

UNIVERSITATEA DIN BUCUREȘTI
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

Metode Moderne de Calcul si Simulare **Simularea unui turneu de tenis**

Autori:

Alexandra Maria Costea, grupa 454,

Cristina Nitu, grupa 451,

Valeria Gabriela Spinu, grupa 451



2020 – 2021

Simularea unui turneu de tenis

I. Rularea proiectului

Proiectul a fost scris folosind limbajului de programare Python. Versiunea in care a fost realizat programul este Python 3.9.1 insa el poate fi rulat pe orice versiune mai mare sau egala decat 3.0.0.

Pentru a rula programul, sunt necesare instalarea urmatoarelor elemente:

- Python 3.9.1
- Intalarea unui IDE cum ar fi Pycharm  sau Visual Studio  (optional)

Rularea se poate realiza prin doua metode:

- Deschiderea programului intr-un IDE instalat, apoi rularea fisierului care contine functia main
- Folosind command prompt, program deschis din interiorul folderului in care se afla proiectul si utilizand comanda: **python main.py**

II. Prezentarea proiectului

Proiectul isi doreste sa simuleze toate etapele unui turneu de Grand Slam din circuitul feminin. Pentru acest proiect au fost selectate primele 128 de jucatoare ale lumii, conform topului stabilit de WTA. Regulile din cadrul turneului find urmatoarele:

- In fiecare tur, sunt 2^{7-n} jucatori, unde n reprezinta etapa in care se afla turneul, respectiv 2^{7-n-1} meciuri.
- Un meci este castigat atunci cand una din jucatoare castiga 2 seturi, deoarece in circuitul feminin turneele sunt de tipul 2 din 3
- Pentru a castiga un set, un jucator trebuie sa castige 6 game-uri iar diferenta de game-uri dintre el si adversar trebuie sa fie mai mare sau egala cu 2. Exista si exceptii, printre care, cazul in care un jucator incheie setul cand castiga si al 7-lea game insa diferenta dintre el si adversar este exact de 2 game-uri. In cazul in care ambele jucatoare au 6 game-uri castigate, jocul se muta in tiebreak unde, pentru a castiga, un jucator trebuie sa ajunga la 7 puncte cu o diferenta de cel putin 2 puncte. In cazul in care un jucator a ajuns la 7 puncte insa diferenta este mai mica de 2 puncte, se joaca pana ce aceasta diferenta ajunge sa fie egala cu 2.
- Un game este castigat de jucatorul care ajunge primul la punctajul 45. In cazul in care ambele jucatoare au ajuns la scorul 40-40, se joaca un avantaj care consta in 2 puncte, acel jucator care realizeaza punctele consecutiv, castiga game-ul.

III. Componentele proiectului

a. Main.py

In acest fisier se gaseste functia main, functie care ne ofera doua posibilitati de a rula programul. Prima este aceea de a simula un singur turneu, iar cea de-a doua ne permite simularea a n turnee, unde n este un input de la tastatura

```

C:\Users\User\PycharmProjects\ProiectLaboratorMMCS\venv\Scripts\p
Alegeti:
    1: Pentru simularea unui singur turneu
    2: Pentru simularea a n turnee
2
Introduceti numarul de turnee pe care doriti sa le simulati:
n= 254

```

b. Game.py

Pentru fiecare game, scorul ambelor jucatoare porneste de la 0-0. Pentru ambele jucatoare, cu ajutorul functiei *Point()*, se genereaza aleator un numar intre 0 si 1 care reprezinta probabilitatea jucatorului de a castiga punctul. Jucatorul cu probabilitatea cea mai mare castiga. Pentru a castiga un game, jucatorul trebuie sa obtine 45 de puncte. Scorul este reprezentat in felul urmator: 0-15-30-40-45. In cazul unui egal in care scorul este 40-40, se apeleaza functia *playAdvantage()* pentru a determina castigatorul game-ului, acela fiind cel care reuseste sa inscrie doua puncte consecutive.

```

Serves the first player
----->ScorGames:  Player1  3  -  2  Player2
s: 0  - r: 0
s: 0  - r: 15
s: 0  - r: 30
s: 0  - r: 40
s: 15 - r: 40
s: 15 - r: 45

Serves the second player
----->ScorGames:  Player1  3  -  3  Player2
s: 0  - r: 0
s: 0  - r: 15
s: 15 - r: 15
s: 30 - r: 15
s: 40 - r: 15
s: 40 - r: 30
s: 40 - r: 40

Play Advantage
s adv: 40 - r adv: 40
s adv: 40 - r adv: A

Serves the first player
----->ScorGames:  Player1  4  -  3  Player2

