

Praktikum iz objektno-orijentisanog programiranja (13S112POOP) Projektni zadatak – C++ 2017/2018.

Napisati skup klasa sa odgovarajućim metodama, konstruktorima, operatorima i destruktorkama za realizaciju softverskog sistema za manipulaciju linijama gradskog prevoza. Potrebno je obezbediti osnovnu manipulaciju nad linijama, kao i generisanje grafovskih fajlova koji se mogu koristiti u alatima za vizuelizaciju. Podržani formati grafovskih fajlova treba da budu GML i CSV, a treba predvideti i mogućnost proširenja drugim formatima. Opis formata navedenih fajlova je dat u prilogu ovog dokumenta.

Korisnik (naručilac) softvera, želi da softver pruži sledeće funkcionalnosti:

- Interakciju sa korisnikom putem tekstualnog menija ili grafičkog korisničkog interfejsa
- Učitavanje podataka o mreži gradskog prevoza
 - učitavanje podataka o postojećim linijama
 - učitavanje podataka o stajalištima kroz koja data linija prolazi
- Osnovnu manipulaciju nad mrežom
 - prikaz podataka o linijama i stajalištima
 - izmena i brisanje podataka o linijama i stajalištima
 - filtriranje linija
 - određivanje parametara mreže
- Generisanje grafovskih fajlova
 - generisanje grafovskih fajlova po C i L-prostor modelu mreže gradskog saobraćaja
 - formatiranje fajlova po GML ili CSV formatu
- Kraj rada

Za uspešno rešenje zadatka potrebno je izvršiti analizu zahteva. Kao rezultat analize, potrebno je dopuniti i precizirati funkcionalnu specifikaciju softverskog alata. Na osnovu specifikacije, potrebno je napisati sistem klasa u jeziku C++ koje realizuju traženi softver. U nastavku su navedeni neki elementi specifikacije. Od studenata se očekuje da dopune one stavke koje nisu dovoljno precizno formulisane, odnosno dodaju nove stavke (tamo gde to ima smisla) ukoliko uoče prostor za unapređenje. Izmene i dopune specifikacije mogu da donekle odudaraju od zahteva naručioca softvera u onoj meri u kojoj to neće narušiti traženu funkcionalnost. Prilikom izrade specifikacije voditi računa o potencijalnom unapređenju softvera na osnovu naknadnih zahteva.

Prilikom izrade rešenja, od studenata se očekuje intenzivno korišćenje svih onih mogućnosti koje pružaju specifikacija jezika C++ i biblioteke STL i Boost, kao što su šablonske funkcije, kolekcije, algoritmi, regularni izrazi, iteratori, lambda izrazi i sl. **Rešenja koja ne vode računa o ovom aspektu neće moći da dobiju maksimalan broj poena.** Takođe, voditi računa o objektno orijentisanom dizajnu rešenja, čistoći, čitkosti i komentaranju programskog koda.

Funkcionalna specifikacija

U nastavku je zadat deo korisničkih zahteva koje treba razraditi i, po potrebi, dopuniti tako da se dobije funkcionalna aplikacija.

Interakcija sa korisnikom

Korisnik može da interaguje sa programom bilo izborom odgovarajućih opcija iz tekstualnog menija putem tastature ili izborom u datom trenutku dostupnih opcija putem grafičkog korisničkog interfejsa. Nije potrebno realizovati oba načina. U oba slučaja obezbediti da se interakcija sa korisnikom odvija putem statičkih ekrana. Interakcija u slučaju grafičkog interfejsa može da se vrši putem tastature ili miša. U zavisnosti od izabrane opcije i njenih parametara, program izvršava zadatak opciju ili ispisuje poruku greške. Poruka greške treba da bude što je moguće detaljnija da bi korisniku pomogla da grešku otkloni. Sve eventualne parametre koji su potrebni prilikom rada aplikacije je potrebno zatražiti od korisnika. Ukoliko korisnik ne zada ništa, koristiti vrednosti fiksirane u programu.

Učitavanje podataka o mreži gradskog prevoza

Potrebno je obezbediti učitavanje podataka o linijama i stajalištima iz tekstualnih fajlova i to u grupnom i pojedinačnom režimu rada. Grupni režim rada podrazumeva čitanje spiska svih postojećih linija i stajališta kroz koja svaka od učitanih linija prolazi, dok pojedinačni režim rada podrazumeva učitavanje podataka o samo jednoj liniji i njenim stajalištima. Spisak postojećih linija se učitava iz zadanog fajla u kome su u svakom redu zapisane informacije o jednoj liniji. Svaki red sadrži oznaku linije (tekstualnog tipa zbog postojanja linija EKO1, 25P, 27L i drugih) i naziv početnog i krajnjeg stajališta (okretnice). Sve informacije su odvojene jednim znakom - separatorom, koji se zadaje prilikom učitavanja, po sledećem format (u primerima je korišćen znak uzvika ! kao separator):

7!Ustanicka!Blok 45!

