

F1 Grand Prix

Prvi zadatak iz Jave

Cilj zadatka: U okviru ovog zadatka napravićemo simulaciju Grand Prix šampionata Formule 1. Cilj zadatka će biti da učitamo vozače i staze na početku, a onda, krug po krug, trku po trku, do kraja šampionata, uz mnoštvo raznih slučajnih događaja pratimo šta se dešava sa našim vozačima, evidentiramo pobednike i rangiranje za svaku trku i na kraju dobijemo pobednika šampionata F1 za datu sezonu.

Opis zadatka: Simulacija GP šampionata počinje upitom organizatoru zahtevajući da se unese broj trka u sezoni. Prihvaćen odgovor je 3-5 trka u jednom šampionatu (ponavlja se pitanje sve dok odgovor nije odgovarajući), nakon čega se prelazi na simulaciju svake pojedinačne trke.

Pre početka svake trke, organizator šampionata određuje stazu na kojoj se održava trka. Postoji ukupno 8 staza koje dolaze u obzir i svaka od njih može biti domaćin samo jedne trke tokom šampionata. Trka na određenoj stazi ima fiksni broj krugova, određen samom stazom (broj krugova se razlikuje od staze do staze). Podaci o stazama se učitavaju iz tekstualne datoteke *staze.txt* koja će vam biti data.

Lista vozača koji se takmiče u šampionatu je takođe pročitana iz tekstualne datoteke *vozac.txt*. Na početku trke, svaki vozač dobija poziciju u skladu sa svojim renkingom (brojem poena osvojenih do početka date trke za svaku trku osim prve, odnosno inicijalnog renkinga za prvu trku šampionata), tako da najbolju poziciju u datoj trci dobija vozač sa najboljim renkingom. Pozicija na početku trke određuje inicijalno vremensko kašnjenje dato u tabeli ispod. Vozač na pol poziciji (prva pozicija) nema inicijalno kašnjenje, ali svi ostali vozači će ga imati:

Renking	Početna pozicija	Vremenska kazna (s)
1	1. red (pol pozicija)	0
2	1. red (pored pol pozicije)	3
3	2. red (iza pol pozicije)	5
4	2. red	7
5 i više (svi ostali)	pozadina	10

Trka započinje tako što svaki vozač pokušava da odvoza zahtevani broj krugova u najkraćem mogućem vremenu. Nakon svakog kruga, vreme se računa i dodaje na akumulirano vreme za datu trku. Svaki vozač će dobiti dodeljeno srednje vreme vožnja trke (pročitano iz datoteke *staze.txt* za datu stazu) - međutim, tokom svakog kruga postoje slučajni događaji koji mogu da utiču na vreme vožnje kruga, ili čak na to da vozač bude eliminisan iz trke šampionata. **Ovi događaji moraju biti ispisivani na ekranu kako se budu dešavali.** Na kraju svakog kruga, ime i vreme vodećeg vozača će biti prikazano, pri čemu će vodeći vozač biti onaj koji ima najmanje akumulirano vreme.

Kada se završe svi krugovi, vozači koji su imali 4 najkraća akumulirana vremena u trci biće nagrađeni poenima u skladu sa tabelom ispod. Ukoliko se desi da su dva ili više vozača imali isto akumulirano vreme, pozicije (i samim tim poeni) se dodeljuju na slučajan način, nakon čega se rezultati trke prikazuju na ekranu.

Mesto	Poeni
1.	8
2.	5
3.	3
4.	1

Nakon što se sve trke završe, pobednik šampionata je vozač koji je osvojio najviše akumuliranih poena (sabranih kroz sve trke u šampionatu). Konačni rezultati šampionata će biti prikazani na ekranu.

Pravila za trku i šampionat:

1. Staze na kojima se održavaju trke se čitaju iz datoteke *staze.txt* na početku šampionata. Svaka staza ima svoje ime, broj krugova koji se voze, prosečno vreme vožnje jednog kruga u sekundama i verovatnoću kiše izraženu u procentima;
2. Prosečno vreme vožnje jednog kruga varira od 60 do 90 sekundi. Svaki vozač vozi svoj bolid tačno toliko vremena u svakom krugu, ali stvarno vreme vožnje jednog kruga zavisi od slučajnih događaja koji se dešavaju u svakom krugu;
3. Podaci o vozačima se čitaju iz datoteke *vozaci.txt* na početku šampionata. Svaki vozač ima svoje ime, renking, specijalne veštine, mogućnost da se takmiči i akumulirane poene. Na početku šampionata, svaki vozač može da se takmiči i ima 0 akumuliranih poena;
4. Svaki vozač ima jednu od tri specijalne veštine: kočenje, ulazak u krivinu i preticanje. Korišćenjem ovih veština, može dovesti do toga da završi svaki krug trke brže od prosečnog vremena vožnje jednog kruga:
 - a) Kočenje i ulazak u krivinu mogu da se koriste u svakom krugu i svaki vozač koji ima jednu od ove dve veštine skратиće vreme vožnje jednog kruga za 1 - 8 sekundi (slučajno generisan broj)
 - b) Preticanje se može koristiti u svakom trećem krugu, a svaki vozač koji ima ovu veštinu skратиće vreme vožnje kruga za 10-20 sekundi.
5. Tokom svakog kruga, postoji verovatnoća da svakog vozača zadesi jedan od sledećih problema
 - a) 5% verovatnoće da će bolid imati manji mehanički kvar što rezultuje dodatkom od 20 sekundi na vreme vožnje kruga;
 - b) 3% verovatnoće da će bolid zadesiti ozbiljan mehanički kvar što rezultuje dodatkom od 120 sekundi na vreme vožnje kruga;
 - c) 1% verovatnoće da će bolid zadesiti mehanički kvar koji nije otklonjiv, što će za posledicu imati da taj vozač ne može više da se takmiči u toj trci.

6. Na početku prve trke šampionata, renking vozača se čita iz datoteke *vozaci.txt*. Nakon svake trke, renking vozača se ažurira i koristi prilikom određivanja pozicija vozača u sledećoj trci, pri čemu su prva četiri vozača dobili renking 1 - 4, a svi ostali vozači renking 5. **Napomena: Da bi se vozači sortirali na kraju trke, potrebno je u klasi vozač implementirati interfejs *comparable* (ili interfejs *comparator*, oba primera imate u kodovima uz predavanje Nasledjivanje, kolekcije, obrada izuzetaka) kako bi se vozači sortirali korišćenjem metode *Collections.sort()***
7. Svi bolidi kreću trku sa pneumaticima za suvo vreme. Tokom drugog kruga trke, svaki vozač ima 50% šanse da zameni pneumatike onima za vlažno vreme koji se koriste u slučaju kiše. Ova zamena pneumatika dodaje 10 sekundi na vreme vožnja tog kruga. Verovatnoća kiše u svakom krugu zavisi od staze na kojoj se održava trka i u opsegu je 1% - 20%. Ako padne kiša, vozači koji imaju pneumatike za suvo vreme voziće svoj krug 5 sekundi duže (5 sekundi im je dodato na vreme vožnje kruga).

Klasama ćemo ipak davati nazive na engleskom jeziku, pre svega zbog getter-a i setter-a koje ćemo koristiti. Klase koje ćete koristiti (za svaku klasu navedeni su samo zahtevani atributi i metode, ako treba dodati po potrebi):

1. Driver

atributi (svi privatni) :

- String **name**
- int **ranking**
- String **specialSkill** (kočenje - braking, ulazak u krivinu - cornering ili preticanje - overtaking)
- boolean **eligibleToRace** (biće postavljen na *false* ako se desi mehanički kvar koji nije otklonjiv)
- int **accumulatedTime**
- int **accumulatedPoints**

Metode:

- setteri i getteri za sve attribute
- konstruktor(i) po vašem izboru
- public void **useSpecialSkill**(RNG rng) korišćenjem klase za generisanje slučajnog broja skрати vreme vožnje kruga (koliko sekundi se skraćuje određeno je korišćenjem objekta *rng* - dole će biti objašnjena klasa RNG)

2. Venue

atributi (svi privatni)

- int **averageLapTime**

- double **chanceOfRain**
- int **numberOfLaps**
- String **venueName**

metode

- getteri i setteri za sve attribute
- konstruktor(i) po vašem izboru

3. Championship

atributi (svi privatni)

