

# ATP Tour

Prvi zadatak iz Jave (školska 2023/2024)

## Cilj zadatka:

U ovom zadatku je potrebno implementirati simulaciju ATP Tour šampionata. Turniri koji mogu činiti šampionat, kao i teniseri koji mogu učestvovati na njima, su izlistani u odgovarajućim tekstualnim fajlovima. Prilikom pokretanja simulacije, ove liste se očitavaju, a zatim se, turnir po turnir, simuliraju svi mečevi - set po set, gem po gem, poen po poen, uz razne parametre koji mogu uticati na ishod svakog meča, ali i dalji tok šampionata. U toku celog šampionata se vodi evidencija o osvojenim ATP poenima i rankiranju tenisera, a na samom kraju šampionata će najboljih 8 tenisera odmeriti svoje snage na finalnom ATP Finals turniru.

## Opis zadatka:

Organizator celog šampionata, u okviru ove simulacije, će biti korisnik, koji na samom početku bira koliko turnira će činiti dati šampionat. Radi pojednostavljivanja simulacije, lista turnira je ograničena samo na Grend Slem turnire i ATP Masters 1000 turnire. Dakle, **maksimalan prihvaćen broj turnira**, od strane korisnika, **je 13**, a **minimal prihvaćen broj je 4**, pošto ćemo zahtevati da u šampionatu moraju biti odigrana bar 4 turnira.

Nakon uspešnog unosa broja turnira, prelazi se na simulaciju svakog od njih. Organizator, odnosno korisnik, sada ima mogućnost da bira redosled turnira, tako što unosi naziv turnira koji će se prvo simulirati, pa nekon njega unosi naziv sledećeg, i tako dalje, pri čemu **nije moguće ponavljanje turnira**. Nakon što se završi odgovarajući broj turnira (koji je korisnik inicijalno uneo), automatski počinje ATP Finals turnir. Svaki turnir ima tačno određenu podlogu na kojoj se igraju mečevi, koja je naznačena za svaki turnir u okviru tekstualnog fajla sa listom turnira (*tournaments.txt*). Pored toga, u fajlu je, za svaki turnir, naznačeno da li spada u Grand Slam turnire ili ne, **od čega će zavisiti broj setova koji se igraju u mečevima, kao i broj ATP poena koje takmičari mogu osvojiti u toku turnira**. S obzirom da se ATP Finals turnir automatski pokreće i nije deo liste turnira, njegovi parametri (podloga, broj setova koji se igra u mečevima, itd.) će biti dati kasnije u tekstu zadatka.

Lista tenisera koji mogu učestvovati u šampionatu je takođe data u zasebnom tekstualnom fajlu (*players.txt*), pri čemu su oni izlistani na osnovu trenutnih pozicija na ATP listi za rankiranje, sa kojima ulaze u šampionat. Takođe, za svakog tenisera je dat i njegov glavni adut (forehand, backhand, servis, mentalitet), kao i podloga na kojoj voli da igra – ovi parametri će uticati na verovatnoću ishoda svakog poena na turnirima. Radi pojednostavljivanja simulacije, lista tenisera je ograničena na prvih 16 na ATP listi i samo oni mogu učestvovati na turnirima. Samim time, **simulacija svakog turnira počinje od osmine finala**.

Na početku svakog turnira, potrebno je formirati nasumičan raspored takmičara, odnosno **nasumične duele (nevezavisno od ATP pozicije tenisera) koji će igrati u osmini finala**. Porednici iz ovih duela zatim igraju četvrtinu finala takmičenja, a porednici iz tih duela igraju polufinale. Na kraju, porednici iz polufinala će se boriti u finalu za prvo i drugo mesto na takmičenju. Dužina mečeva

zavisi od toga na kojem tipu turnira se igraju, pa tako za Grand Slam turnire, mečevi se igraju na 3 dobijena seta, a za preostale, na 2 dobijena seta. Kao što je rečeno, u okviru svakog seta je potrebno odraditi simulaciju gem po gem, poen po poen, **pri čemu može doći do tie-break situacija u okviru seta, kao i deuce situacija u okviru gema**. Ishod svakog poena, odnosno da li će teniser koji servira u datom gemu osvojiti poen, zavisi od verovatnoće, na koju utiču određeni parametri. O ovim parametrima, kao i njihovom uticaju na ishod poena, će biti više reči u sekciji Pravila simulacije.