Za svaku liniju potrebno je učitati i stajališta kroz koja ona prolazi. Učitavanje stajališta se vrši za svaku liniju pojedinačno i to iz dva fajla – jedan fajl sadrži spisak stajališta kroz koja linija prolazi u smeru A, dok drugi sadrži spisak stajališta kroz koja linija prolazi u smeru B. Naziv fajla se sastoji od oznake linije i smera na koji se odnosi po sledećem formatu:

7_dirA.txt 7_dirB.txt

U svakom redu fajla koji sadrži spisak stajališta nalaze se informacije o po jednom stajalištu, i to celobrojna šifra, naziv, geografska lokacija (geografska širina i dužina odvojene separatorom) i zona (može biti u opsegu 1-4), odvojene separatorom po sledećem formatu:

64!Vukov spomenik!44.8056059!20.4783626!1

Informacije o stajalištima su u fajlovima navedene onim redosledom kojim linija i prolazi kroz njih, tj. u svakom fajlu je kao prvo stajalište navedeno početno stajalište date linije u datom smeru, a zatim i ostala stajališta po redosledu kretanja linije.

Prilikom učitavanja podataka o mreži koristiti regularne izraze za parsiranje datih fajlova. Predvideti način za oporavak od grešaka u slučaju neispravnog formata datoteke ili nepostojanja tražene datoteke.

Osnovna manipulacija nad mrežom

Osnovna manipulacija nad mrežom treba da omogući prikaz osnovnih podataka o linijama i stajalištima. Osnovni podaci o liniji uključuju njenu oznaku, početno i krajnje stajalište, kao i broj i spisak stajališta kroz koja prolazi, dok osnovni podaci o stajalištu uključuju šifru, naziv, geografsku lokaciju, zonu, kao i spisak svih linija koje prolaze kroz dato stajalište.

Omogućiti izmenu podataka o liniji, kao i brisanje date linije iz mreže. Izmena podataka može uključiti izmenu oznake linije ili brisanje i dodavanje novog stajališta u spisak stajališta date linije za dati smer.

Korisniku je potrebno dozvoliti i da filtrira spisak linija i to po sledećim kriterijumima:

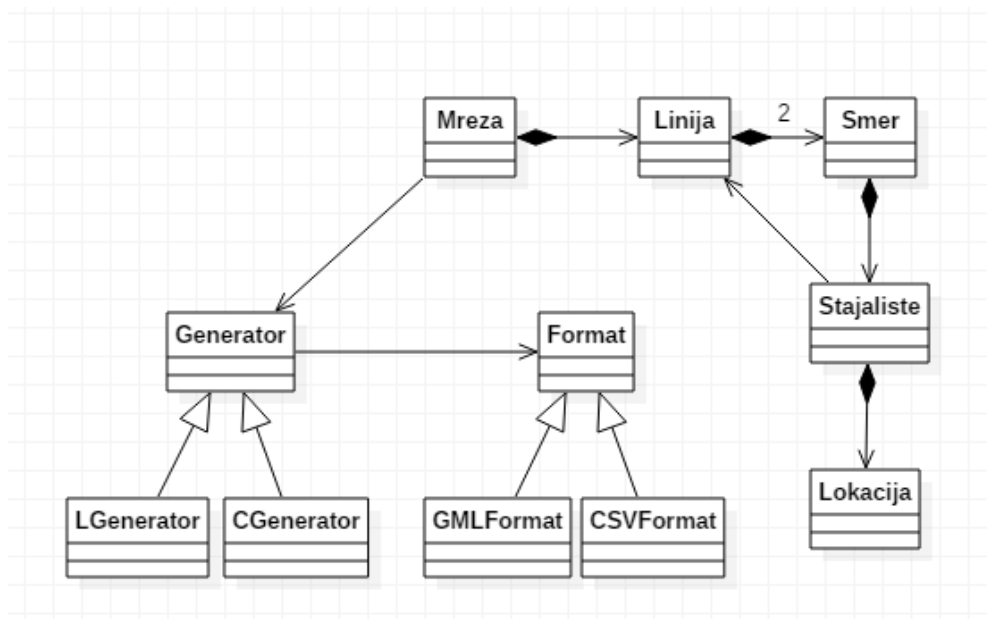
- filtriranje na osnovu zona kroz koje linija prolazi (u skupu linija ostaju linije koje prolaze samo kroz 1. zonu, samo kroz 1. i 2. zonu ili samo kroz 1, 2. i 3. zonu).
- filtriranje linija na osnovu oznake (u skupu linija ostaju linije koje u oznaci imaju broj manji od zadatog broja, veći od zadatog broja ili u zadatom opsegu, pri čemu se sva slova u sufiksu oznake linije zanemaruju – broj linije 25P je 25)
- filtriranje linija na osnovu broja stajališta kroz koja prolaze (u skupu ostaju linije koje prolaze kroz veći ili manji broj stajališta od zadatog broja)

