorator.java DecoratorPatternDemo.java Game.java Cricket.java Football.java TemplatePatternDemo.java
Source History
Find: Employee
Previous Next
No matches
BehavioralDesignPattern.Game
Output Navigator
29:2 INS
Type here to search
ENG IN 19:22 12-11-2019


DesignPattern - NetBeans IDE 8.0.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
FacadePatternDemo.java
ShapeDecorator.java
RedShapeDecorator.java
DecoratorPatternDemo.java
Game.java
Cricket.java
Football.java
TemplatePatternDemo.java
Source
History

```java
/**
 *
 * @author shaik
 */
public class Cricket extends Game {

    @Override
    void endPlay() {
        System.out.println("Cricket Game Finished!");
    }

    @Override
    void initialize() {
        System.out.println("Cricket Game Initialized! Start playing.");
    }

    @Override
    void startPlay() {
        System.out.println("Cricket Game Started. Enjoy the game!");
    }
}
```



```
/**
 *
 * @author shaik
 */
public class Football extends Game {

    @Override
    void endPlay() {
        System.out.println("Football Game Finished!");
    }

    @Override
    void initialize() {
        System.out.println("Football Game Initialized! Start playing.");
    }

    @Override
    void startPlay() {
        System.out.println("Football Game Started. Enjoy the game!");
    }
}
```



```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 BehavioralDesignPattern;

/**
 *
 * @author shaik
 */
public class TemplatePatternDemo {
    public static void main(String[] args) {

        Game game = new Cricket();
        game.play();
        System.out.println();
        game = new Football();
        game.play();
    }
}

```

DesignPattern - NetBeans IDE 8.0.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
...ava FacadePatternDemo.java ShapeDecorator.java RedShapeDecorator.java DecoratorPatternDemo.java Game.java Cricket.java Football.java TemplatePatternDemo.java
Source History
Find: Employee Previous Next No matches
BehavioralDesignPattern.TemplatePatternDemo main
Output Navigator
15:1 INS
Type here to search
ENG IN 19:22 12-11-2019

Mediator Pattern

"to define an object that encapsulates how a set of objects interact"

Mediator pattern is used to reduce communication complexity between multiple objects or classes.

Mediator Pattern

Benefits:

- It decouples the number of classes.
- It simplifies object protocols.
- It centralizes the control.
- The individual components become simpler and much easier to deal with because they don't need to pass messages to one another.
- The components don't need to contain logic to deal with their intercommunication and therefore, they are more generic.

