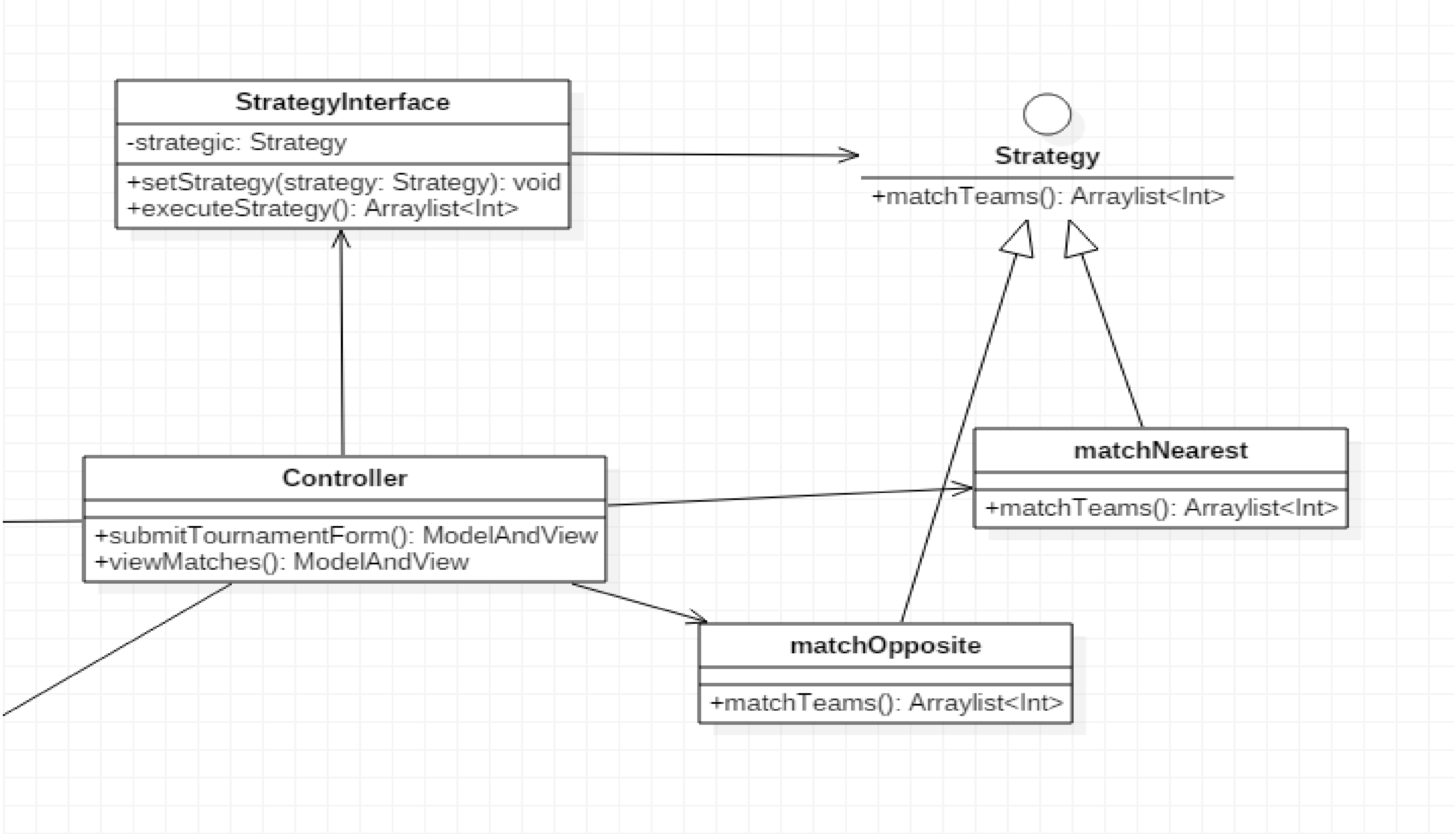
08 GameDashBoard

- Team Members: Yang Yang
- Zhaozhong Peng
- Yuhou Wang
-
- Vision: Create a place where you can discuss games and Compete against one another in tournaments

