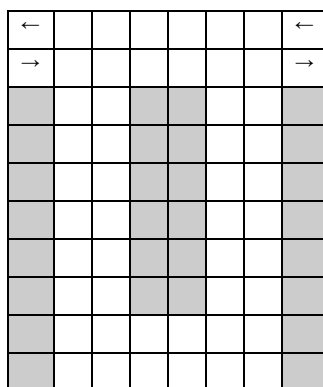


## Projektni zadatak maj 2018

Potrebno je napraviti aplikaciju **Garaža**. Aplikacija se sastoji iz dva dijela: korisničkog i administratorskog. Administratorski dio je GUI aplikacija kroz koju je administratoru omogućeno dodavanje i parkiranje proizvoljnog broja vozila određenog tipa u garažu, na osnovu kojih se formira trenutno stanje u garaži. Garaža može imati  $n$  platformi i svaka platforma ima sljedeći oblik:



Sa lijeve strane se ulazi i izlazi sa platforme garaže a sa desne prelazi na narednu platformu ili vraća sa iste (osim na poslednjem nivou). Za kretanje svakog od vozila važe stvarna pravila u saobraćaju. Vozila mogu da voze samo desnom stranom osim u slučaju ako naiđu na slobodno mjesto sa lijeve strane te mogu da pređu preko tog jednog polja. Ovo je dozvoljeno samo u slučaju ako se na tom polju ne nalazi vozilo iz suprotnog smijera. Važi i pravilo desne strane te vozilo ni u kom slučaju ne može preći na polje na kojem se već nalazi vozilo. Garaža za svaki nivo vodi i evidenciju o broju slobodnih mjesta. Ukoliko su svi nivoi popunjeni ne treba dozvoliti vozilima ulaz u garažu. Ukoliko je etaža popunjena vozila mogu samo da idu na naredni nivo tj. pravo, a samo u slučaju da na etaži postoji određeno mjesto vozilo skreće desno i pravi kružni prolazak po toj etaži prema narednom nivou.

Svako vozilo posjeduje naziv, broj šasije, broj motora, fotografiju i registarski broj. Tipovi vozila su motocikl, automobil (ima broj vrata) i kombi (ima nosivost). Garažu koriste i određene javne ustanove i institucije tako da kombi, motocikl i automobil mogu da budu i policijska vozila, kombi i automobil mogu da budu sanitetska vozila, a kombi i vatrogasno vozilo. Svako od ovih vozila može da ima upaljenu rotaciju. Ukoliko je upaljena rotacija ta vozila imaju prioritet te mogu da preteču ostala vozila pri čemu ostala vozila moraju da stanu. Pri tome saniteti imaju prioritet nad vatrogascima, a vatrogasci nad policijom. Takođe, policijsko vozilo sadrži i fajl koji predstavlja spisak registarskih brojeva vozila za kojima je raspisana potjera.

Prilikom pokretanja administratorskog dijela aplikacije prikazuje se početna forma koja sadrži (minimalno) padajući meni popunjen svim tipovima vozila, dugme za dodavanje novih vozila, tabelarni prikaz svih vozila koje su već dodana na određenu etažu i dugme za pokretanje

korisničkog dijela aplikacije. U slučaju da se aplikacija pokreće prvi put, tabela će biti prazna. Administrator vrši dodavanje vozila na etažu tako što na početnom prozoru bira iz padajućeg menija koje vozilo želi da doda, nakon čega mu se prikazuje forma sa odgovarajućim poljima za unos u zavisnosti od tipa samog vozila. Treba da postoji samo jedna forma za unos podataka o vozilu, pri čemu se polja za unos dinamički kreiraju i prikazuju u zavisnosti od tipa samog vozila. Za dodavanje fajlova obavezno je koristiti `FileDialog` ili ekvivalentnu JavaFX komponentu. Dodavanjem vozila automatski se osvježava tabela sa vozilima. Odabirom etaže u kombo boksu se prikazuje tabela za tu etažu. Potrebno je obezbijediti mogućnost izmjene vozila i brisanja vozila iz tabele. Administrator vrši pokretanje korisničke aplikacije klikom na odgovarajuće dugme. Prije samog prikaza korisničke aplikacije, garaža se serijalizuje u fajl *garaza.ser*, koji se deserijalizuje svaki put prilikom pokretanja administatorske aplikacije i koristi za popunjavanje tabele.

