



Dane su klase kojima se opisuju rezultati šahovskog turnira.

```
enum Result { RESULT_UNDEFINED, RESULT_WHITE_WIN, RESULT_BLACK_WIN, RESULT_DRAW; };

class Player {
    private String name;
    private int rating;
    private double points;
    private int standing;

    // ... GETTER I SETTER I

    public Player(String name, int rating) {
        this.name = name;
        this.rating = rating;
        this.points = 0.0;
        this.standing = 0;
    }
}

class Match {
    private String whitePlayer;
    private String blackPlayer;
    private Result result;

    // ... GETTER I SETTER I

    public Match(String whitePlayer, String blackPlayer, Result result) {
        this.whitePlayer = whitePlayer;
        this.blackPlayer = blackPlayer;
        this.result = result;
    }
}
```

ZADATAK: Potrebno je nadopuniti sljedeći kod koristeći lambda izraze i/ili anonimne klase, tako da se na temelju odigranih šahovskih partija (matches) popune bodovi svim igračima.

Možete pretpostaviti da su prije izvođenja navedenog koda svim igračima bodovi postavljeni na nula i da su svi igrači dodani u mapu.

```
List<Match> matches = new ArrayList();
Map<String, Player> mplayers = new HashMap();    // key = player name, value = Player object

// Popunjavanje mape s igračima i liste partija

matches.forEach(

    // OVDJE DOLAZI VAŠ KOD !!!!!!!!!!!!!

);
```

Primjer: Ako podatke popunimo sljedećim naredbama:

```
mplayers.put("Caruana", new Player("Caruana", 2820));
mplayers.put("Carlsen", new Player("Carlsen", 2847));
mplayers.put("Radjabov", new Player("Radjabov", 2765));

matches.add(new Match("Carlsen", "Radjabov", Result.RESULT_WHITE_WIN));
matches.add(new Match("Caruana", "Carlsen", Result.RESULT_BLACK_WIN));
matches.add(new Match("Caruana", "Radjabov", Result.RESULT_DRAW));
```

Nakon izvođenja gornje naredbe (koju ćete dopuniti), mapa s igračima treba sadržavati sljedeće podatke:

- Caruana - points: 0.5
- Carlsen - points: 2.0
- Radjabov - points: 0.5

NAPOMENA: Za pobjedu se dobije 1 bod, za poraz nula bodova, dok kod neriješenog rezultata svaki igrač dobije pola boda.

NAPOMENA: Kao rješenje zadatka morate predati samo odsječak programskog koda koji dolazi unutar metode forEach, ništa drugo. Stoga ne smijete koristiti import naredbe.

UPUTA: unutar metode forEach možete za svaku partiju postupno ažurirati bodove (za oba igrača).

1 match -> {
2 if (match.getResult() == Result.RESULT_BLACK_WIN) {
3 mplayers.get(match.getBlackPlayer()).setPoints(mplayers.get(match.getBlackPlayer()).getPoints() + 1);
4 mplayers.get(match.getWhitePlayer()).setPoints(mplayers.get(match.getWhitePlayer()).getPoints() + 0);
5 }
6 else if (match.getResult() == Result.RESULT_WHITE_WIN) {
7 mplayers.get(match.getWhitePlayer()).setPoints(mplayers.get(match.getWhitePlayer()).getPoints() + 1);
8 mplayers.get(match.getBlackPlayer()).setPoints(mplayers.get(match.getBlackPlayer()).getPoints() + 0);
9 }
10 else {
11 mplayers.get(match.getWhitePlayer()).setPoints(mplayers.get(match.getWhitePlayer()).getPoints() + 0.5);
12 mplayers.get(match.getBlackPlayer()).setPoints(mplayers.get(match.getBlackPlayer()).getPoints() + 0.5);
13 }
14 }

Run

Save

Correct	Incorrect	Partial	Unanswered	Score	Score %	Hint
		✓		0.875	25	Partial score

Status: Finished

Started at: June 1st 2022, 6:37:20 pm

Finished at: 6:37:23 pm

Duration: 3031 ms

Language	Compiler options	Compiler output	Message(s)
Java (OpenJDK 17.0.2)	/usr/lib/jvm/jdk-17.0.2/bin/javac -classpath /usr/lib/jvm/edgar-libs/edgar.jar:. Main.java	Note: Main.java uses unchecked or unsafe operations. Note: Recompile with -Xlint:unchecked for details.	

Tests(2):

1. Correct

2. Incorrect

-75.00%

EdgarLibrary v0.11.1
Testing functionality on a larger example
The code does not work correctly on a larger example.
[1/1] failed test