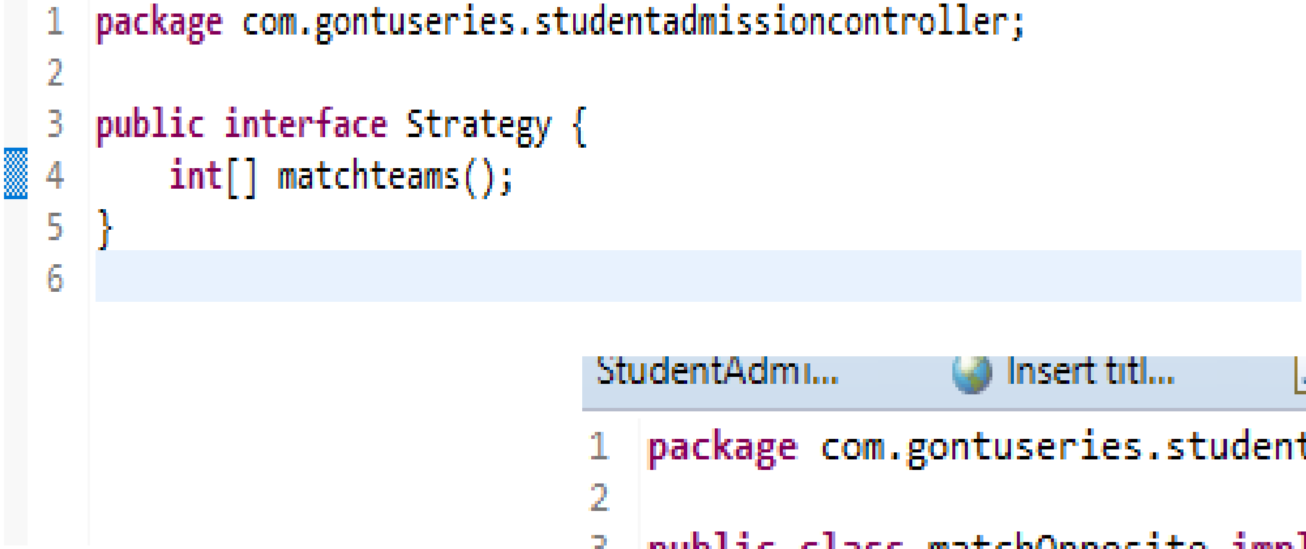
Project Description

- Originally, we wanted to implement both a game dashboard and a tournament hosting site
-
- We only implemented tournament hosting where users can create tournaments, sign up for tournaments, and compete against one another after being matched by our system.