Korisnička aplikacija je GUI aplikacija koja predstavlja simulaciju kretanja vozila u garaži. Na korisničkoj aplikaciji, korisnik zadaje minimalan broj vozila koji se postavljaju na slučajan način u svaku od etaža. Tip vozila, kao i da li je od neke institucije, se slučajno generiše s tim da su u 90% slučajeva u pitanju obična vozila. Nakon svih unosa, korisnik pokreće simulaciju koja predstavlja kretanje vozila. Takmičenje započinje tako što se na matricu zadatih dimenzija na slučajne pozicije postavljaju prvo vozila iz fajla *garaza.ser*. U slučaju da je broj vozila u fajlu manji od minimalnog broja koje je korisnik zadao prilikom pokretanja simulacije, onda je dozvoljeno dodavanje novih vozila sa slučajno odabranim tipom. Nakon postavljanja svih učesnika, generiše se *text-area* za svaku od etaža (etaža se bira kombo boksom). Pri tome se prazna parking mjesta prikazuju sa \*, slovo **V** gdje se nalazi vozilo, slovo **H** gdje se nalazi vozilo hitne, slovo **P** gdje se nalazi vozilo policije, slovo **F** gdje se nalazi vozilo vatrogasaca. Ukoliko je upaljena rotacija na ispis se dodaje slovo **R**. Nakon toga, na istoj *text-area* nastavlja se prikaz simulacije. U tom trenutku se bira 15% vozila sa svake od etaža koja treba da napuste garažu. Korisnik tokom simulacije može da dodaje nova vozila u garažu sa svim parametrima ukoliko ista nije popunjena. Prilikom kretanja vozila u garaži i susretanja istih na raskrsnici ili tokom isparkiravanja u 10% slučajeva će se desiti udes. Ukoliko se to desi na mjesto udesa se šalju najbliža policijska patrola, hitna i vatrogasci pod rotacijom i na toj etaži se blokira saobraćaj dok se ne završi uviđaj za koji je odgovorna policija (od 3 do 10 sekundi). Na ostalim etažama saobraćaj normalno funkcioniše. Ukoliko policija detektuje vozilo za kojim je raspisana potjera isto mora da prati policijsko vozilo ka izlazu garaže. Uviđaj u ovom slučaju traje od 3 do 5 sekundi i na etaži je moguć saobraćaj za ostala vozila. U oba slučaja uviđaja policija evidentira vozila koja su učestvovala u istom, vrijeme kao i fotografije vozila te isto čuva u binarnoj datoteci.

Garaža vodi i evidenciju vozila po registarskom broju koja su ušla u istu i vrijeme kada su ušla. Prilikom izlaska vozila iz garaže vrši se naplata parkinga po sljedećem cjenovniku: 1 KM sat, zadržavanje do 3 sata 2 KM i 24 sata 8 KM. Naplata parkinga za policijska vozila, te vozila hitne pomoći i vatrogasaca se ne vrši.

Simulacija se završava u onom trenutku kada su slučajno odabrana vozila izašla iz garaže kao i ona koja je dodao klijent su se parkirala u garažu. Stanje garaže se nakon toga ponovo serijalizuje u fajl *garaza.ser*.

Evidenciju naplate parkinga moguće je preuzeti u obliku CSV fajla. Evidenciju uviđaja se čuva kao txt fajl u folderu *user.home*. Sve izuzetke koji se dese u aplikaciji evidentirati u fajl *error.log* upotrebom *Logger* klase.

Studenti koji predmet slušaju po starom planu i programu (šifra A401) treba da implementiraju sistem kao klijent server aplikaciju.