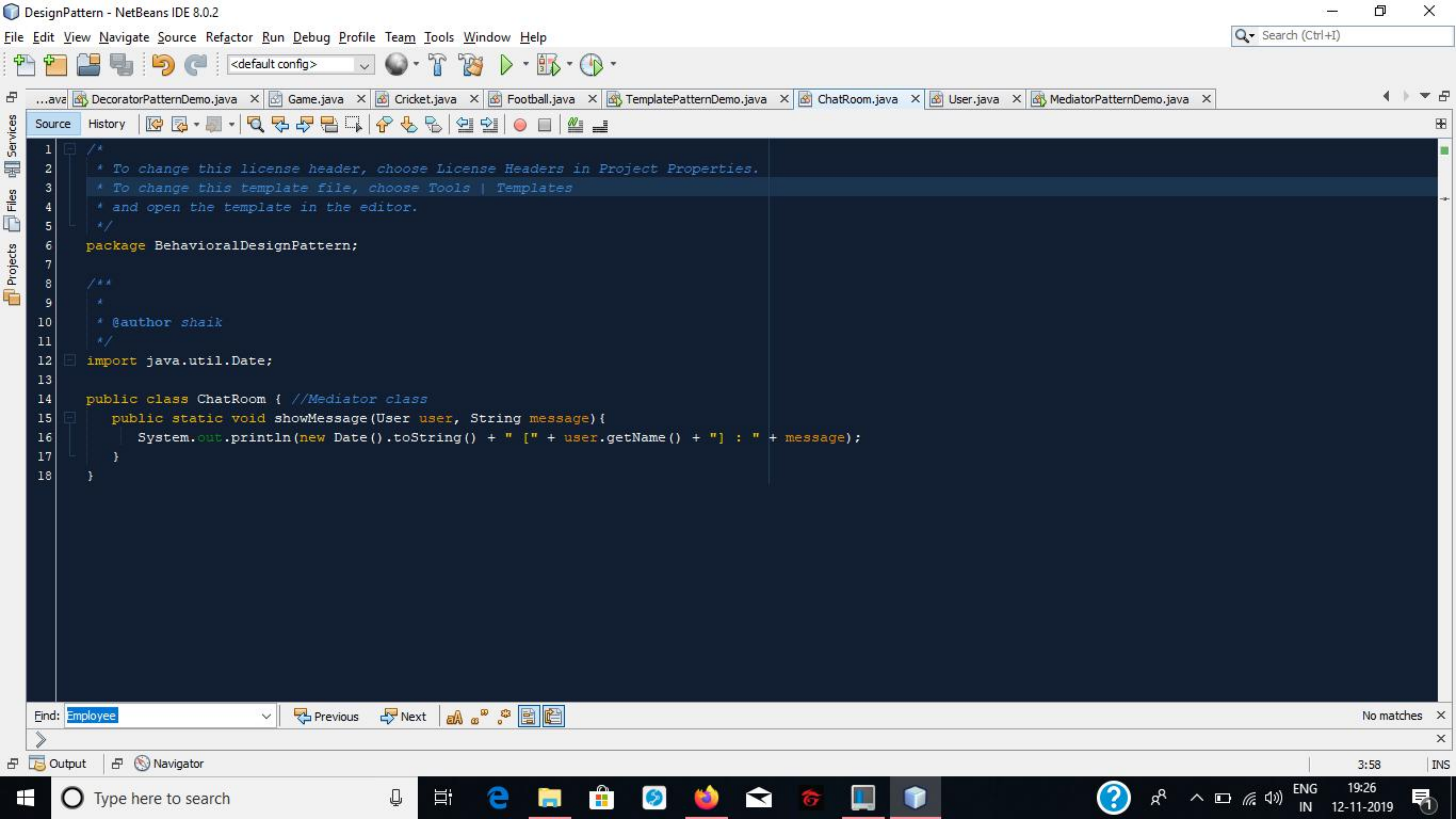
Mediator Pattern

Usage:

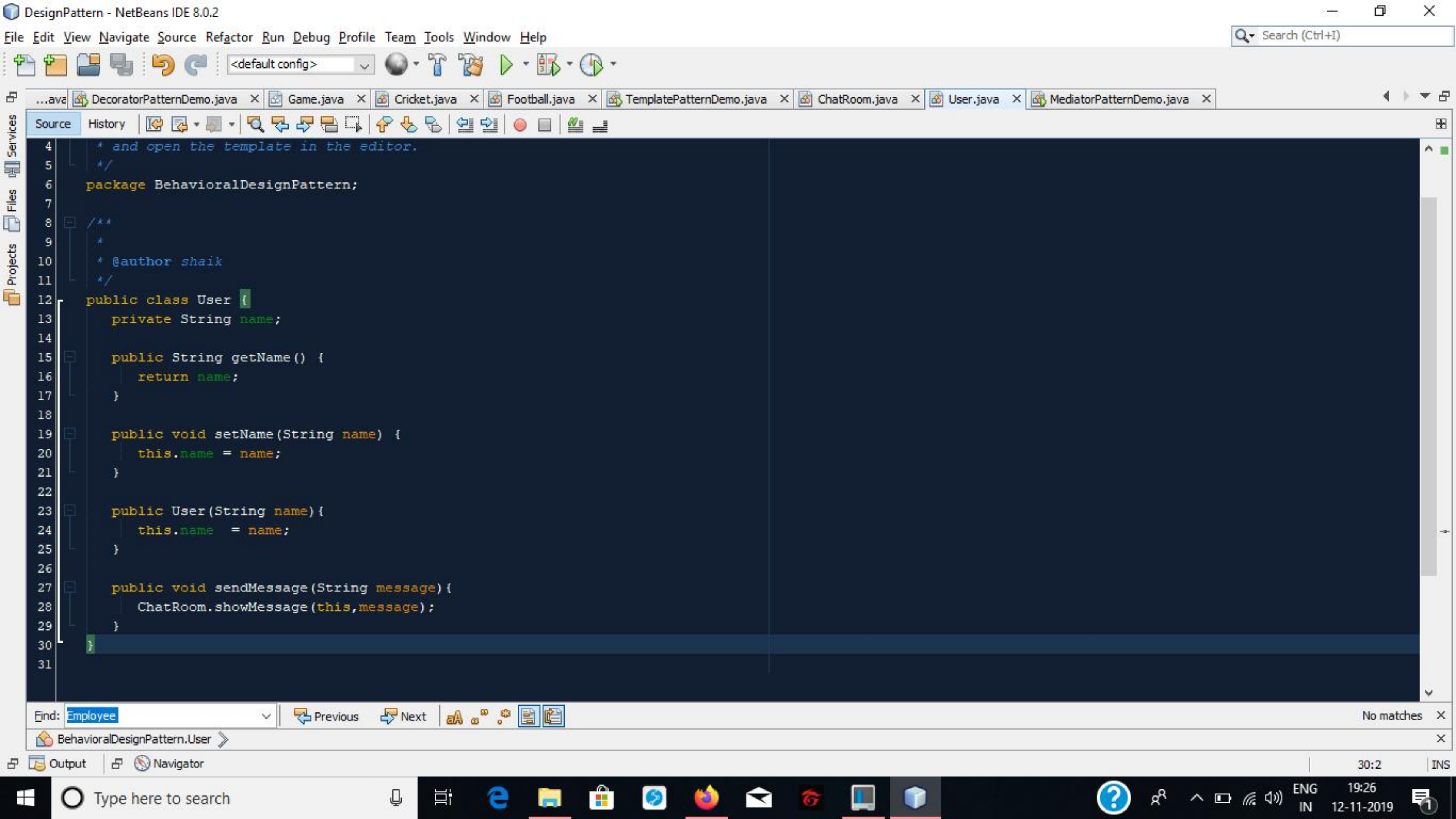
- It is commonly used in message-based systems likewise chat applications.
- When the set of objects communicate in complex but in well-defined ways.