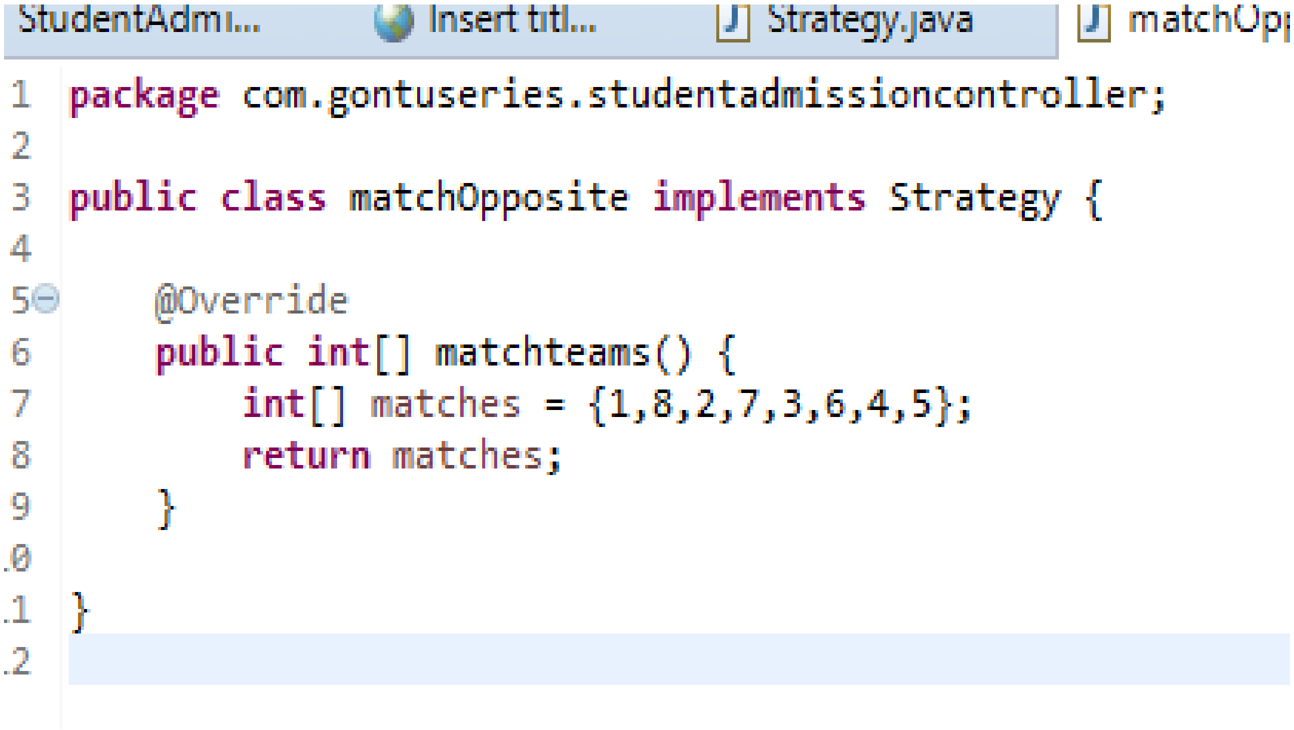
08_GameDashBoard Strategy


StrategyInterface
-strategic: Strategy
+setStrategy(strategy: Strategy): void
+executeStrategy(): Arraylist<Int>
Strategy
+matchTeams(): Arraylist<Int>
Controller
+submitTournamentForm(): ModelAndView
+viewMatches(): ModelAndView
matchNearest
+matchTeams(): Arraylist<Int>
matchOpposite
+matchTeams(): Arraylist<Int>

Strategy

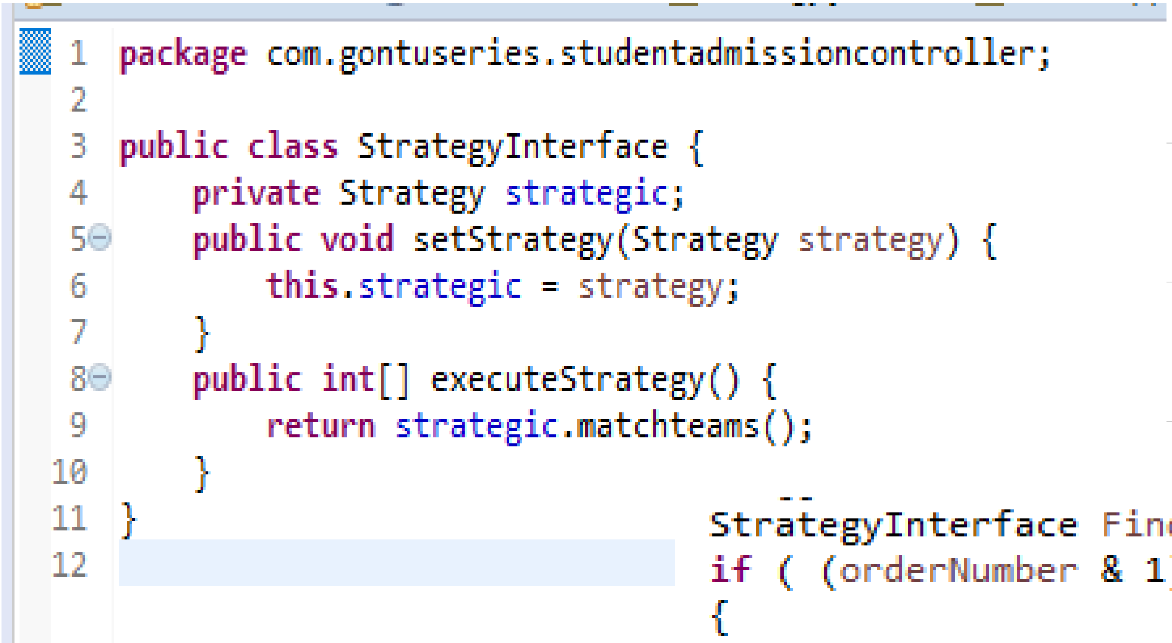


```
package com.gontuseries.studentadmissioncontroller;

public interface Strategy {
    int[] matchteams();
}

