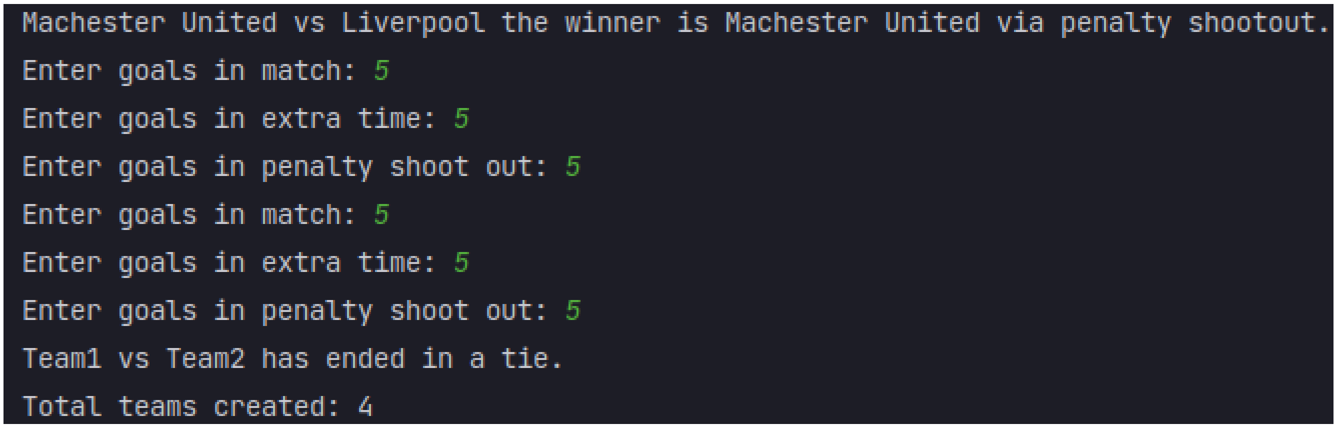
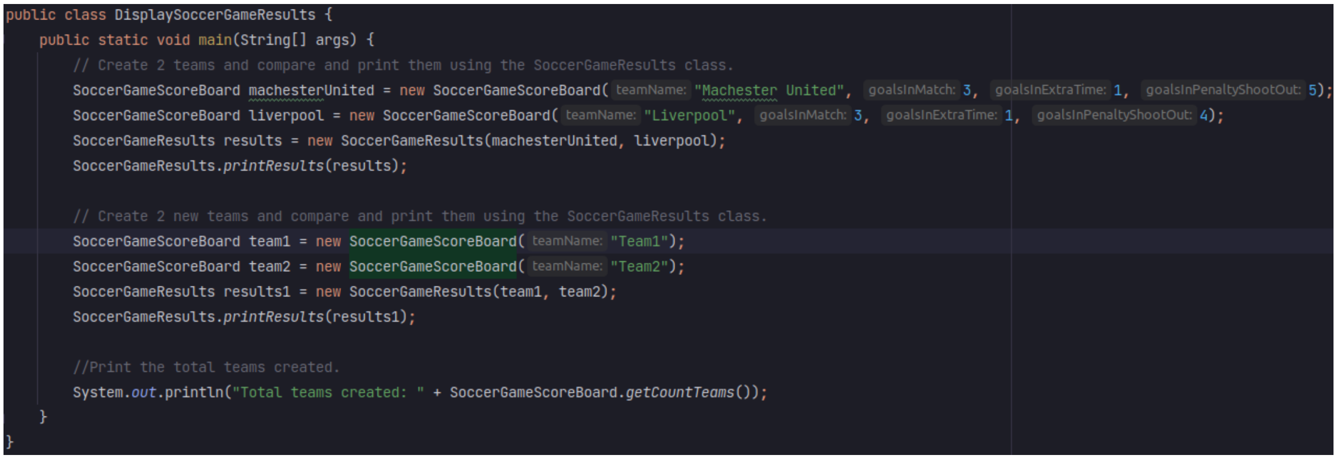
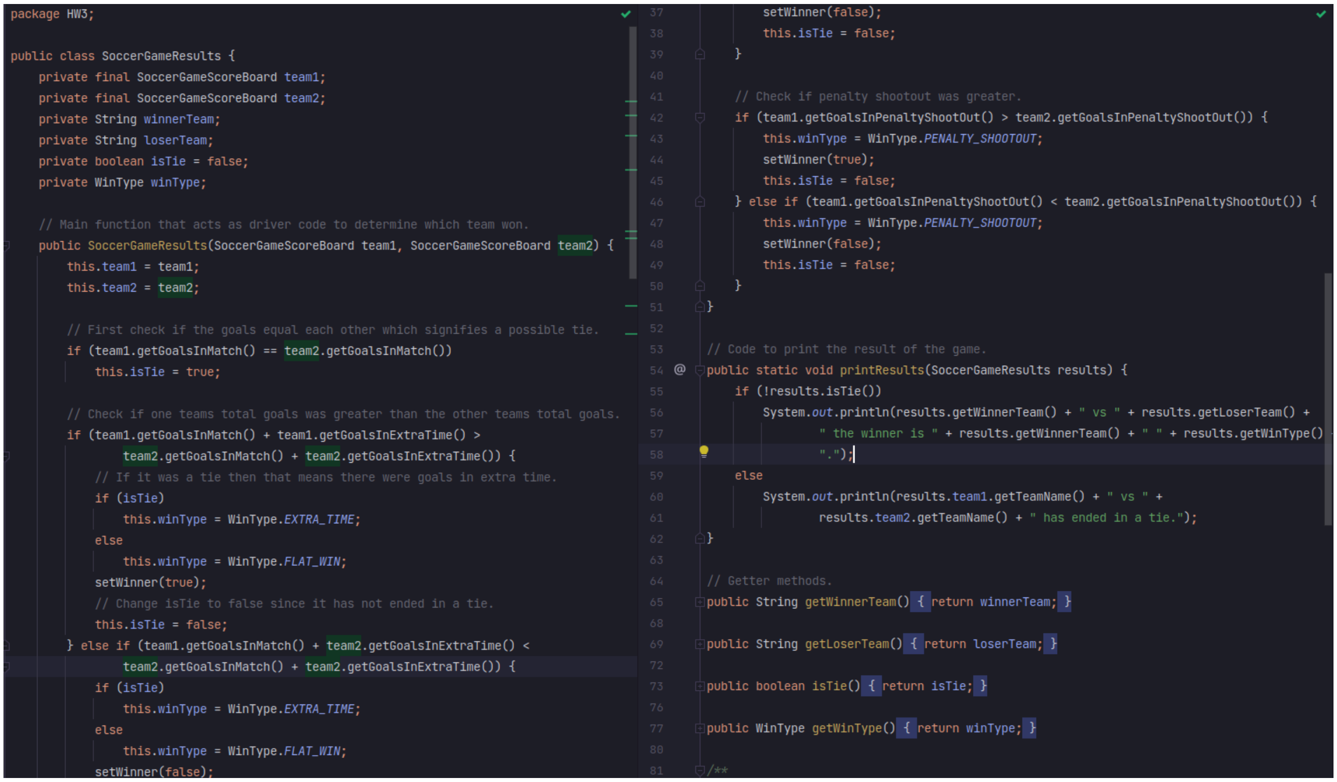


```
Machester United vs Liverpool the winner is Machester United via penalty shootout.
Enter goals in match: 5
Enter goals in extra time: 5
Enter goals in penalty shoot out: 5
Enter goals in match: 5
Enter goals in extra time: 5
Enter goals in penalty shoot out: 5
Team1 vs Team2 has ended in a tie.