Mediator Pattern

Implementation of Mediator Design Pattern

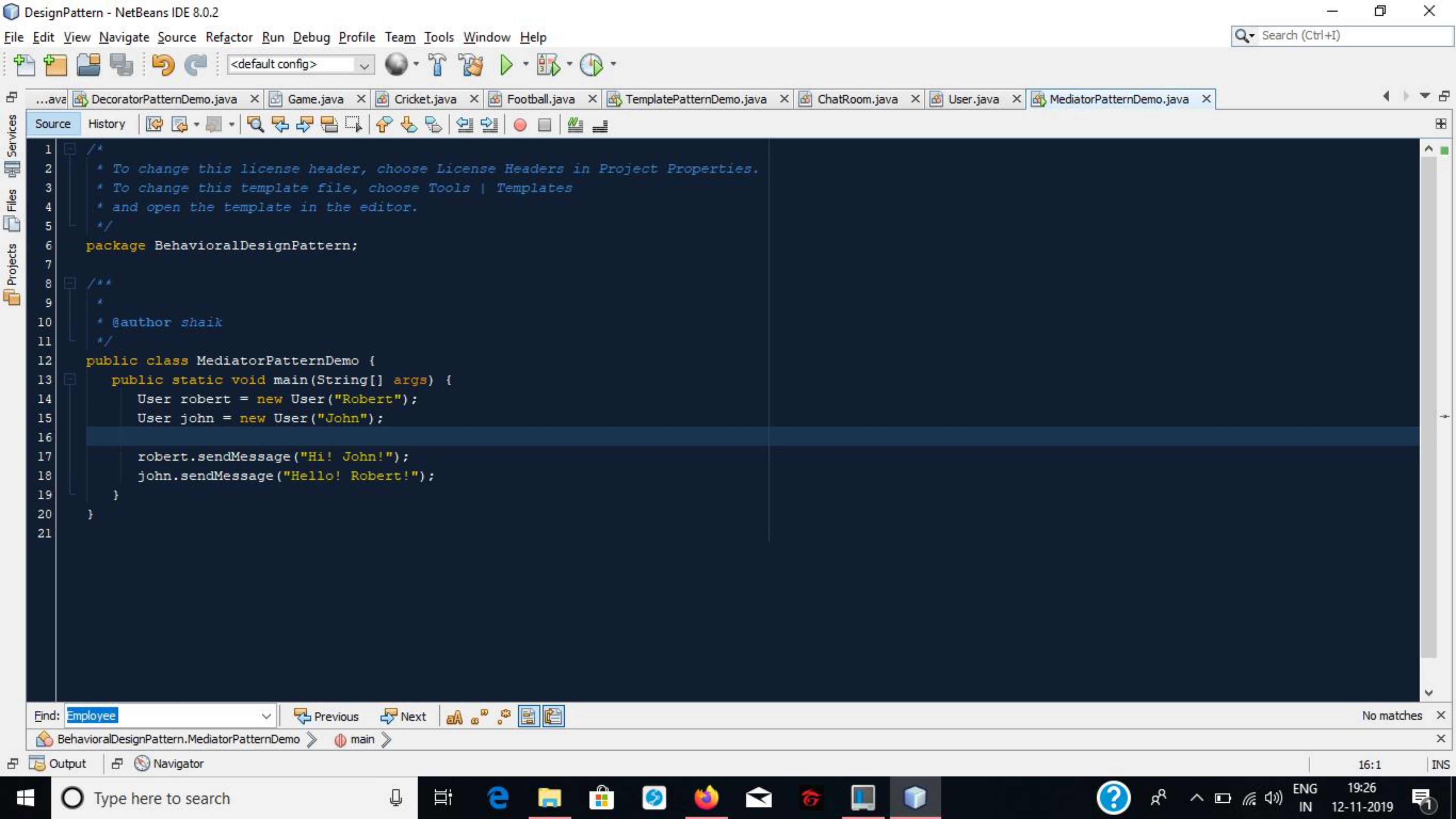

DesignPattern - NetBeans IDE 8.0.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
...ava DecoratorPatternDemo.java Game.java Cricket.java Football.java TemplatePatternDemo.java ChatRoom.java User.java MediatorPatternDemo.java
Source History
Find: Employee
Previous Next
No matches
Output Navigator
3:58 INS
Type here to search
ENG IN 19:26 12-11-2019

```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 BehavioralDesignPattern;

/**
 *
 * @author shaik
 */
import java.util.Date;

public class ChatRoom { //Mediator class
   public static void showMessage(User user, String message){
      System.out.println(new Date().toString() + " [" + user.getName() + "] : " + message);
   }
}
```



```java
 * and open the template in the editor.
 */
package BehavioralDesignPattern;

/**
 *
 * @author shaik
 */
public class User {
    private String name;

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public User(String name){
        this.name  = name;
    }

    public void sendMessage(String message){
        ChatRoom.showMessage(this,message);
    }
}
```

