Player.java

```
public class Player {
    String pos;
    int points;
    int offRebounds;//Επιθετικά rebounds
    int defRebounds;//Αμυντικά rebounds
    int fg_made;
    int fg_attempted;

    public Player(String p) {
        pos = p;
        points = 0;
        offRebounds = 0; //Επιθετικά rebounds
        defRebounds = 0;//Αμυντικά rebounds
        fg_made = 0;
        fg_attempted = 0;
    }

    public void increasePoints(int p) {
        points += p;}

    //Αυξάνω τα Επιθετικά offRebounds
    public void increaseOffRebounds() {
        offRebounds++;
    }

    //Αυξάνω τα Αμυντικά defRebounds
    public void increaseDefRebounds() {
        defRebounds++;
    }

    public void increaseFG_made() {
        fg_made++;
    }

    public void increaseFG_attempted() {
        fg_attempted++;
    }

    public String getName() {
        return pos;
    }

    public int getFGattempted() {
        return fg_attempted;
    }

    public int getFGmade() {
        return fg_made;
    }

    //Getter Επιθετικά offRebounds
    public int getOffRebounds() {
        return offRebounds;
    }

    //Getter Αμυντικά defRebounds
    public int getDefRebounds() {
        return defRebounds;
    }

    //Get Points
    public int getPoints() {
        return points;
    }
}
```

Team.java

```
import java.util.Arrays;
import java.util.Random;

public class Team {

    String name;
    int score;
    Player players[] = new Player[5];

    public Team(String n) {
        name = n;
        score = 0;
        // Create the first line-up by position
        players[0] = new Player("Point Guard");
        players[1] = new Player("Shooting Guard");
        players[2] = new Player("Small Forward");
        players[3] = new Player("Power Forward");
        players[4] = new Player("Center");
    }

    public void increaseScore(int s) {
        score += s;
    }

    public String getName() {
        return name;
    }

    public int getScore() {
        return score;
    }

    public int shoot() {
        // There is a 50% chance to miss, 40% to get 2-points, and 10% for 3-points
        Random rand = new Random();
        Random pl_rand = new Random();
        int shot_outcome = rand.nextInt(1000);
        int shooter = pl_rand.nextInt(5);
        System.out.println(name + "'s " + players[shooter].getName() + " shoots");
        players[shooter].increaseFG_attempted();
        if (shot_outcome < 500) {
            return 0;
        } else if (shot_outcome < 900) {
            players[shooter].increasePoints(2);
            players[shooter].increaseFG_made();
            return 2;
        } else {
            players[shooter].increasePoints(3);
            players[shooter].increaseFG_made();
            return 3;
        }
    }

    public int offensiveRebound() {
        // There is a 20% chance to get an offensive rebound
        Random rand = new Random();
        Random pl_rand = new Random();
        int reb_outcome = rand.nextInt(1000);
        int rebounder = pl_rand.nextInt(5);
        if (reb_outcome > 800) {
            System.out.println(name + "'s " + players[rebounder].getName() + " gets
the offensive rebound");
            //increaseOffRebounds
            players[rebounder].increaseOffRebounds();
            return 1;
```

```
        } else {
            return 0;
        }
    }

    public void defensiveRebound() {
        Random pl_rand = new Random();
        int rebounder = pl_rand.nextInt(5);
        System.out.println(name + "'s " + players[rebounder].getName() + " gets the
defensive rebound");
        //increaseDefRebounds
        players[rebounder].increaseDefRebounds();
    }

    public void showStats() {
        // Show the statistics of each team in following format
        // Name
        System.out.println("Team Name :" + name);

        // Points:
        System.out.println("Points    :" + score);

        // Shoots Attempted:      Shots Made:         Percentage:
        int attempted = 0;
        int made = 0;
        //Διαπερνάω τον πίνακα με τους παίχτες και υπολογίζω τα
        //Shoots Attempted και Shots Made
        for (int i = 0; i < this.players.length; i++) {
            //Shoots Attempted
            attempted = attempted + this.players[i].fg_attempted;
            //Shots Made
            made = made + this.players[i].fg_made;
        }
        //Υπολογισμός ποσοστού
        int percentage = (made * 100) / attempted;
        System.out.println("Shoots Attempted:" + attempted + "   Shots Made:" + made +
"   Percentage:" + percentage + "%");

        // Rebounds
        int defRebounds = 0;
        int offRebounds = 0;

        //Διαπερνάω τον πίνακα με τους παίχτες και υπολογίζω τα Rebounds
        for (int i = 0; i < this.players.length; i++) {
            //Υπολογίζω τα DefRebounds και τα OffRebounds
            defRebounds = defRebounds + this.players[i].getDefRebounds();
            offRebounds = offRebounds + this.players[i].getOffRebounds();
        }
        System.out.println("DefRebounds :" + defRebounds + "  OffRebounds :" +
offRebounds);
    }

    public void showPlayersStats() {
        // Show the statistics of each player in following format
        // Position Points (... rebounds, ... / ... shoots, index)
        // index = points + rebounds - missed shots

        // Name
        System.out.println("Team Name : " + name);

        for (int i = 0; i < this.players.length; i++) {
            //Player name
            System.out.println("Player:" + this.players[i].getName());
            //Rebounds = DefRebounds + OffRebounds
            int defRebounds = this.players[i].getDefRebounds();
            int offRebounds = this.players[i].getOffRebounds();
            //Shoots Attempted
```

```
            int attempted = this.players[i].fg_attempted;
            //Shots Made
            int made = this.players[i].fg_made;

            // index = points + rebounds - missed shots
            int missed = attempted - made;
            int index = this.players[i].getPoints() + (defRebounds + offRebounds) -
missed;
            System.out.println("Position Points (" + defRebounds + " DefRebounds, " +
offRebounds + " OffRebounds, " + made + "/" + attempted + " shoots, index:" + index +
")");
        }
    }
}
```