Nakon svakog odigranog kola, ispisati rezultate mečeva. Ispod je dat primer ispisa jednog kola:

===== Semi finals =====

Novak Djokovic    7 3 5 5    1

Jannik Sinner    6 6 7 7    3

Andrey Rublev    7 4 6 4    1

Holger Rune    6 6 7 6    3

U ispisu je potrebno prikazati, za svaki meč, ishod svakog seta, kao i ukupan broj osvojenih setova svakog igrača.

Nakon završenog turnira, ispisati trenutnu situaciju na ATP listi, nakon čega se zahteva unos sledećeg turnira za simulaciju.

Nakon što su svi turniri završeni, korisniku se ispisuje da počinje ATP Finals turnir i, nakon potvrde korisnika, kreće simulacija tog turnira. Tok ovog turnira se razlikuje od ostalih turnira, pošto je sada potrebno **prvih 8 takmičara, na ATP listi, grupisati u 2 grupe (grupa A i grupa B)**, na sledeći način: grupu A čine teniseri sa neparnom pozicijom na ATP listi (1. mesto, 3. mesto, 5. mesto, 7. mesto), a grupu B čine teniseri sa parnom pozicijom na ATP listi (2. mesto, 4. mesto, 6. mesto, 8. mesto). Unutar ovih grupa, takmičari igraju mečeve u round-robin maniru, odnosno svako sa svakim, a prva dva mesta iz obe grupe će činiti polufinaliste. Pobednici polufinala zatim igraju finale za prvo i drugo mesto.

Za ATP Finals turnir nije potrebno ispisivati ishode mečeva, već je dovoljno ispisati tenisere grupe A i grupe B, zatim polufinaliste, a zatim finaliste, kao i konačnog pobednika turnira.

Nakon završenog ATP Finals turnira, ispisati konačno stanje na ATP listi.

## Pravila bodovanja tenisera na turnirima:

U zavisnosti od toga u kojem kolu teniser **završi** svoje učešće na turniru, kao i koji tip turnira je u pitanju, teniseri dobijaju sledeće iznose ATP poena:

Kolo	Grand Slam	Masters1000
Osmina finala (roundOf16)	180	100
Četvrtina finala	360	200
Polufinale	720	400
2. mesto	1200	650
1. mesto	2000	1000

Na **ATP Finals** turniru bodovanje funkcioniše malo drugačije:

- Za svaku pobjedu na početku turnira, unutar grupe, takmičar dobija 200 ATP poena.
- Za pobjedu u polufinalu takmičar dobija 400 poena.
- Za pobjedu u finalu takmičar dobija 500 poena.

## Pravila simulacije:

1. Lista turnira, koji mogu da se biraju za šampionat, je data u fajlu *tournaments.txt*. Svaki turnir ima svoj naziv, podlogu na kojoj se igra, kao informaciju o tome da li spada u Grand Slam turnir ili Masters1000 – ovo je sve potrebno pročitati iz fajla.
2. Mečevi na ATP Finals turniru se igraju na 2 dobijena seta i na betonu.
3. Ukoliko turnir spada u Grand Slam turnir, mečevi se igraju na 3 dobijena seta (prvi teniser koji osvoji 3 seta je pobedio), a u suprotnom, mečevi se igraju na 2 dobijena seta.
4. Prilikom igranja jednog seta, teniseri se međusobno smenjuju na servisu, odnosno, u prvom gemu servira prvi teniser, u sledećem gemu servira drugi teniser, pa u narednom gemu prvi teniser, i tako dalje.
5. Ukoliko neki teniser, u toku jednog seta, osvoji 6 gema, a protivnik ima 4 ili manje osvojenih gemova – teniser koji je osvojio 6 gema osvaja dati set.
6. Ukoliko neki teniser, u toku jednog seta, osvoji 6 gema, a protivnik ima 5 osvojenih gemova, igra se normalno naredni gem i od ishoda tog narednog gema razlikujemo dve situacije:
  - a. Teniser, koji je prethodno imao 6 osvojenih gemova, je osvojio naredni gem. Rezultat je onda 7 prema 5 u gemovima, pa taj teniser osvaja i dati set.
  - b. Teniser, koji je prethodno imao 5 osvojenih gemova, je osvojio naredni gem. Rezultat je onda 6 prema 6 u gemovima, što rezultuje u *tie-break* situaciji, koja će biti opisana u narednoj tački.
7. Ukoliko je rezultat u gemovima, u toku jednog seta, 6 prema 6, imamo *tie-break* situaciju i potrebno je odigrati još jedan, odlučujući gem. U ovom odlučujućem gemu, igra se dokle god

neki teniser ne dođe prvi do 7 osvojenih poena. Ukoliko se desi situacija da oba tenisera imaju 6 osvojenih poena, igra se na razliku, odnosno pobednik gema, pa i celog tog seta, će biti teniser koji osvoji 2 poena više u odnosu na svog protivnika (npr. 8 prema 6, ili npr. 10 prema 8). Takođe, smatraćemo da se u toku *tie-break* gema teniseri stalno smenjuju na servisu, posle svakog odigranog poena.