DesignPattern - NetBeans IDE 8.0.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
...ava DecoratorPatternDemo.java Game.java Cricket.java Football.java TemplatePatternDemo.java ChatRoom.java User.java MediatorPatternDemo.java
Source History
Find: Employee
Previous Next
No matches
BehavioralDesignPattern.User
Output Navigator
30:2 INS
Type here to search
ENG IN 19:26 12-11-2019



```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 BehavioralDesignPattern;

/**
 *
 * @author shaik
 */
public class MediatorPatternDemo {
    public static void main(String[] args) {
        User robert = new User("Robert");
        User john = new User("John");

        robert.sendMessage("Hi! John!");
        john.sendMessage("Hello! Robert!");
    }
}
```

DesignPattern - NetBeans IDE 8.0.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
...ava DecoratorPatternDemo.java Game.java Cricket.java Football.java TemplatePatternDemo.java ChatRoom.java User.java MediatorPatternDemo.java
Source History
Find: Employee
Previous Next
No matches
BehavioralDesignPattern.MediatorPatternDemo main
Output Navigator
16:1 INS
Type here to search
ENG IN 19:26 12-11-2019

Observer Pattern

"just define a one-to-one dependency so that when one object changes state, all its dependents are notified and updated automatically"

Observer Pattern

Benefits:

- It describes the coupling between the objects and the observer.
- It provides the support for broadcast-type communication.