- ArrayList<Driver> **drivers**
- ArrayList<Venue> **venues**

konstante

- final int **MINOR_MECHANICAL_FAULT** = 5
- final int **MAJOR_MECHANICAL_FAULT** = 3
- final int **UNRECOVERABLE_MECHANICAL_FAULT** = 1

metode

- setteri i getteri za attribute
- konstruktor po vašoj želji (neophodno je barem učitati podatke iz datoteka *vozaci.txt* i *staze.txt*)
- void **prepareForTheRace()** - inicijalizacija atributa za svakog vozača
- void **driveAverageLapTime()** - svakom vozaču koji vozi trku, dodeli srednje vreme voženja kruga (određeno samom stazom)
- void **applySpecialSkills()** - primeni specijalne veštine za svakog vozača u datom krugu
- void **checkMechanicalProblem()** - proveriti da li su neki vozači imali kvar (jedan od tri) i ako treba ažuriraj im vreme voženja kruga, odnosno atribut **eligibleToRace**
- void **printLeader**(int lap) - ispiši ko je na prvom mestu nakon kruga *lap*
- void **printWinnersAfterRace**(String venueName) - ispiši imena pobednika (četiri najbolje rangirana vozača) na stazi *venueName*
- void **printChampion**(int numOfRaces) - ispisati poruku o tome ko je šampion na kraju šampionata, tj. nakon *numOfRaces* odvozenih trka

4. RNG

atributi (svi privatni)

- int **minimumValue**
- int **maximumValue**
- Random **rnd**

metode

- getteri i setteri za atribut
- konstruktor(i) po vašoj želji
- int **getRandomValue()** vraća slučajan broj u opsegu minimumValue - maximumValue. Ovo je zapravo suština ove klase, obzirom da klasa Random vraća broj u opsegu 0 - max, a nama je u nekim slučajevima (definisano zadatkom) potrebno da generišemo brojeve u opsegu min - max (npr 10-20 ili 1 – 8). Na vežbama ćemo vam pokazati kako se koristi klasa Random.

5. Simulate

klasa u kojoj se nalazi main metoda od koje se pokreće program. U okviru main metode potrebno je implementirati sledeće:

1. instanciraj objekat klase Scanner koji će se koristiti za unos podataka sa tastature
2. kreiraj objekat klase Championship (u konstruktoru pročitaj vozače i staze iz odgovarajućih datoteka)
3. Postavlja pitanje koliko će se trka voziti u trenutnom šampionatu sve dok odgovor nije u opsegu 3-5
4. Pitaj za mesto održavanja trenutne trke i vodi računa da ta staza ne može biti ponuđena prilikom odabira staze u narednim trkama tekućeg šampionata
5. Za svaku trku šampionata:
 - inicijalizuj atribut svih vozača (metoda prepareForTheRace klase Championship)
 - za svaki od krugova:
 - dodeli prosečno vreme vožnje svim vozačima (driveAverageLapTime)
 - primeni specijalne veštine svakog vozača (applySpecialSkills)
 - proveriti eventualne kvarove (checkMechanicalProblem)
 - ispiši vozača koji se nalazi na čelu nakon odvozanog kruga (printLeader)
 - prati eventualnu pojavu kiše i uticaj na vozače koji imaju, odnosno nemaju adekvatne pneumatike
 - na kraju trke ispiši pobednike trke (printWinnersAfterRace) koje ćeš utvrditi sortiranjem vozača na osnovu akumuliranog vremena vožnje

(koristiti `Collections.sort()` i implementirati interfejs `comparable/comparator` u klasi vozač – **na predavanjima će biti pokazano kako**)

- na kraju šampionata ispiši porednika (`printChampion`). Vozače takođe sortirati, ali ovaj put na osnovu osvojenih poena, a ne akumuliranog vremena u okviru jedne trke

Ocenjivanje (ukupno **25 poena**):

- Implementiran korisnički interfejs (odabir broja trka u šampionatu, odabir jedne po jedne staze na kojoj se održavaju trke, ispisivanje adekvatnih poruka o prispelim događajima, stanju na kraju kruga, porednicima trke i poredniku nakon završenog šampionata) - **5 poena**
- inicijalizacija kolekcija *drivers* i *venues* u klasi *Championship* iz datoteka *vozaci.txt* i *staze.txt* - **3 poena**
- postavljanje vozača na startne pozicije na osnovu renkinga, dodavanje vremenskih penala za sve vozače koji nisu na pol poziciji u skladu sa prvom tabelom od gore - **2 poena**
- simulacija jednog kruga trke za sve vozače (ažuriranje atributa `accumulatedTime`: prosečno vreme vožnje, primena specijalnih veština za svakog od vozača, eventualni kvarovi za svakog od vozača, kiša i uticaj na vreme vožnje kod vozača koji imaju/nemaju pneumatike, ispis vozača koji se nalazi na prvom mestu na kraju svakog kruga) - **10 poena**
- sortiranje vozača na kraju trke (`Collections.sort()`, ažuriranje `accumulatedPoints` atributa i ažuriranje renkinga na osnovu toga za svakog od vozača) - **5 poena**