8. Ukoliko je rezultat u poenima, u okviru gema, 40 prema 40 (*deuce*) – igra se na razliku, odnosno, gem osvaja teniser koji osvoji za dva poena više u odnosu na protivnika.
9. Simulaciju svakog poena je potrebno odraditi u odnosu na to kolika je verovatnoća da određeni teniser osvoji poen. Ova verovatnoća je inicijalno 50%, a na nju utiču aduti svakog tenisera, kao i njegova pozicija na ATP listi, tako što će se inicijalna verovatnoća od 50% umanjivati ili povećavati za određeni broj. Ovo će biti opisano u narednim tačkama:
10. Aduti tenisera i njihov uticaj na ishod poena:
  - a. **Backhand** – povećava teniseru verovatnoću za 8% da osvoji poen, u situaciji kada protivnik servira (gledano iz ugla protivnika, to umanjuje protivniku verovatnoću za 8% da osvoji poen na svoj servis);
  - b. **Forehand** – povećava teniseru za 10% verovatnoću da osvoji poen na svoj servis;
  - c. **Servis** – povećava teniseru za 15% verovatnoću da osvoji poen na svoj servis. Takođe, umanjuje za 5% verovatnoću da teniser osvoji poen, u situaciji kada protivnik servira (gledano iz ugla protivnika, to uvećava protivniku verovatnoću za 5% da osvoji poen na svoj servis);
  - d. **Mentalitet** – povećava teniseru za 10% verovatnoću da osvoji poen, u situaciji kada protivnik servira (gledano iz ugla protivnika, to umanjuje protivniku verovatnoću za 10% da osvoji poen na svoj servis). Takođe, povećava teniseru verovatnoću za 5% da osvoji poen na svoj servis.
11. Uticaj ostalih parametara na ishod poena:
  - a. Ukoliko se meč igra na podlozi koja odgovara takmičaru, to teniseru povećava verovatnoću za 5% da osvoji poen na svoj servis;
  - b. Verovatnoća tenisera da osvoji poen se umanjuje zarazliku pozicija tog tenisera i protivnika na ATP listi (na primer, ako je teniser 8. na ATP listi, a protivnik 1. na listi, njegova konačna verovatnoća da osvoji poen će biti umanjena za 7%);
  - c. Na početku svakog meča, postoji 1% verovatnoća da **jedan** od tenisera zadobije povredu zbog koje automatski predaje meč i završava turnir. **U slučaju da se ovo desi, ispisati informaciju o tome da je došlo do povrede i kojeg tenisera;**
12. Nakon što se utvrdi konačna verovatnoća da teniser osvoji dati poen, generisati nasumičan broj od 1 do 100 i proveriti, na osnovu verovatnoće, ishod poena. Na primer, ukoliko je verovatnoća da teniser osvoji poen jednaka 30%, proveriti da li je generisan broj manji ili jednak sa 30 – ako jeste, teniser je osvojio poen, ako nije, protivnik je osvojio poen.

### Primer jednog duela:

Novak Đoković (1. na ATP listi) vs Daniil Medvedev (4. na ATP listi)

Ukoliko servira Novak Đoković, njegova konačna verovatnoća da osvoji poen je:

$50\%$  (inicijalna verovatnoća) +  $5\%$  (adut Medvedeva je servis, što mu daje  $15\%$  veću verovantoću kada on servira, ali u situaciji kada protivnik servira onda protivnik ima  $5\%$  veću verovantoću da osvoji poen) +  $5\%$  (adut Đokovića je mentalitet) +  $3\%$  (razlika pozicija na ATP listi) =  $63\%$ . Ukoliko bi se meč igrao na betonu, ova vrednost bi se uvećala još za  $5\%$ , jer Đoković preferira tu podlogu.

