

## Projektni zadatak

maj 2024.

ePJ2 je *e-mobility* kompanija koja se bavi iznajmljivanjem električnih automobila, bicikala i trotineta, na užem i širem prostoru grada Java. Cilj projektnog zadatka je da se napravi program koji će simulirati korišćenje prevoznih sredstava na osnovu predefinisanih podataka i generisati detaljne finansijske obračune, statistike i pratiti stanja svih prevoznih sredstava koji se koriste.

Kompanija ima na raspolaganju određen broj automobila za koje se čuvaju sljedeći podaci: jedinstveni identifikator (ID), datum nabavke, cijena nabavke, proizvođač, model, opis, trenutni nivo baterije. Za električne bicikle čuvaju se sljedeći podaci: jedinstveni identifikator (ID), proizvođač, model, cijena nabavke, trenutni nivo baterije, domet sa jednim punjenjem (autonomija). Za električne trotinete čuvaju se sljedeći podaci: jedinstveni identifikator (ID), proizvođač, model, trenutni nivo baterije, cijena nabavke i maksimalna brzina. Sva prevozna sredstva se mogu pokvariti, pri čemu se evidentira razlog kvara (opis) i datum i vrijeme. Sva prevozna sredstva imaju mogućnost punjenja baterije (metoda koja povećava trenutni nivo). Pražnjenje baterije se obavlja tokom kretanja. Automobil ima mogućnost prevoženja više ljudi.

Osnovni posao kompanije je iznajmljivanje prevoznih sredstava. Prilikom iznajmljivanja evidentiraju se datum i vrijeme iznajmljivanja, ime korisnika, trenutna lokacija gdje je prevozno sredstvo preuzeto, lokacija gdje se prevozno sredstvo ostavlja nakon korišćenja i trajanje korišćenja u sekundama. Prilikom iznajmljivanja automobila potrebno je dostaviti identifikacioni dokument (pasoš za strane državljane i ličnu kartu za domaće) i vozačku dozvolu (broj). Na osnovu ovih podataka generiše se račun za plaćanje koji se dostavlja korisnicima kao *txt* fajl.

Kompanija definiše cijene iznajmljivanja svakog tipa prevoznih sredstava po vremenu korišćenja u sekundama. Na osnovu trajanja vožnje i dodatnih faktora računa se ukupna cijena za plaćanje. Dodatni faktori koji utiču na cijenu su upotreba prevoznog sredstva u širem dijelu grada, kvarovi, popusti i promocije, a način obračuna dat je u tabeli. Svi parametri čuvaju se izvan programa u *properties* fajlovima (<https://www.baeldung.com/java-properties>).

Opis	Formula (velikim slovima su parametri koji se čuvaju u prop. fajlovima)
Udaljenost	Osnovna cijena * DISTANCE_NARROW za uži dio grada, osnovna cijena * DISTANCE_WIDE za širi dio grada.  Ako je bar jedno polje putanje u širem dijelu grada onda se primijenjuje tarifa za širi dio grada (DISTANCE_WIDE ).
Kvarovi	Osnovna cijena = 0, samo ako se kvar desio
Popust	Za svako 10. iznajmljivanje: Osnovna cijena * DISCOUNT (%)
Promocije	Osnovna cijena * DISCOUNT_PROM (%) za iznajmljivanje gdje je bila promocija
Osnovna cijena	Za automobile: CAR_UNIT_PRICE * sekunde trajanja vožnje

	Za bicikle: BIKE_UNIT_PRICE * sekunde trajanja vožnje Za trotinete: SCOOTER_UNIT_PRICE * sekunde trajanja vožnje
--	---

Primjer obračuna:

Iznos (osnovna cijena \* udaljenost) = ((SCOOTER\_UNIT\_PRICE \* Trajanje) \*  
DISTANCE\_WIDE)

Ukupno za plaćanje(Iznos - popust - promocija) = Iznos - (DISCOUNT \* Iznos) -  
(DISCOUNT\_PROM \* Iznos)

Sve stavke treba navesti pojedinačno na računima. Ako postoji kvar tada je Ukupno za plaćanje = 0.

Svi podaci o prevoznim sredstvima i iznajmljivanjima biće dostupni na Moodle stranici predmeta. Za svaku odbranu projektnih zadataka testni podaci mogu biti promijenjeni, tako da ništa ne smije biti definisano u kodu.

Podaci o iznajmljivanjima koriste se kako bi se simuliralo poslovanje kompanije. Iz fajla sa podacima učitava se jedan po jedan red sortirano po vremenu iznajmljivanja, i ti podaci se koriste da bi se u odvojenim nitima za svaki red prikazalo kretanje po mapi. Obavezno je da se simulacija obavlja po redu iznajmljivanja (datum i vrijeme). Jedna nit je vezana za samo jedno iznajmljivanje, a mapa je zajednička za sve niti. Simulacija se obavlja onoliko sekundi koliko je definisano trajanje iznajmljivanja. Kako su poznate početna i krajnja tačka iznajmljivanja, studenti treba sami da kreiraju direktnu putanju između njih i tako pomjere prikaz na mapi. Prikaz se može implementirati tako da se polja mape označe drugom bojom i da se ispiše identifikator prevoznog sredstva i nivo baterije. Kada se završi simulacija svih iznajmljivanja za određeni datum i vrijeme treba napraviti pauzu od 5s i onda krenuti simulacije iznajmljivanja za sljedeći datum i vrijeme. Kada se završi simuliranje jednog iznajmljivanja kreira se račun koji ima sve podatke vezane za to iznajmljivanje i ukupnu cijenu za plaćanje. Osim toga potrebno je navesti i stavke vezane za popuste ili promocije, ako postoje. Fajlovi se čuvaju u folderu na određenoj putanji (parametar u *properties* fajlu).

Program treba imati nekoliko grafičkih interfejsa za prikaz podataka koji se mogu implemetirati kao JavaFX ili Swing GUI:

- glavni ekran za prikaz mape,
- ekran za prikaz svih prevoznih sredstava (3 tabele, ispisati sve informacije),
- ekran za prikaz kvarova (1 tabela, kolone: vrsta prevoznog sredstva, ID, vrijeme, opis kvara) i
- ekran za prikaz rezultata poslovanja.

Mapa se sastoji od 20x20 polja. Polja označena bijelom bojom na slici (vanjska) su širi dio grada, a polja od označena plavom bojom su uži dio grada. Polje 0,0 je prvo lijevo gore, a polje 19,19 je kranje desno dole. Testni podaci su definisani na ovaj način. Dozvoljene putanje su isključivo pravolinijske, a trajanje simulacije se dijeli sa brojem polja koje će prevozno sredstvo preći po mapi i tako se definiše vrijeme zadržavanja na jednom polju. Pozicije prevoznog sredstva na mapi se prikazuju u realnom vremenu.

Rezultati poslovanja treba da prikažu sve prihode (iznose sa generisanih računa) i sve dodatne parametre bitne za poslovanje. Postoje 2 vrste izvještaja, sumarni i dnevni. Za sumarni izvještaj potrebno je ispisati sljedeće podatke:

[illegible]

Napomene:

- ovaj projektni zadatak vrijedi do objave novog projektnog zadatka,
- obavezno dodati JavaDoc komentare i generisati dokumentaciju,
- obavezno napraviti pakete,
- izbjegavati dupliranje koda i
- voditi računa o kvaliteti koda (nazivi klasa, metoda, promjenljivih, poravnanja, performanse...).