Game.java

```
public class Game {
	Team homeTeam;
	Team awayTeam;

	public Game(Team hT, Team aT) {
		homeTeam = hT;
		awayTeam = aT;
	}

	public void simulateGame() {
		char ball = 'h';
		int score = 0;

		// We consider that a basketball game has 120 plays (3 per minute)
		for (int i=1; i<=120; i++) {
			//System.out.println(i + ": play");
			if (ball=='h') {
				// Returns if the team scores a basket (2-point or 3-
points or misses)
				score = homeTeam.shoot();
				if (score > 0) {
					System.out.println(homeTeam.getName() + "
scores " + score + " points");
				} else {
					System.out.println(homeTeam.getName() + "
misses the shot");
				}
				homeTeam.increaseScore(score);
				if (score == 0) {
					// Check if there is an offensive rebound to
continue play
					int rebound = homeTeam.offensiveRebound();
					if (rebound == 0) {
						// Check the player who gets the
rebound, and switch possession
						awayTeam.defensiveRebound();
						ball = 'a';
					}
				} else {
					ball = 'a';
				}
			} else {
				// Returns if the team scores a basket (2-point or 3-
points or misses)
				score = awayTeam.shoot();
				if (score > 0) {
					System.out.println(awayTeam.getName() + "
scores " + score + " points");
				} else {
					System.out.println(awayTeam.getName() + "
misses the shot");
				}
				awayTeam.increaseScore(score);
				if (score == 0) {
					// Check if there is an offensive rebound to
continue play
					int rebound = awayTeam.offensiveRebound();
					if (rebound == 0) {
						// Check the player who gets the
rebound, and switch possession
						homeTeam.defensiveRebound();
						ball = 'h';
					}
				} else {
					ball = 'h';
```

```
                                }
                        }
                        if (score > 0) {
                                System.out.print("\t" + homeTeam.getName() + ":" +
homeTeam.getScore() + "  -  ");
                                System.out.println(awayTeam.getName() + ":" +
awayTeam.getScore() + "\n");
                        }
                }
                if (score ==0) {
                        System.out.print("\t" + homeTeam.getName() + ":" +
homeTeam.getScore() + "  -  ");
                        System.out.println(awayTeam.getName() + ":" +
awayTeam.getScore() + "\n");
                }
        }

        public void showTeamStats() {
                System.out.println("====================================");
                homeTeam.showStats();
                System.out.println("====================================");
                awayTeam.showStats();
        }

        public void showPlayersStats() {
                System.out.println("====================================");
                homeTeam.showPlayersStats();
                System.out.println("====================================");
                awayTeam.showPlayersStats();

        }


}
```

MainClass.java

```
public class MainClass {

    public static void main(String[] args) {

        //Δημιουργήστε δυο ομάδες της αρεσκείας σας
        Team team1 = new Team("Barcelona");
        Team team2 = new Team("Real");

        // Create a Game
        Game game1 = new Game(team1, team2);

        // Simulate Game
        game1.simulateGame();

        // Show Team Statistics
        System.out.println("***** Team Statistics *****");
        game1.showTeamStats();
        System.out.println("");

        // Show Players Statistics
        System.out.println("***** Players Statistics *****");
        game1.showPlayersStats();

    }

}
```