```

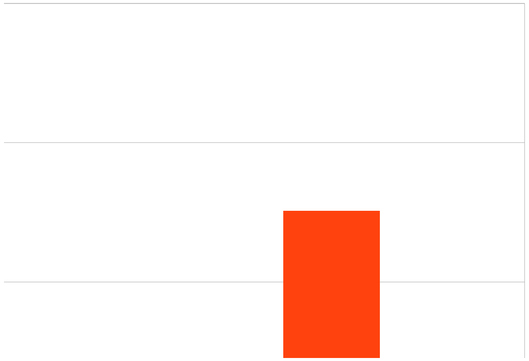


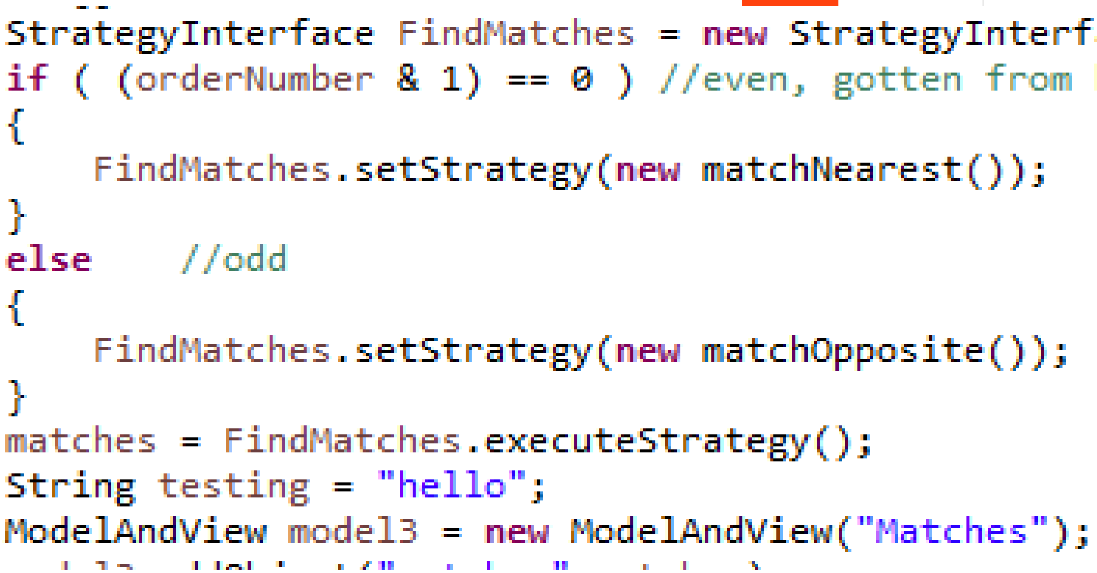
```
package com.gontuseries.studentadmissioncontroller;

public class matchOpposite implements Strategy {

    @Override
    public int[] matchteams() {
        int[] matches = {1,8,2,7,3,6,4,5};
        return matches;
    }

}

```

Strategy Interface/Context

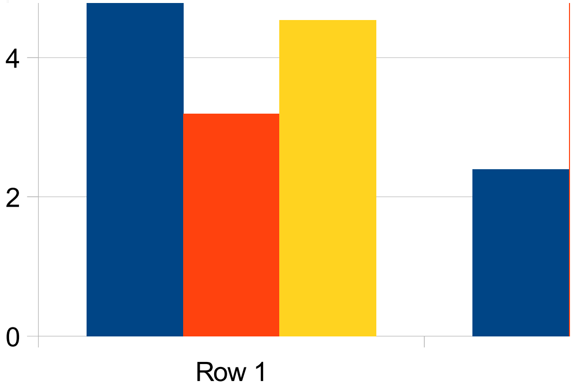


```
package com.gontuseries.studentadmissioncontroller;

public class StrategyInterface {
    private Strategy strategic;
    public void setStrategy(Strategy strategy) {
        this.strategic = strategy;
    }
    public int[] executeStrategy() {
        return strategic.matchteams();
    }
}
```