Total teams created: 4
```

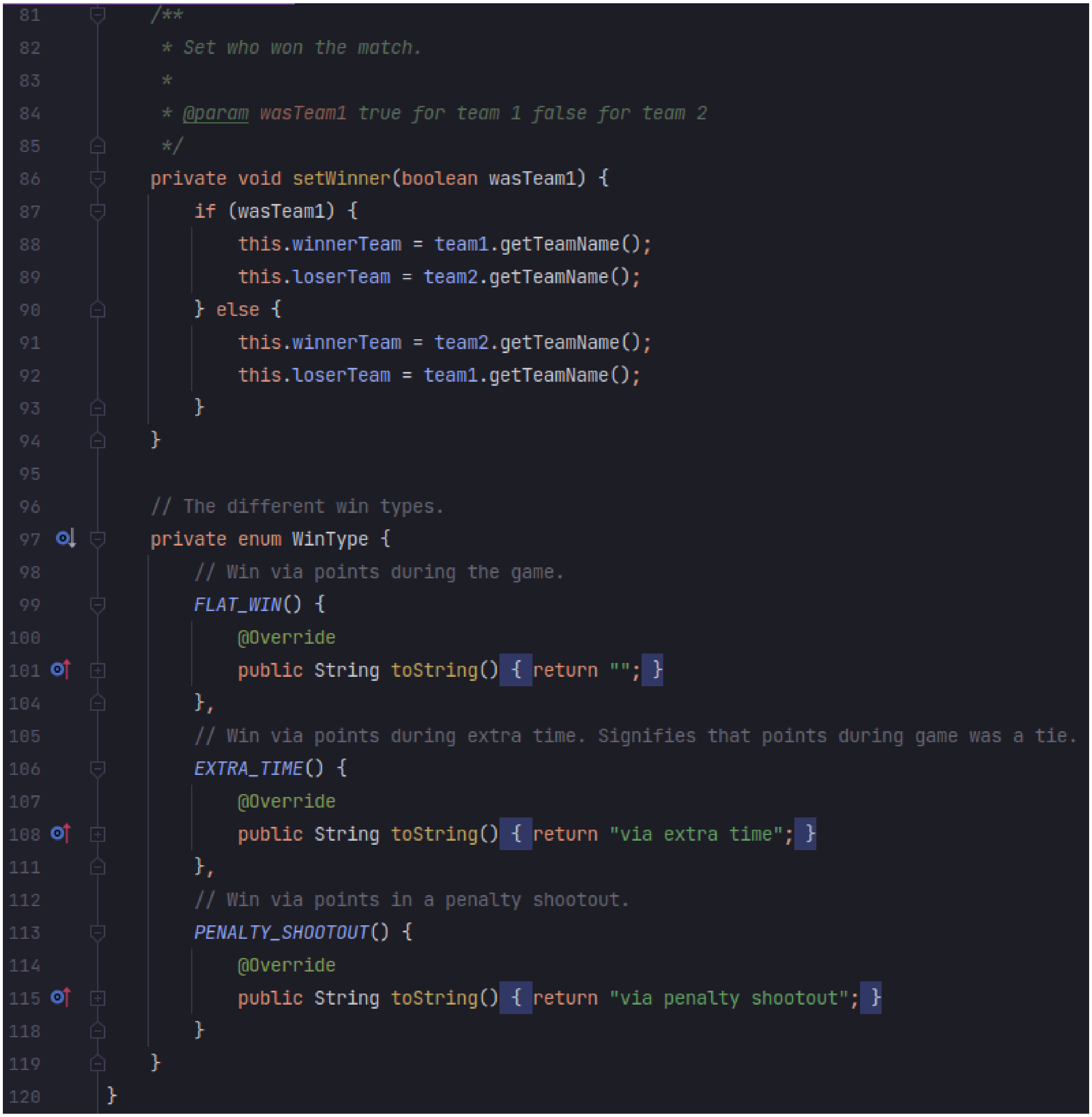


```java
public class DisplaySoccerGameResults {
    public static void main(String[] args) {
        // Create 2 teams and compare and print them using the SoccerGameResults class.
        SoccerGameScoreBoard machesterUnited = new SoccerGameScoreBoard( teamName: "Machester United",  goalsInMatch: 3,  goalsInExtraTime: 1,  goalsInPenaltyShootOut: 5);
        SoccerGameScoreBoard liverpool = new SoccerGameScoreBoard( teamName: "Liverpool",  goalsInMatch: 3,  goalsInExtraTime: 1,  goalsInPenaltyShootOut: 4);
        SoccerGameResults results = new SoccerGameResults(machesterUnited, liverpool);
        SoccerGameResults.printResults(results);

        // Create 2 new teams and compare and print them using the SoccerGameResults class.
        SoccerGameScoreBoard team1 = new SoccerGameScoreBoard( teamName: "Team1");
        SoccerGameScoreBoard team2 = new SoccerGameScoreBoard( teamName: "Team2");
        SoccerGameResults results1 = new SoccerGameResults(team1, team2);
        SoccerGameResults.printResults(results1);

        //Print the total teams created.
