****

**Name: Kashan Javed**

**Reg: FA20-BSE-040**

**Lab: MID**

**Subject Design Pattern**

**Code:**

package mid.lab;

/\*\*

\*

\* @author Administrator

\*/

import java.util.ArrayList;

import java.util.List;

interface MatchSubject {

void registerObserver(MatchObserver observer);

void removeObserver(MatchObserver observer);

void notifyObservers();

}

interface MatchObserver {

void update(String score);

}

class CricketMatch implements MatchSubject {

private List<MatchObserver> observers = new ArrayList<>();

private String score;

private String matchInfo;

public void registerObserver(MatchObserver observer) {

observers.add(observer);

}

public void removeObserver(MatchObserver observer) {

observers.remove(observer);

}

public void notifyObservers() {

for (MatchObserver observer : observers) {

observer.update(score);

}

}

public void startMatch() {

matchInfo = "Match Started";

notifyObservers();

}

public void endMatch() {

matchInfo = "Match Ended";

notifyObservers();

}

public void updateScore(String score) {

this.score = score;

notifyObservers();

}

public String getMatchInfo() {

return matchInfo;

}

}

class LiveMatchList implements MatchObserver {

private CricketMatch currentMatch;

public void update(String score) {

System.out.println("Live Match List: " + currentMatch.getMatchInfo() + " - " + score);

}

public void displayMatchList(List<CricketMatch> matches) {

System.out.println("Live Matches:");

for (CricketMatch match : matches) {

System.out.println("- " + match.getMatchInfo());

}

}

public void selectMatch(CricketMatch match) {

match.registerObserver(this);

currentMatch = match;

}

}

class BallByBallCoverage implements MatchObserver {

public void update(String score) {

System.out.println("Ball-by-Ball Coverage: " + score);

}

public void displayCoverage(CricketMatch match) {

match.registerObserver(this);

}

}

public class MIDLAB {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

List<CricketMatch> liveMatches = new ArrayList<>();

CricketMatch match1 = new CricketMatch();

CricketMatch match2 = new CricketMatch();

liveMatches.add(match1);

liveMatches.add(match2);

LiveMatchList liveMatchList = new LiveMatchList();

BallByBallCoverage ballByBallCoverage = new BallByBallCoverage();

liveMatchList.displayMatchList(liveMatches);

liveMatchList.selectMatch(match1);

match1.startMatch();

match1.updateScore("Team A: 100/1");

match1.updateScore("Team A: 150/2");

match1.endMatch();

}

}