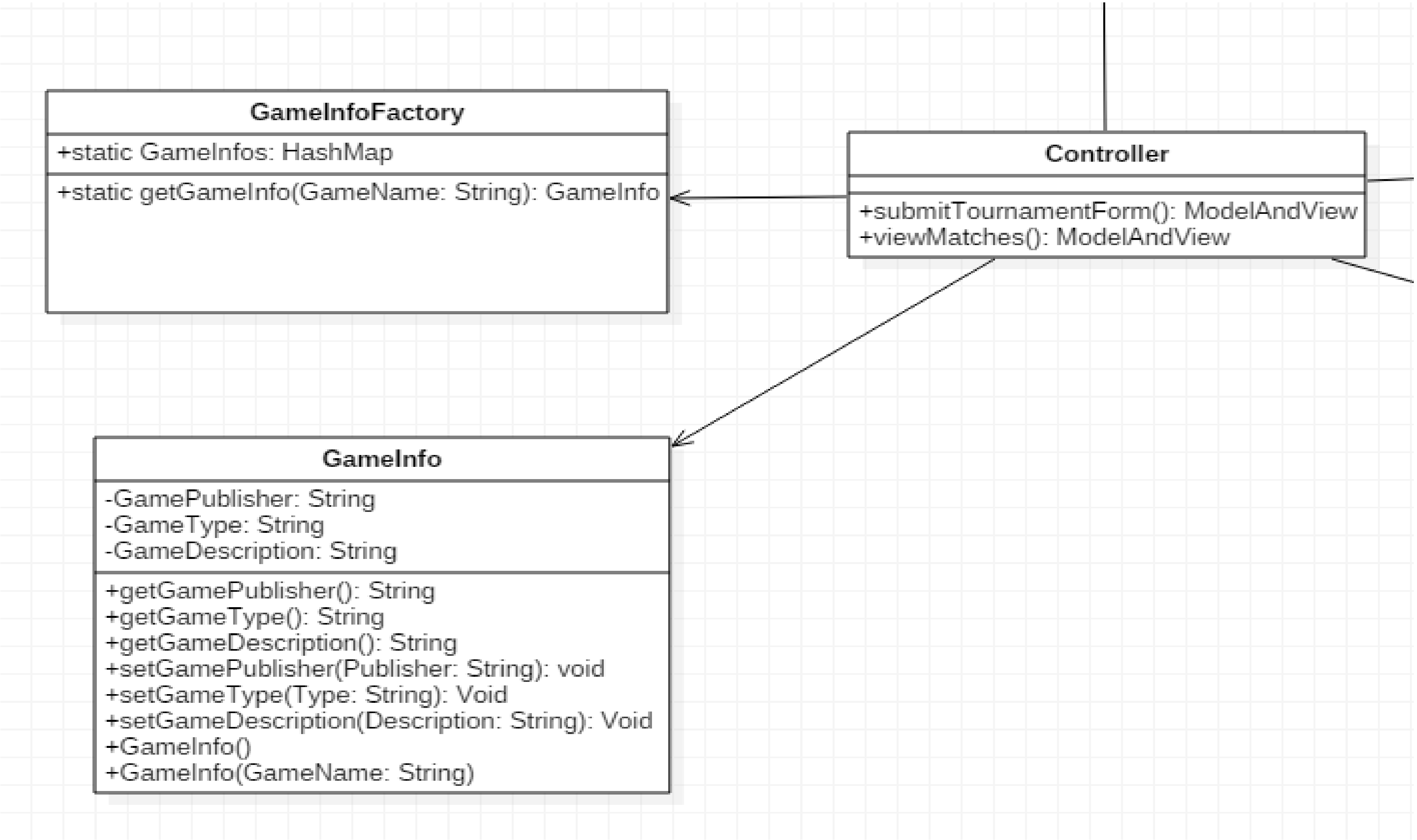




```
StrategyInterface FindMatches = new StrategyInterfac
if ( (orderNumber & 1) == 0 ) //even, gotten from ht
{
    FindMatches.setStrategy(new matchNearest());
}
else    //odd
{
    FindMatches.setStrategy(new matchOpposite());
}
matches = FindMatches.executeStrategy();
String testing = "hello";
ModelAndView model3 = new ModelAndView("Matches");
```