Ukoliko servira Daniil Medvedev, njegova konačna verovatnoća da osvoji poen je:

$50\%$  (inicijalna verovatnoća) –  $10\%$  (adut Đokovića je mentalitet) +  $15\%$  (adut Medvedeva je servis) –  $3\%$  (razlika pozicija na ATP listi) =  $52\%$ . Ukoliko bi se meč igrao na betonu, ova verovatnoća bi se dodatno uvećala za  $5\%$ , jer Medvedev preferira tu podlogu.

U poenima u kojima bi servirao Đoković bi se generisao random broj od 1 do 100 i proveravalo da li je dobijen broj  $\leq 63\%$ , pa ako jeste, Đoković je osvojio poen, a ako nije, Medvedev je osvojio poen.

U poenima u kojima bi servirao Medvedev bi se generisao random broj od 1 do 100 i proveravalo da li je dobijen broj  $\leq 52\%$ , pa ako jeste, Medvedev je osvojio poen, a ako nije, Đoković je osvojio poen.

### Format datoteke *players.txt*:

<ATP pozicija>,<ime i prezime>,<adut tenisera>,<podloga koju preferira>,<broj ATP poena>

### Format datoteke *tournaments.txt*:

<naziv turnira>,<podloga na kojoj se igra>,<tip turnira (Grand Slam ili Masters 1000)>

## Opis za klase i implementacije:

### 1. Player

Atributi (svi privatni):

- String name
- String ability
- String preferredSurface
- int atpRank
- int atpPoints
- boolean injured

Metode:

- getteri i setteri za sve attribute
- Konstruktor(i) po vašem izboru
- public int servePointChance (Player opponent, String surface) – izračunava kolika je verovatnoća da teniser osvoji poen na svoj servis, na osnovu parametara tog igrača, prosleđenog protivničkog igrača i podloge na kojoj se igra
- public String toString() – override-ovana metode klase Object. Može se koristiti za jednostavan ispis igrača, kada je to potrebno

### 2. Match

Atributi (svi privatni):

- Player p1 – prvi igrač u meču
- Player p2 – drugi igrač u meču
- String matchSurface – podloga na kojoj se igra meč
- int winSetNum – u koliko dobijenih setova se igra dati meč
- int p1Sets, p1Gems, p2Sets, p2Gems – indikatori o broju osvojenih gemova u setu oba igrača, kao i o broju osvojenih setova u meču
- int p1ScorePerSet[], p2ScorePerSet[] – nizovi koji čuvaju rezultate gemova po setovima; može se koristiti za ispis konačnog rezultata meča
- Random rng – koristiće se za generisanje slučajnoj broja za proveru ishoda poena

Metode:

- Konstruktor(i) po vašem izboru
- public Player playMatch() – započinje meč
- private void playSet() – započinje set
- private void playGame() – započinje gem
- private void playTieBreak() – započinje tie-break (ukoliko dođe do njega)
- private boolean chanceEvent(int probability) – koristi „rng“ za generisanje random broja od 1 do 100 i proverava, u odnosu na prosleđenu verovatnoću da se neki događaj desi, da li se događaj desio (npr. osvajanje poena od strane igrača)
- public printMatchResult() – ispisuje rezultat meča (videti primer ispisa iznad, u tekstu zadatka)

### 3. Apstraktna klasa Tournament

Atributi (svi protected):

- String tourName
- String tourType
- String tourSurface
- boolean playable – može pomoći kod unosa turnira za simulaciju od strane korisnika, kako se ne bi pokretali turniri koji su već odigrani
- int numOfSets – u koliko dobijenih setova se igra dati turnir
- ArrayList<Player> contestants

Metode:

- Konstruktor(i) po vašem izboru
- Proizvoljni getteri i setteri
- abstract public void play() – služi za pokretanje simulacije turnira, a njena implementacija će zavisiti od vrste turnira, tako da mora da se override-uje u klasama naslednicama (AtpFinals i SeasonTournament)

### 4. SeasonTournament (nasleđuje klasu Tournament)

Atributi (svi privatni):

- ArrayList<Player> roundOf16
- ArrayList<Player> quaterFinalists
- ArrayList<Player> semiFinalists
- ArrayList<Player> finalists

Ovo su liste igrača u svakom kolu turnira, koje vam mogu pomoći u realizaciji simulacije, a i davanju ATP poena igračima na osnovu uspeha.