```

c. Set.py

Fiecare set incepe prin a seta numarul de game-uri jucate si castigate de fiecare jucatoare la 0. Dupa aceea, cat timp nicio jucatoare nu a atins target-ul de game-uri necesare pentru a castiga setul, 6, se joaca noi game-uri pana ce target-ul este atins si diferenta dintre jucatori este de 2 sau mai multe game-uri. In situatia in care ambele jucatoare au atins targetul, diferenta fiind de 0, se ajunge in tiebreak, unde punctele target pe care un jucator trebuie sa le atinga sunt de 7 cu o diferenta de 2. De asemenea, la inceputul fiecarui set se genereaza aleator un numar cuprins intre 0 si 1 care determina care jucator va servi prima oara, urmand ca dupa fiecare game, jucatorul care serveste sa fie schimbat.

```

Serves al doilea jucator
----->ScorGames: Player1 6 - 6 Player2
tiebreak server: 0 - 0 tiebreak returner
tiebreak server: 0 - 1 tiebreak returner
tiebreak server: 1 - 1 tiebreak returner
tiebreak server: 2 - 1 tiebreak returner
tiebreak server: 2 - 2 tiebreak returner
tiebreak server: 2 - 3 tiebreak returner
tiebreak server: 3 - 3 tiebreak returner
tiebreak server: 4 - 3 tiebreak returner
tiebreak server: 5 - 3 tiebreak returner
tiebreak server: 5 - 4 tiebreak returner
tiebreak server: 6 - 4 tiebreak returner
tiebreak server: 7 - 4 tiebreak returner
----->Punctaj set:6 - 7

```

d. Match.py

Clasa match seteaza pentru fiecare meci numarul de seturi castigate la 0 pentru ambele jucatoare. Prima jucatoare care ajunge sa castige 2 seturi este declarata castigatoarea meciului.

e. Tournament.py

In aceasta clasa avem principala metoda numita *startTournament()*, metoda din care apelam celelalte metode.

Metoda *readPlayers()* citeste linie cu linie din fisierul players.txt si adauga datele noi citite intr-o lista de jucatori.

Urmatoarele functii sunt *seed(n)* si *splitPlayers()* prin care se stabilesc capii de serie si se impart jucatorii astfel incat jucatorii de pe pozitia 1 si 2 sa se poata intalni doar in faza finala a competitiei.

Metoda *getWinner()* are ca scop intoarcerea castigatoarei competitiei. In prima etapa a competitiei participa 128 de jucatori, urmand ca in etapa 2 sa joaca 64, etapele continua pana cand ramane doar un jucator. Dupa cum am spus si putin mai sus, jucatorii sunt impartiti in 2 tabele, jocurile urmand sa se desfasoare ca in schema de mai jos.



Ultima functie este *writeCSV()* care va scrie intr-un fisier CSV toti jucatorii care au participat in turneu, contine pe primele linii toti jucatorii care au participat in prima faza a competitiei, urmatoarele linii fiind completate de jucatorii care au participat in turul 2 si asa mai departe.

f. Player.py

Clasa Player stocheaza informatii despre fiecare jucator. De asemenea aici avem si metodele *getFirstName()*, *getLastName()*, *getCountry()* si *getRank()* ce returneaza datele despre un jucator.