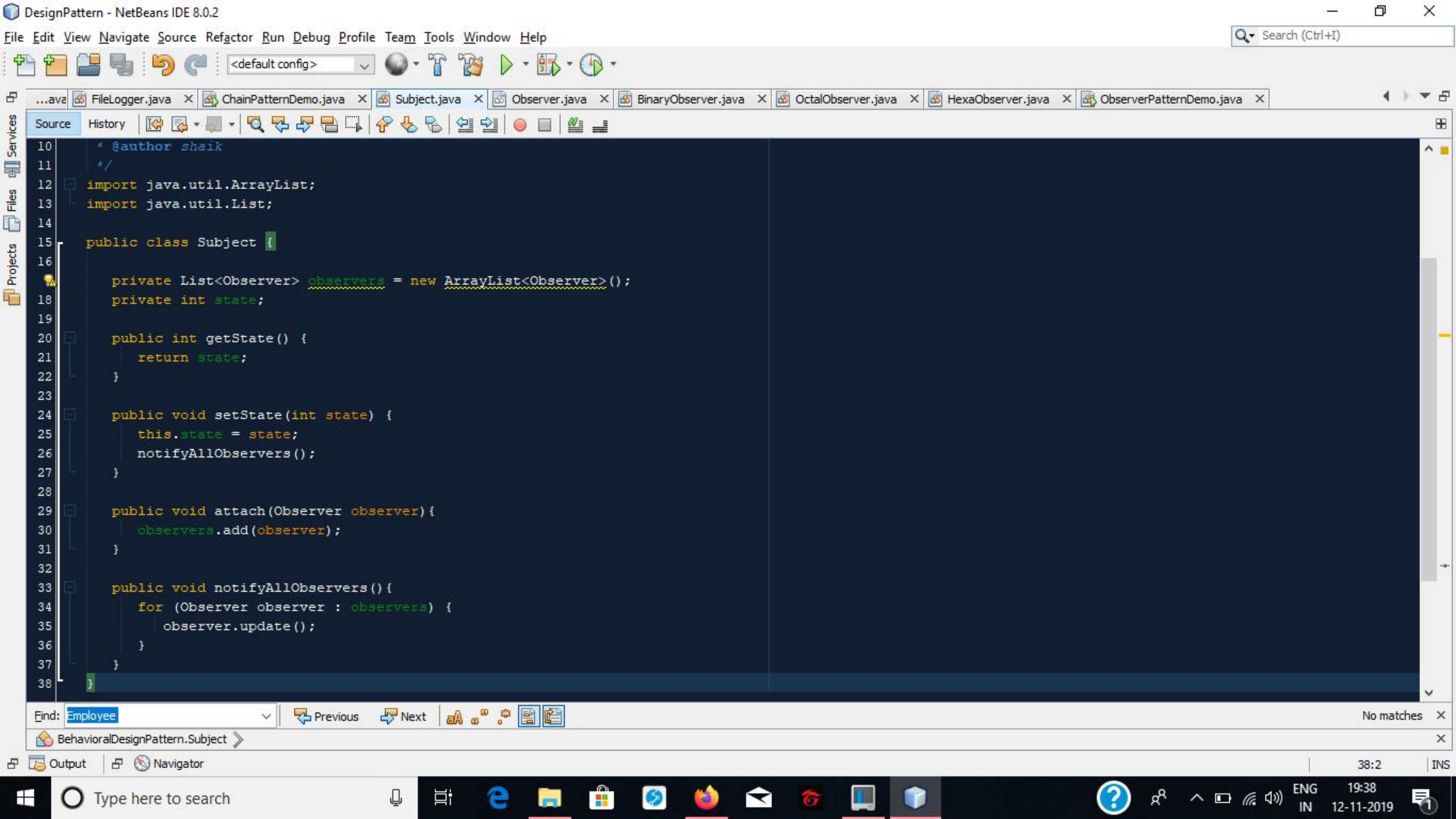
Observer Pattern

Usage:

- When the change of a state in one object must be reflected in another object without keeping the objects tight coupled.
- When the framework we writes and needs to be enhanced in future with new observers with minimal changes.