Flyweight



GameInfoFactory

+static GameInfos: HashMap

+static getGameInfo(GameName: String): GameInfo

Controller

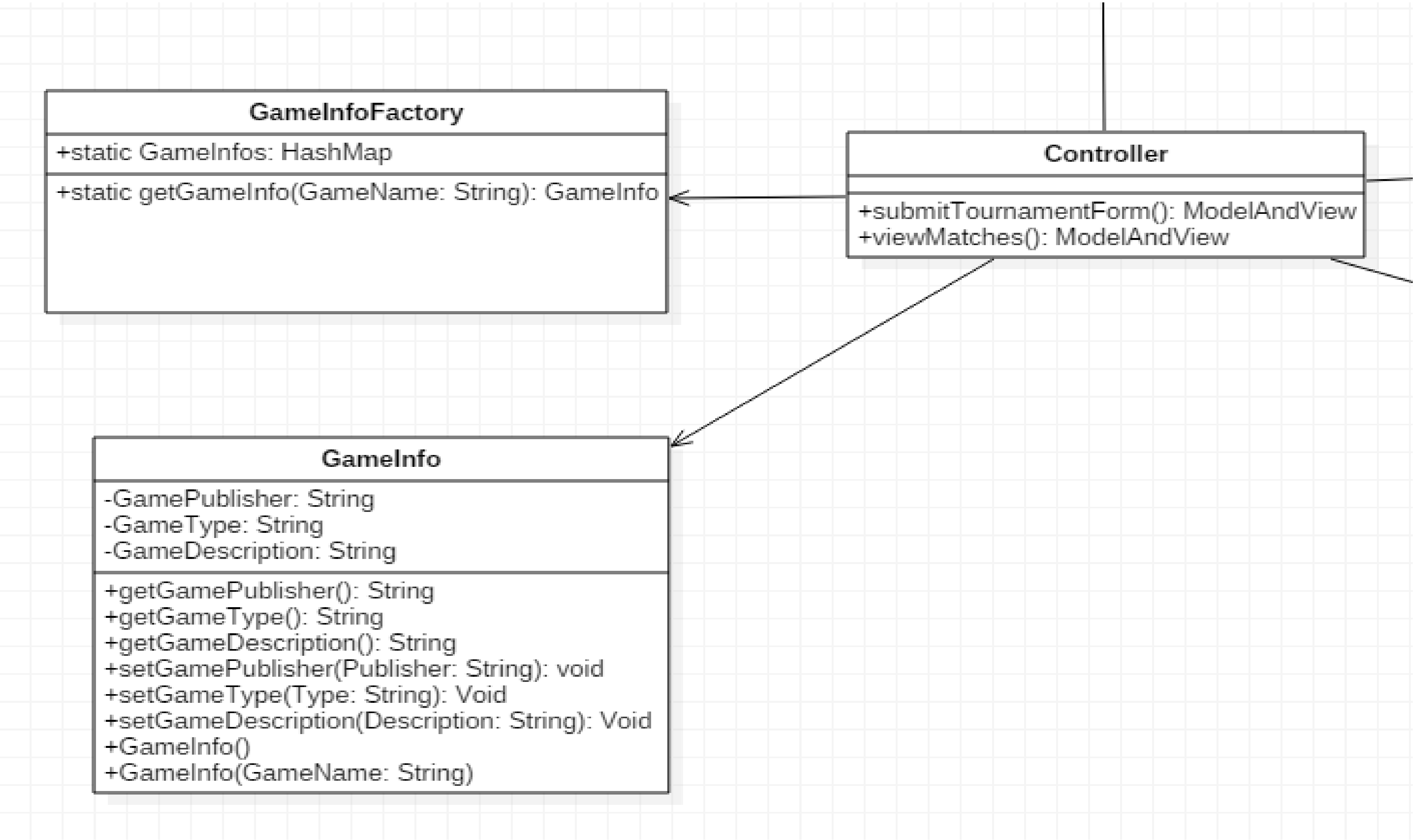
+submitTournamentForm(): ModelAndView
+viewMatches(): ModelAndView

GameInfo

-GamePublisher: String
-GameType: String
-GameDescription: String

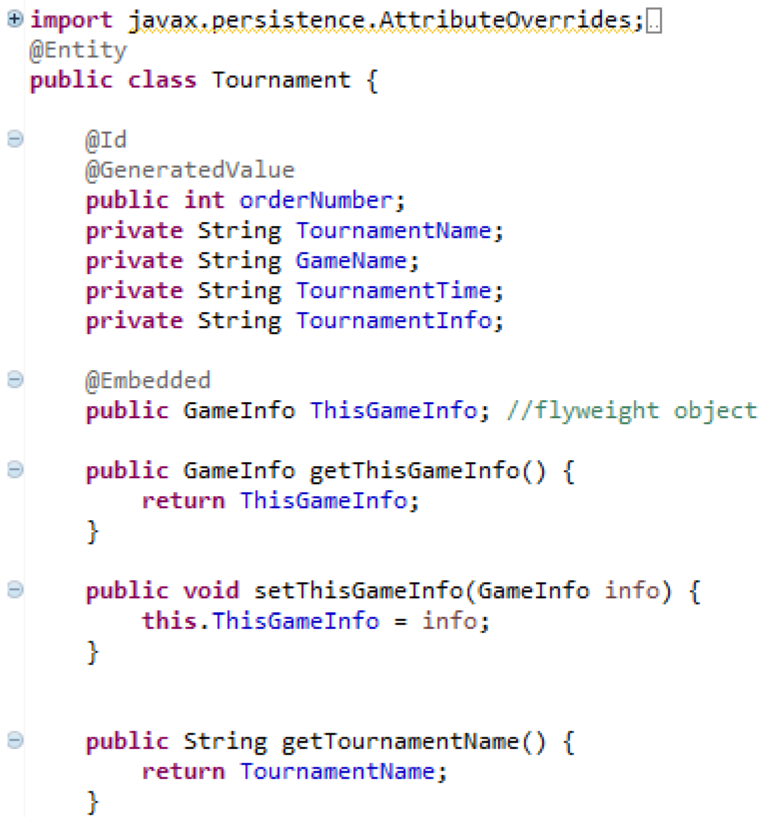
+getGamePublisher(): String
+getGameType(): String
+getGameDescription(): String
+setGamePublisher(Publisher: String): void
+setGameType(Type: String): Void
+setGameDescription(Description: String): Void
+GameInfo()
+GameInfo(GameName: String)

