Name **Muhammad Nawaz Khan**

Reg No **FA20-BSE-073**

Lab Mid Exam

Match.java

```
public class Match extends MatchSubject {

    private int score = 0;
    private int bowls = 0;
    String match;

    public Match(String match) {
        this.match = match;
    }

    public int getScore() {
        return score;
    }

    public void setScore(int score) {

        synchronized (this) {
            this.score = score;
            notifyObservers();
        }
    }

    public int getBowls() {
        return bowls;
    }

    public void setBowls(int bowls) {

        synchronized (this) {
            this.bowls = bowls;
            notifyObservers();
        }
    }
```

```
    public String getName() {
        return this.match;
    }

}
```

MatchSubject.java

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

public abstract class MatchSubject {
    List<MatchObserver> matchObservers = new ArrayList<>();

    public void notifyObservers() {
        for (MatchObserver observer : matchObservers) {
            observer.update();
        }
    }

    public void attach(MatchObserver observer) {
        matchObservers.add(observer);
    }

    public void detach(MatchObserver observer) {
        matchObservers.remove(observer);
    }
}
```

matchObserver.java

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

public abstract class MatchSubject {
    List<MatchObserver> matchObservers = new ArrayList<>();

    public void notifyObservers() {
        for (MatchObserver observer : matchObservers) {
```

```
            observer.update();
        }
    }

    public void attach(MatchObserver observer) {
        matchObservers.add(observer);
    }

    public void detach(MatchObserver observer) {
        matchObservers.remove(observer);
    }
}
```

Viewer1.java

```
public class Viewer1 extends MatchObserver {

    public Viewer1(Match matchSubject) {

        this.matchSubject = matchSubject;
        this.matchSubject.attach(this);

    }

    @Override
    public void update() {

        System.out
                .println("match Score: " + matchSubject.getScore() + "\n" + "
match bowls: " + matchSubject.getBowls());

    }

}
```

Viewer2.java

```
public class Viewer2 extends MatchObserver {

    public Viewer2(Match matchSubject) {

        this.matchSubject = matchSubject;
```

```
        this.matchSubject.attach(this);

    }

    @Override
    public void update() {

        System.out
                .println("match Score: " + matchSubject.getScore() + "\n" + "
match bowls: " + matchSubject.getBowls());

    }
}
```

MatchDemo.java

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

public class MatchDemo {

    public static void main(String[] args) {
        List<Match> matches = createSampleMatches();

        Scanner scanner = new Scanner(System.in);

        do {
            // Display the list of available matches
            System.out.println("Available Matches:");
            for (int i = 0; i < matches.size(); i++) {
                System.out.println(i + ". " + matches.get(i).getName());
            }

            // Ask the user to choose a match
            System.out.print("Enter the match number you want to subscribe to (or
-1 to exit): ");
            int matchNumber = scanner.nextInt();

            if (matchNumber >= 0 && matchNumber < matches.size()) {
                Match selectedMatch = matches.get(matchNumber);

                List<Viewer1> viewers = new ArrayList<>();
```

```
                // Ask the user how many viewers to attach
                System.out.print("Enter the number of viewers you want to attach:
");
                int numViewers = scanner.nextInt();

                for (int i = 0; i < numViewers; i++) {
                    Viewer1 viewer = new Viewer1(selectedMatch);
                    viewers.add(viewer);
                }

                System.out.println("You are now subscribed to updates for " +
selectedMatch.getName());

                // Simulate updates (you can implement your own logic for
updates)
                System.out.println("Simulating real-time updates...");
                for (int i = 1; i <= 5; i++) {
                    selectedMatch.setScore(i * 10);
                    selectedMatch.setBowls(i * 6);
                    try {
                        Thread.sleep(2000);
                    } catch (InterruptedException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                    }
                }
            } else if (matchNumber == -1) {
                System.out.println("Exiting the program.");
                break;
            } else {
                System.out.println("Invalid match selection.");
            }
        } while (true);
    }

    private static List<Match> createSampleMatches() {
        List<Match> matches = new ArrayList<>();
        matches.add(new Match("Match 1"));
        matches.add(new Match("Match 2"));
        matches.add(new Match("Match 3"));
        return matches;
    }
}
```