Metode:

- Konstruktor(i) po vašem izboru
- public void play() – override-ovana metoda klase Tournament, koja služi za pokretanje simulacije turnira. Na ovom tipu turnira je potrebno simulirati osminu finala (roundOf16), četvrtinu finala, polufinale i finale
- Proizvoljne metode po potrebi, za realizaciju simulacije turnira

### 5. AtpFinals (nasleđuje klasu Tournament)

Atributi (svi privatni):

- ArrayList<Player> groupA
- ArrayList<Player> groupB
- ArrayList<Player> semiFinalists
- ArrayList<Player> finalists

Metode:

- Konstruktor(i) po vašem izboru
- `public void play()` – override-ovana metoda klase `Tournament`, koja služi za pokretanje simulacije turnira. Na ovom tipu turnira je potrebno prvo simulirati mečeve po grupama A i B (unutar grupe svako igra sa svakim, jedanput), nakon čega se formiraju polufinalni parovi tako što se prvoplasirani igrač iz grupe A uparuje sa drugoplasiranim iz grupe B, i obrnuto, prvi iz grupe B sa drugim iz grupe A. Pobjednici iz polufinala zatim igraju finalni meč.
- Proizvoljne metode po potrebi, za realizaciju simulacije turnira

## 6. Championship

Atributi (svi privatni):

- `ArrayList<Player> players` – lista svih igrača na šampionatu
- `ArrayList<Tournaments> tournaments` – lista svih turnira koji mogu da se igraju

Metode:

- Konstruktor(i) po vašem izboru
- getteri i setteri za attribute
- `public void updateAtpRanks()` – služi da se, nakon svakog odigranog turnira, ažuriraju pozicije igrača na ATP listi
- `public void recoverPlayers()` – služi da se, nakon svakog odigranog turnira, oporave igrači koji su zadobili povredu na odigranom turniru (kako bi mogli da igraju naredni)
- `public void loadFiles()` – služi za očitavanje liste igrača i turnira iz datih fajlova (`players.txt` i `tournaments.txt`) u liste `"players"` i `"tournaments"`

## 7. Simulate

Klasa unutar koje se nalazi `main()` metoda od koje se pokreće program. U okviru `main` metode je potrebno odraditi sledeće:

1. Instanciraj objekat klase `Championship`
2. Pozvati `loadFiles()` metodu nad instanciranim objektom
3. Zahtevati od korisnika unos koliko turnira će se igrati na šampionatu. Ukoliko korisnik unese broj van dozvoljenog opsega, zahtevati novi unos. Ukoliko korisnik unese neki string, a ne broj, ispisati poruku o pogrešnom unosu i zahtevati novi unos.
4. Nakon ispravnog unosa broja turnira, od korisnika se traži unos naziva turnira koji će se igrati, pri čemu je potrebno voditi računa o kojima je bilo reč ranije (ne sme se jedan turnir igrati više od jednom; nakon turnira ažurirati pozicije igrača na listi; ...). Ovaj korak je potrebno ponavljati sve dok se ne odigra uneti broj turnira. **Teniseri se uvek sortiraju na osnovu akumuliranih ATP poena – da bi se ovo izvelo, koristiti `Collections.sort()` i implementirati interfejs `comparable/comparator` u klasi `Player`.**



5. Nakon odigranih turnira, ispisati stanje na ATP listi i obavestiti korisnika da počinje ATP Finals turnir. Nakon potvrde od strane korisnika (bilo kojeg unosa), započeti simulaciju ATP Finals turnira.
6. Nakon završenog ATP Finals turnira ispisati ponovo stanje na ATP listi, kao i informaciju o tome koji teniser je završio godinu na poziciji broj 1.

### **Ocenjivanje (ukupno 25 poena):**

- Implementiran korisnički interfejs (odabir broja turnira u šampionatu, odabir jednog po jednog turnira za simulaciju, ispisivanje adekvatnih poruka, stanju na kraju svakog turnira i pobedniku nakon završenog šampionata) - **5 poena**
- Inicijalizacija kolekcija players i tournaments u klasi Championship iz datoteka players.txt i tournaments.txt - **3 poena**
- Implementacija simulacije meča (simulacija poena, gema, seta, *tie-break*, *deuce*, dobijanje pobednika, ...) – **5 poena**
- Simulacija SeasonTournament turnira (simulacija svakog kola turnira) – **4 poena**
- Simulacija ATP Finals turnira (simulacija grupnog dela turnira, polufinala i finala) – **3 poena**
- Sortiranje tenisera na kraju svakog turnira (Collections.sort()), ažuriranje atpPoints atributa i ažuriranje renkinga na osnovu toga za svakog od tenisera) – **5 poena**