Flyweight

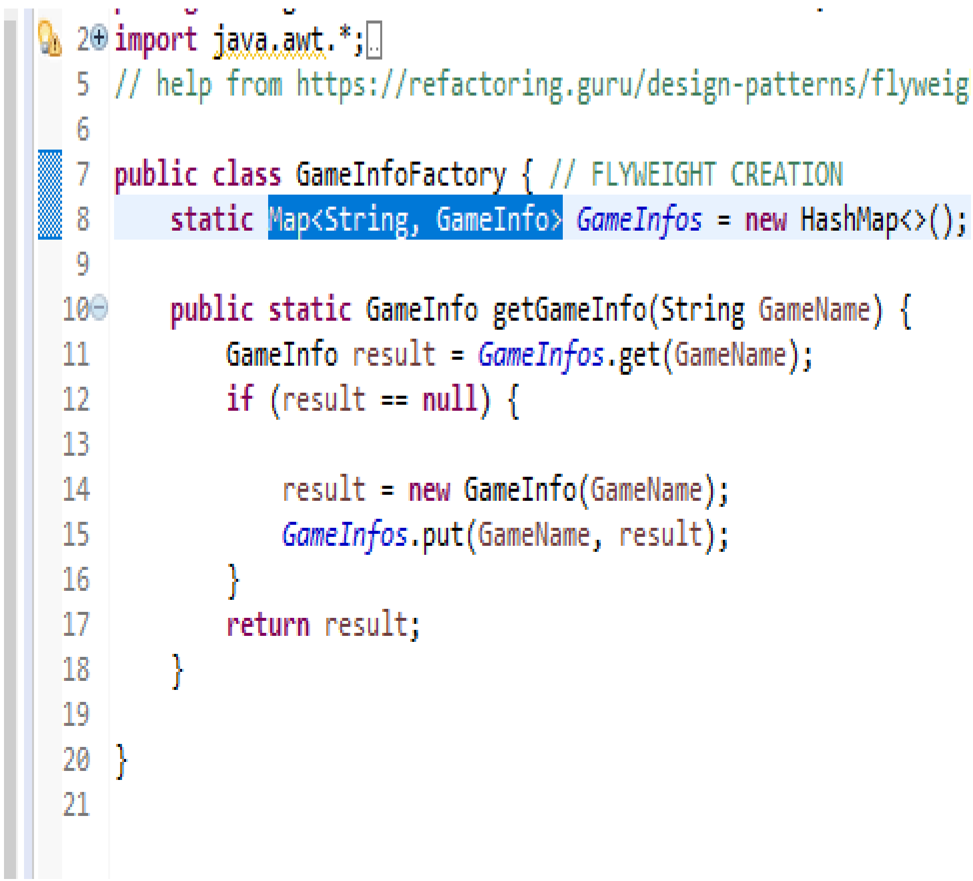


GameInfoFactory
+static GameInfos: HashMap
+static getGameInfo(GameName: String): GameInfo
Controller
+submitTournamentForm(): ModelAndView
+viewMatches(): ModelAndView
GameInfo
-GamePublisher: String
-GameType: String
-GameDescription: String
+getGamePublisher(): String
+getGameType(): String
+getGameDescription(): String
+setGamePublisher(Publisher: String): void
+setGameType(Type: String): Void
+setGameDescription(Description: String): Void
+GameInfo()
+GameInfo(GameName: String)