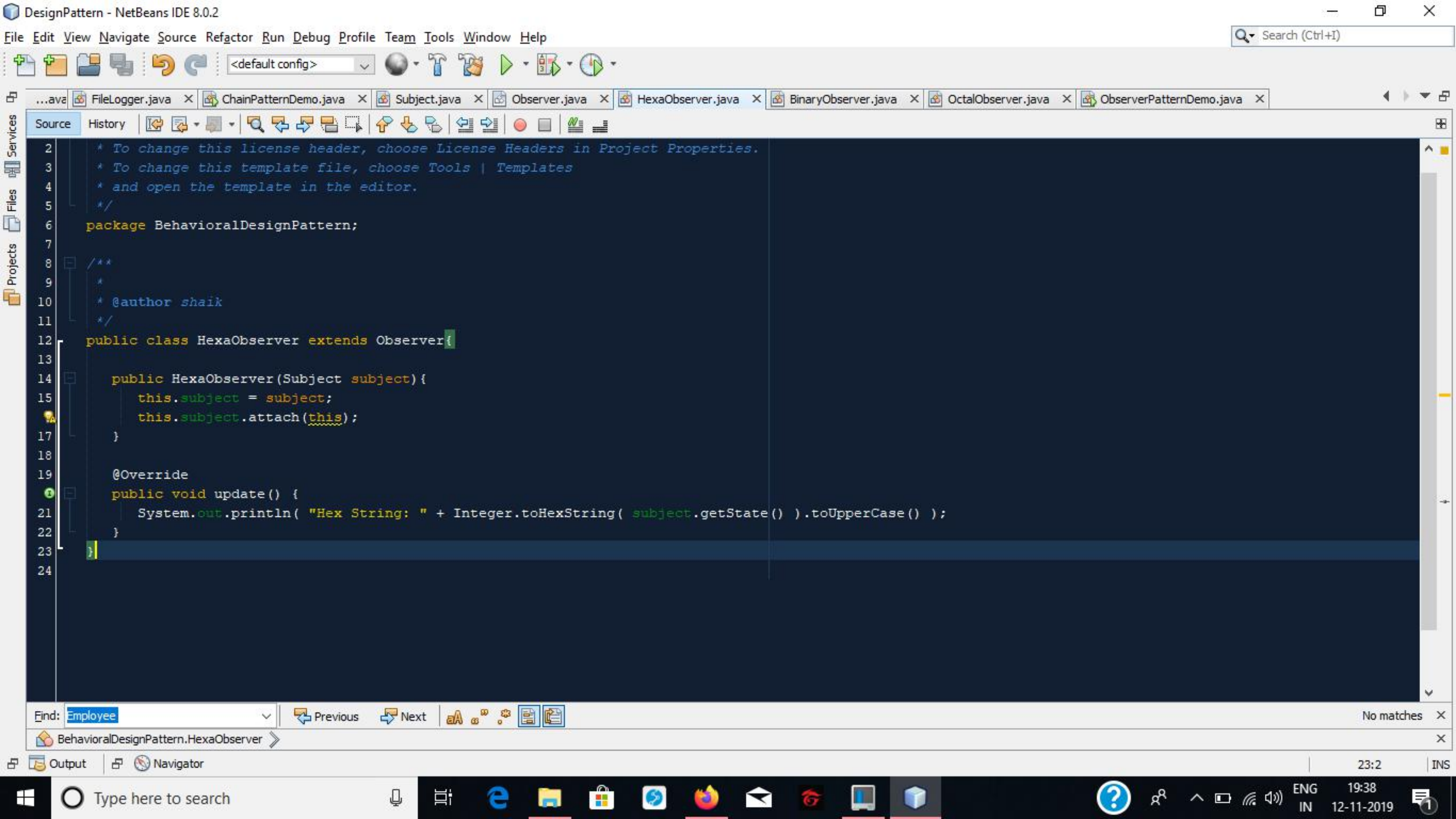
Observer Pattern

Implementation of Observer Design Pattern

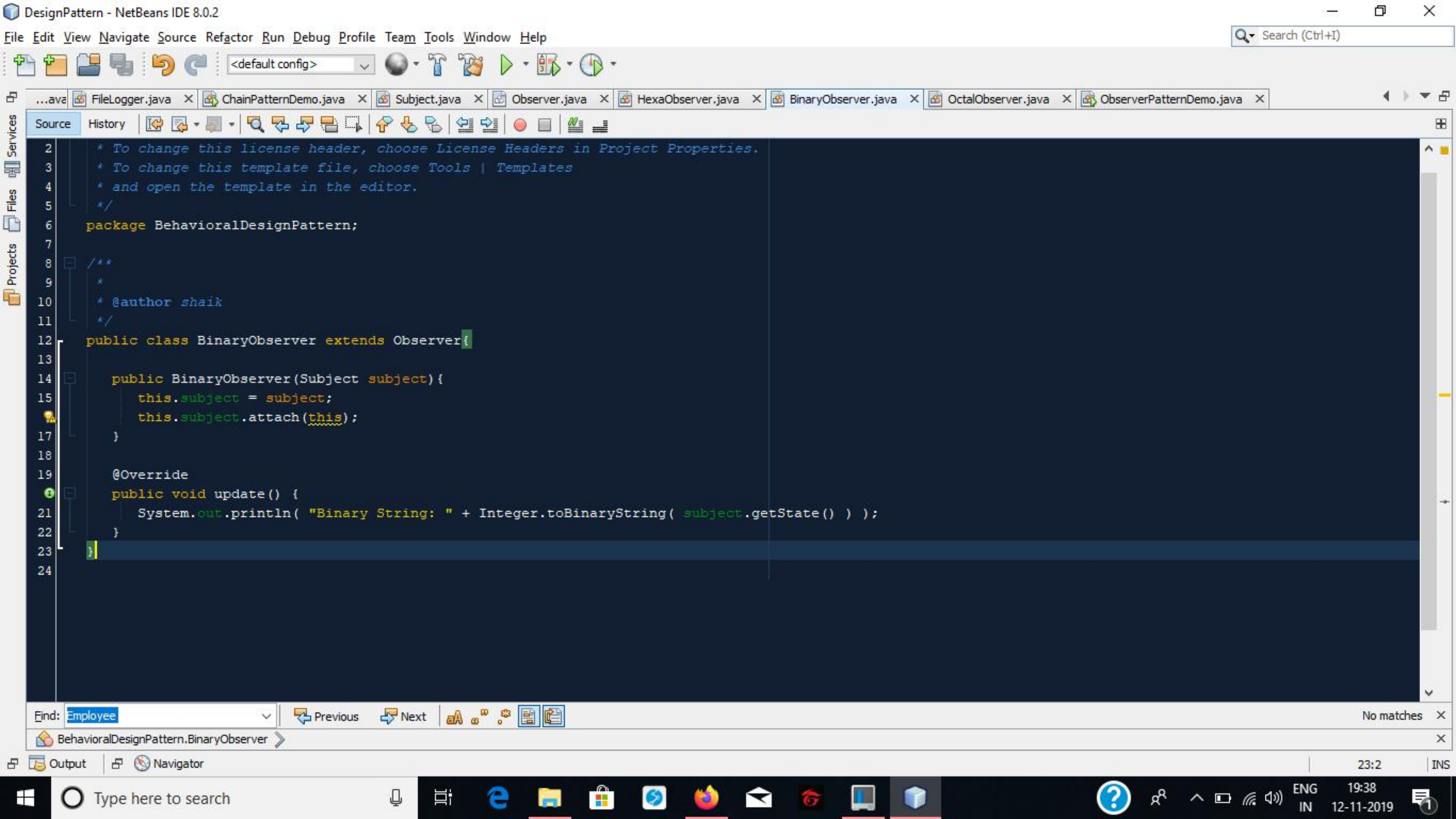


```
 * @author shaik
 */
import java.util.ArrayList;
import java.util.List;

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

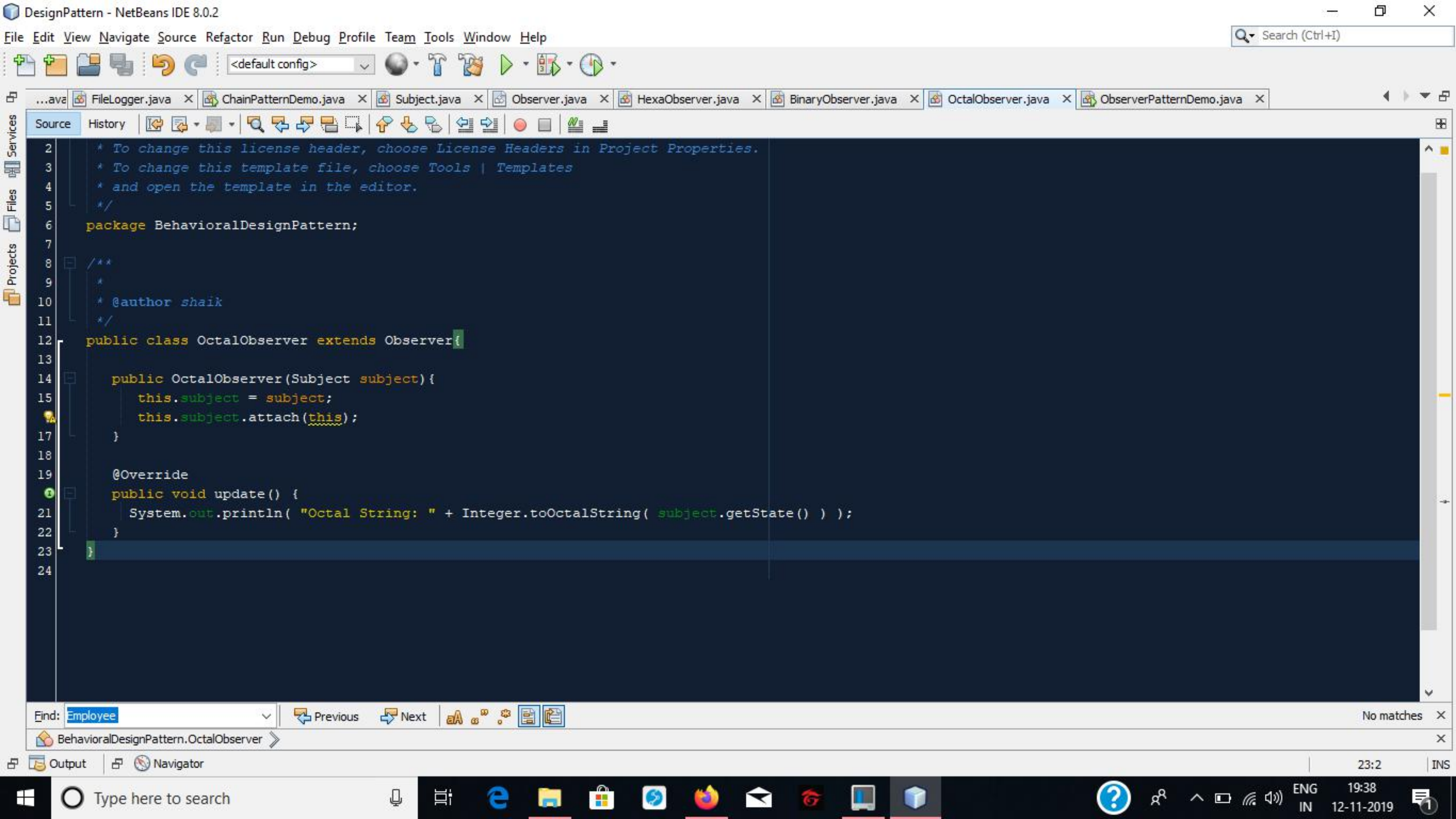
DesignPattern - NetBeans IDE 8.0.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
...ava FileLogger.java ChainPatternDemo.java Subject.java Observer.java BinaryObserver.java OctalObserver.java HexaObserver.java ObserverPatternDemo.java
Source History
Find: Employee
Previous
Next