        System.out.println("Total teams created: " + SoccerGameScoreBoard.getCountTeams());
    }
}
```

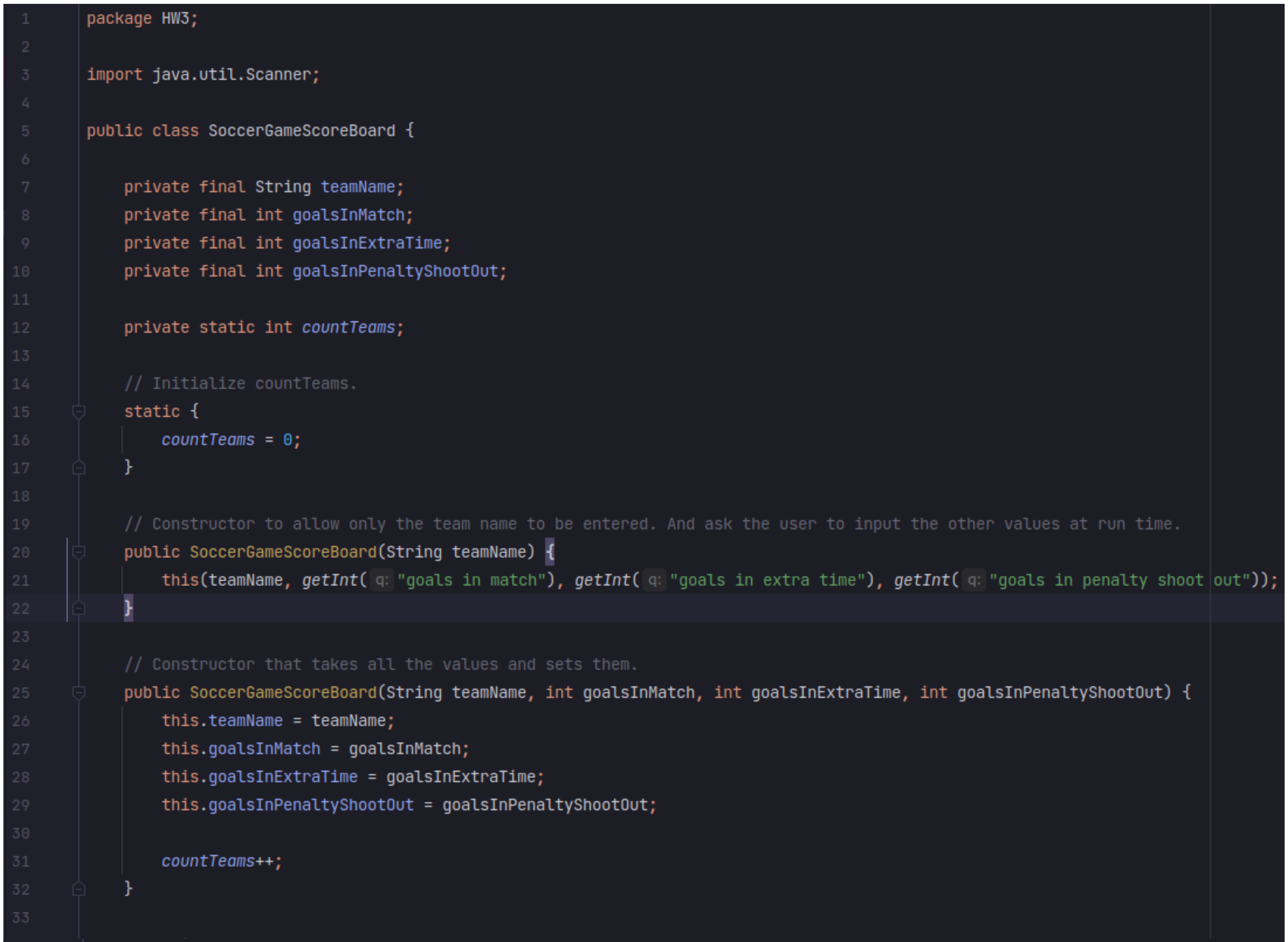


```java
package HW3;

public class SoccerGameResults {
    private final SoccerGameScoreBoard team1;
    private final SoccerGameScoreBoard team2;
    private String winnerTeam;
    private String loserTeam;
    private boolean isTie = false;
    private WinType winType;

    // Main function that acts as driver code to determine which team won.
    public SoccerGameResults(SoccerGameScoreBoard team1, SoccerGameScoreBoard team2) {
        this.team1 = team1;
        this.team2 = team2;

        // First check if the goals equal each other which signifies a possible tie.