g. Players.txt

In acest fisier regasim informatii despre toti cei 128 de jucatori, informatii stocate pe un rand precum: numele, prenumele, tara si pozitia in clasament

h. Consola

Consola se va comporta ca un live text, deoarece, prin intermediul acesteia putem sa vedem in oricemoment ce jucator se afla la serva, care este scorul si in ce etapa a competitiei se afla jocul.

```
*****
*   FINALA   *
*****

----->Start Match
Incepe sa serveasca primul jucator
----->ScorGames:  Player1  0  -  0  Player2
s: 0  - r: 0
s: 0  - r: 15
s: 15 - r: 15
s: 15 - r: 30
s: 30 - r: 30
s: 30 - r: 40
s: 30 - r: 45

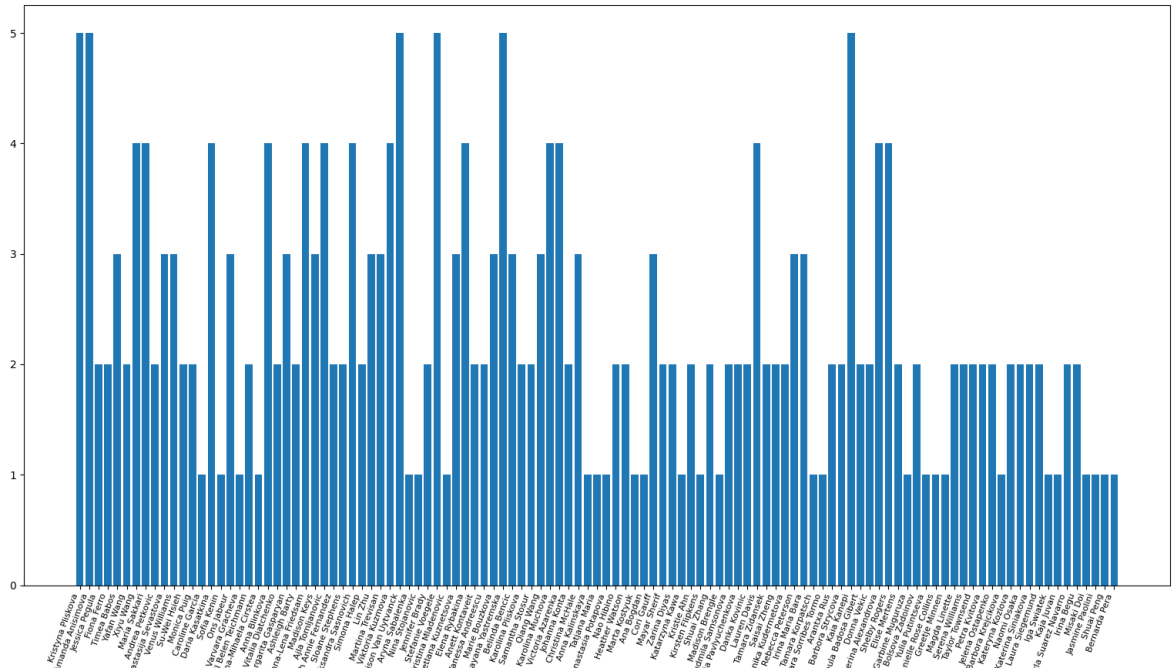
Serveste al doilea jucator
----->ScorGames:  Player1  0  -  1  Player2
s: 0  - r: 0
s: 15 - r: 0
s: 15 - r: 15
s: 15 - r: 30
s: 15 - r: 40
s: 15 - r: 45

Serveste primul jucator
----->ScorGames:  Player1  1  -  1  Player2
s: 0  - r: 0
s: 15 - r: 0
s: 30 - r: 0
s: 30 - r: 15
s: 40 - r: 15
s: 45 - r: 15

Serveste al doilea jucator
----->ScorGames:  Player1  2  -  1  Player2
s: 0  - r: 0
----->ScorGames:  Player1  3  -  6  Player2

----->Punctaj set:3 - 6
Castigatorul turneului dintre:Belinda Bencic si Simona Halep este: Simona Halep
```

De asemenea, tot in consola vom primi si timpul de executie pentru simularea a n turnee dar si graficul in care putem observa distributia datelor. Pentru simularea a 256 de turnee , timpul a fost de 0:22:39.631828. Deoarece sansa de castig este egala cu sansa de esec pentru fiecare jucatoare, observam ca pe axa x avem un numar foarte mare de castigatori obtinuti in urma generarii.



i. **TennisTournament.html**

Cu ajutorul acestui fisier, vom putea accesa interfata grafica a aplicatiei. Interfata reprezinta un cod html si un scrip java-scrip prin care citim datele din fisierul CSV si le transpunem pe ecran.