No matches
BehavioralDesignPattern.Subject
Output
Navigator
38:2
INS
Type here to search
ENG IN
19:38
12-11-2019


DesignPattern - NetBeans IDE 8.0.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
FileLogger.java ChainPatternDemo.java Subject.java Observer.java HexaObserver.java BinaryObserver.java OctalObserver.java ObserverPatternDemo.java
Source History
Find: Employee
Previous Next
No matches
BehavioralDesignPattern.HexaObserver
Output Navigator
23:2 INS

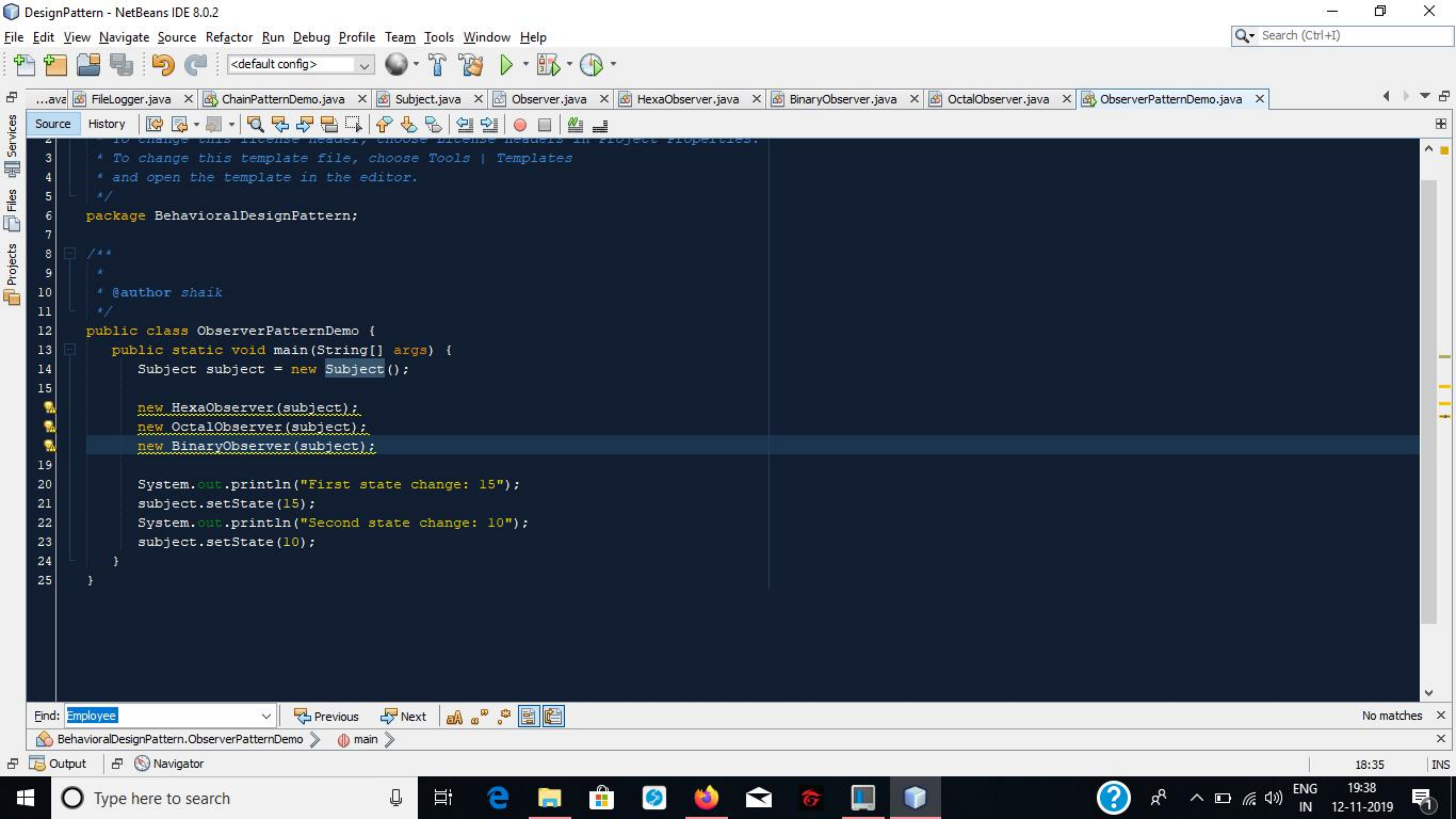
```java
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package BehavioralDesignPattern;