        if (team1.getGoalsInMatch() == team2.getGoalsInMatch())
            this.isTie = true;

        // Check if one teams total goals was greater than the other teams total goals.
        if (team1.getGoalsInMatch() + team1.getGoalsInExtraTime() >
                team2.getGoalsInMatch() + team2.getGoalsInExtraTime()) {
            // If it was a tie then that means there were goals in extra time.
            if (isTie)
                this.winType = WinType.EXTRA_TIME;
            else
                this.winType = WinType.FLAT_WIN;
            setWinner(true);
            // Change isTie to false since it has not ended in a tie.
            this.isTie = false;
        } else if (team1.getGoalsInMatch() + team2.getGoalsInExtraTime() <
                team2.getGoalsInMatch() + team2.getGoalsInExtraTime()) {
            if (isTie)
                this.winType = WinType.EXTRA_TIME;
            else
                this.winType = WinType.FLAT_WIN;
            setWinner(false);
```

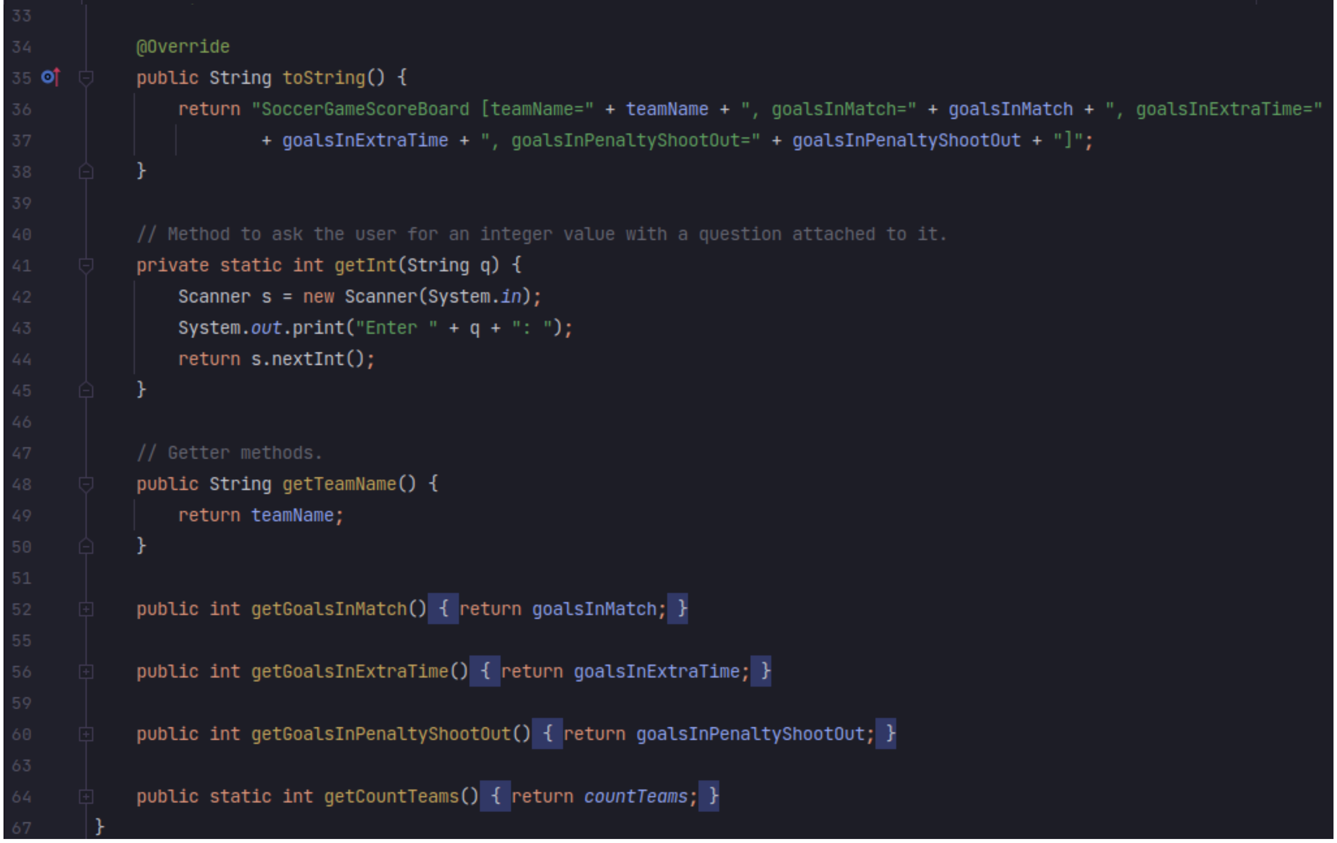
```java
            setWinner(false);
            this.isTie = false;
        }

        // Check if penalty shootout was greater.