Pored osnovnih manipulacija potrebno je obezbediti i sledeće:

- određivanje skupa linija sa kojima data linija ima zajednička stajališta (bez obzira na smer kretanja)
- određivanje da li data linija prolazi kroz zadata dva stajališta u istom smeru svog kretanja
- određivanje linije sa kojom data linija ima najviše zajedničkih stajališta
- određivanje najbližeg stajališta u odnosu na zadata geografsku lokaciju, uz mogućnost određivanja najbližeg stajališta samo određenje linije
- određivanje broja zajedničkih stajališta za sve parove linija koje imaju zajedničko stajalište, uz mogućnost filtriranja na parove linija koje imaju zajedničkih stajališta više od zadatog broja
- određivanje svih linija koje prolaze kroz dato stajalište
- određivanje svih stajališta do kojih je moguće stići iz zadatog stajališta uz vožnju maksimalno jednu stanicu
- određivanje najmanjeg potrebnog broja presedanja na putu između dva zadata stajališta
- određivanje najkraćeg puta između dva stajališta (ne uzimati u obzir geografsku lokaciju, već za najkraći put uzeti onaj koji se sastoji od najmanjeg broja stajališta)

Dijagram klasa

Na osnovu prethodne funkcionalne specifikacije formiran je sledeći dijagram klasa. Dijagram klasa nije detaljan, te ga treba tumačiti kao skicu koja načelno ukazuje na arhitekturu softvera. Studenti mogu da koriste ovaj dijagram kao referencu i, po potrebi, prošire ga da bi ga usaglasili sa eventualnim dopunama specifikacije. Studenti imaju slobodu da arhitekturu sistema isprojektuju i na drugačiji način, poštujući principe objektno-orijentisanog programiranja i dizajna.



Prilikom implementacije rešenja, obratiti pažnju na objektno orijentisani dizajn i intenzivno koristiti kolekcije i algoritme standardne biblioteke jezika C++ i lambda funkcije gde god je to moguće. Primiti da postoje dva različita modela generisanog grafa sa dva različita formata izlaznih fajlova.

Specifikacija GML formata

GML (Graph Modeling Language) je tekstualni format fajla za predstavljanje grafova pomoću veoma jednostavne sintakse. Koriste ga alati Gephi, Graphlet, Pajek, zEd, LEDA i NetworkX.

Specifikacija grafa se sastoji od navođenja čvorova i grana grafa po format koji je prikazan u primeru, uz mogućnost zadavanja i labele za čvor ili granu.

Primeri fajlova u ovom formatu:

```
graph
[
  node
  [
    id A
  ]
  node
  [
    id B
  ]
  node
  [
    id C
  ]
  edge
  [
    source B
    target A
  ]
  edge
  [
    source C
    target A
  ]
]
```

```
graph
[
  node
  [
    id A
    label "Node A"
  ]
  node
  [
    id B
    label "Node B"
  ]
  node
  [
    id C
    label "Node C"
  ]
  edge
  [
    source B
    target A
    label "Edge B to A"
  ]
  edge
  [
    source C
    target A
    label "Edge C to A"
  ]
]
```

Specifikacija CSV formata

Alat Gephi podržava i CSV format pomoću koga je lako predstaviti grafove. CSV format podržavaju i baze podataka, kao i Microsoft Excel. Svaka linija kod ovog formata mora sadržati bar dva elementa odvojena specijalnim znakom – separatorom, za koji se uglavnom koristi zarez. Vrednosti se mogu pisati pod navodnicima u slučaju kada sadrže specijalni znak. Predstavljanje grafova pomoću ovog formata moguće je realizovati na više načina.

Prvi način je lista grana i kod ovog načina se u svakom redu zadaju dva identifikatora čvora, koji su povezani granom grafa, odvojena separatorom. U sledećem primeru predstavljen je graf koji ima grane između čvorova a-b i čvorova b-c.

```
a;b  
b;c
```

Drugi način za reprezentaciju grafa pomoću ovog formata jeste lista susednosti. U svakom redu navodi se identifikator čvora grafa i svih ostalih čvorova sa kojima je on povezan nekom granom grafa. U navedenom primeru predstavljen je graf koji ima grane između čvorova a-b, b-c i b-d.

```
a;b  
b;c;d
```

Treći način za predstavljanje grafa ovim formatom jeste kvadratna matrica u kojoj se pomoću jedinice označava postojanje grane između dva čvora, kao što je i prikazano u sledećem primeru:

```
;A;B;C;D;E  
A;0;1;0;1;0  
B;1;0;0;0;0  
C;0;0;1;0;0  
D;0;1;0;1;0  
E;0;0;0;0;0
```