/**
 *
 * @author shaik
 */
public 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() );
    }
}
```


DesignPattern - NetBeans IDE 8.0.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
...ava FileLogger.java ChainPatternDemo.java Subject.java Observer.java HexaObserver.java BinaryObserver.java OctalObserver.java ObserverPatternDemo.java
Source History
Services Files Projects
Find: Employee Previous Next No matches
BehavioralDesignPattern.BinaryObserver
Output Navigator
23:2 INS
Type here to search
ENG IN 19:38 12-11-2019

```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 BehavioralDesignPattern;

/**
 *
 * @author shaik
 */
public 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() ) );
    }
}
```



```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 BehavioralDesignPattern;

/**
 *
 * @author shaik
 */
public 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() ) );
    }
}
```



DesignPattern - NetBeans IDE 8.0.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
...ava FileLogger.java ChainPatternDemo.java Subject.java Observer.java HexaObserver.java BinaryObserver.java OctalObserver.java ObserverPatternDemo.java
Source History
Services Files Projects
Find: Employee
Previous Next
No matches
BehavioralDesignPattern.ObserverPatternDemo
main
Output Navigator
18:35 INS
Type here to search
ENG IN 19:38 12-11-2019

```java
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package BehavioralDesignPattern;

/**
 *
 * @author shaik
 */
public class ObserverPatternDemo {
    public static void main(String[] args) {
        Subject subject = new Subject();

        new HexaObserver(subject);
        new OctalObserver(subject);
        new BinaryObserver(subject);

        System.out.println("First state change: 15");
        subject.setState(15);
        System.out.println("Second state change: 10");
        subject.setState(10);
    }
}
```

References

- https://www.javatpoint.com/template-pattern
- https://www.tutorialspoint.com/design_pattern/template_pattern.htm
- https://www.javatpoint.com/mediator-pattern
- https://www.tutorialspoint.com/design_pattern/mediator_pattern.htm
- https://www.javatpoint.com/chain-of-responsibility-pattern
- https://www.tutorialspoint.com/design_pattern/chain_of_responsibility_pattern.htm
- https://www.javatpoint.com/observer-pattern
- https://www.tutorialspoint.com/design_pattern/observer_pattern.htm