        if (team1.getGoalsInPenaltyShootOut() > team2.getGoalsInPenaltyShootOut()) {
            this.winType = WinType.PENALTY_SHOOTOUT;
            setWinner(true);
            this.isTie = false;
        } else if (team1.getGoalsInPenaltyShootOut() < team2.getGoalsInPenaltyShootOut()) {
            this.winType = WinType.PENALTY_SHOOTOUT;
            setWinner(false);
            this.isTie = false;
        }
    }

    // Code to print the result of the game.
    public static void printResults(SoccerGameResults results) {
        if (!results.isTie())
            System.out.println(results.getWinnerTeam() + " vs " + results.getLoserTeam() +
                    " the winner is " + results.getWinnerTeam() + " " + results.getWinType()
                    ".");
        else
            System.out.println(results.team1.getTeamName() + " vs " +
                    results.team2.getTeamName() + " has ended in a tie.");
    }

    // Getter methods.
    public String getWinnerTeam() { return winnerTeam; }

    public String getLoserTeam() { return loserTeam; }

    public boolean isTie() { return isTie; }

    public WinType getWinType() { return winType; }

    /**
```



```java
    /**
     * Set who won the match.
     *
     * @param wasTeam1 true for team 1 false for team 2
     */
    private void setWinner(boolean wasTeam1) {
        if (wasTeam1) {
            this.winnerTeam = team1.getTeamName();
            this.loserTeam = team2.getTeamName();
        } else {
            this.winnerTeam = team2.getTeamName();
            this.loserTeam = team1.getTeamName();
        }
    }

    // The different win types.
    private enum WinType {
        // Win via points during the game.
        FLAT_WIN() {
            @Override
            public String toString() { return ""; }
        },
        // Win via points during extra time. Signifies that points during game was a tie.
        EXTRA_TIME() {
            @Override
            public String toString() { return "via extra time"; }
        },
        // Win via points in a penalty shootout.
        PENALTY_SHOOTOUT() {
            @Override
            public String toString() { return "via penalty shootout"; }
        }
    }
}
```



```java
package HW3;

import java.util.Scanner;

public class SoccerGameScoreBoard {

    private final String teamName;
    private final int goalsInMatch;
    private final int goalsInExtraTime;
    private final int goalsInPenaltyShootOut;

    private static int countTeams;

    // Initialize countTeams.
    static {
        countTeams = 0;
    }

    // Constructor to allow only the team name to be entered. And ask the user to input the other values at run time.
    public SoccerGameScoreBoard(String teamName) {
        this(teamName, getInt( q: "goals in match"), getInt( q: "goals in extra time"), getInt( q: "goals in penalty shoot out"));
    }

    // Constructor that takes all the values and sets them.
    public SoccerGameScoreBoard(String teamName, int goalsInMatch, int goalsInExtraTime, int goalsInPenaltyShootOut) {
        this.teamName = teamName;
        this.goalsInMatch = goalsInMatch;
        this.goalsInExtraTime = goalsInExtraTime;
        this.goalsInPenaltyShootOut = goalsInPenaltyShootOut;

        countTeams++;
    }

```



```java

    @Override
    public String toString() {
        return "SoccerGameScoreBoard [teamName=" + teamName + ", goalsInMatch=" + goalsInMatch + ", goalsInExtraTime="
                + goalsInExtraTime + ", goalsInPenaltyShootOut=" + goalsInPenaltyShootOut + "]";
    }

    // Method to ask the user for an integer value with a question attached to it.
    private static int getInt(String q) {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter " + q + ": ");
        return s.nextInt();
    }

    // Getter methods.
    public String getTeamName() {
        return teamName;
    }

    public int getGoalsInMatch() { return goalsInMatch; }

    public int getGoalsInExtraTime() { return goalsInExtraTime; }

    public int getGoalsInPenaltyShootOut() { return goalsInPenaltyShootOut; }

    public static int getCountTeams() { return countTeams; }
}
```