GameInfo is part of Tournament

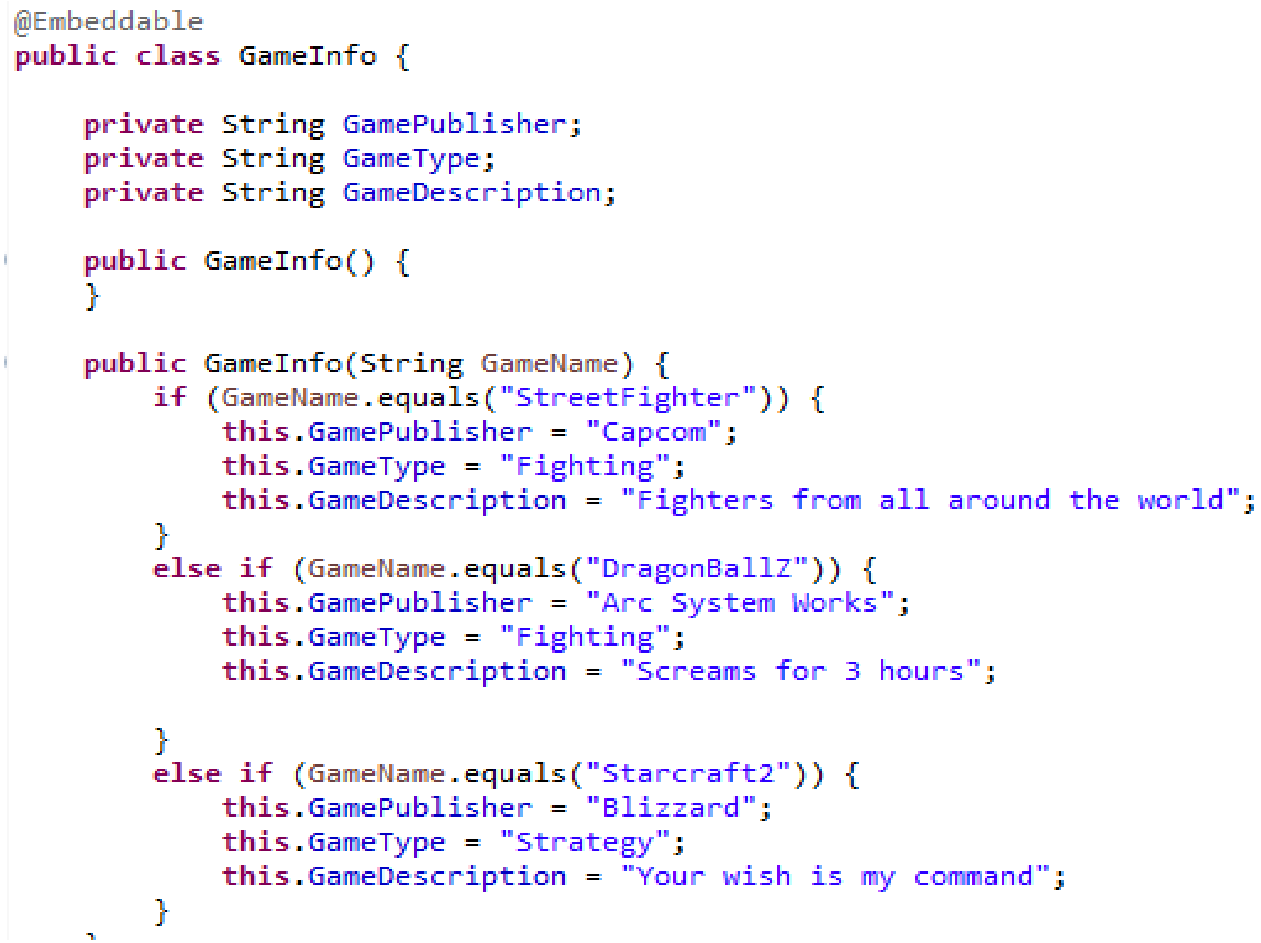


```java
import javax.persistence.AttributeOverrides;
@Entity
public class Tournament {

    @Id
    @GeneratedValue
    public int orderNumber;
    private String TournamentName;
    private String GameName;
    private String TournamentTime;
    private String TournamentInfo;

    @Embedded
    public GameInfo ThisGameInfo; //flyweight object

    public GameInfo getThisGameInfo() {
        return ThisGameInfo;
    }

    public void setThisGameInfo(GameInfo info) {
        this.ThisGameInfo = info;
    }


    public String getTournamentName() {
        return TournamentName;
    }
```



```java
import java.awt.*;
// help from https://refactoring.guru/design-patterns/flyweig

public class GameInfoFactory { // FLYWEIGHT CREATION
    static Map<String, GameInfo> GameInfos = new HashMap<>();

    public static GameInfo getGameInfo(String GameName) {
        GameInfo result = GameInfos.get(GameName);
        if (result == null) {

            result = new GameInfo(GameName);
            GameInfos.put(GameName, result);
        }
        return result;
    }

}
```

GameInfo



```
@Embeddable
public class GameInfo {

    private String GamePublisher;
    private String GameType;
    private String GameDescription;

    public GameInfo() {
    }

    public GameInfo(String GameName) {
        if (GameName.equals("StreetFighter")) {
            this.GamePublisher = "Capcom";
            this.GameType = "Fighting";
            this.GameDescription = "Fighters from all around the world";
        }
        else if (GameName.equals("DragonBallZ")) {
            this.GamePublisher = "Arc System Works";
            this.GameType = "Fighting";
            this.GameDescription = "Screams for 3 hours";

        }
        else if (GameName.equals("Starcraft2")) {
            this.GamePublisher = "Blizzard";
            this.GameType = "Strategy";
            this.GameDescription = "Your wish is my command";
        }
```

- Video download link:
- https://github.com/niru32868/CSCI4448Project/blob
-
- Video View Link(if download no work):
- https://www.screencast.com/t/TQSBYwZkS