Praktikum iz objektno orijentisanog programiranja (13S112POOP)
Projektni zadatak – Java

Napisati skup klasa sa odgovarajućim metodama i konstruktorima za realizaciju softverskog sistema za rad sa bazama podataka. Implementacija projektnog zadatka treba da se oslanja na rešenje realizovano na programskom jeziku C++ u sklopu prvog projektnog zadatka.

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

- Interakciju sa korisnikom putem grafičkog korisničkog interfejsa
- Kreiranje baze podataka uz mogućnost učitavanja postojeće baze podataka
- Osnovna manipulacija
    - Kreiranje, brisanje i prikazivanje tabela
    - Izvršavanje upita i prikazivanje rezultata
- Eksportovanje baze podataka
- 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 Java 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. Takođe, priloženi UML dijagram koji opisuje zahtevani softver se ne mora obavezno poštovati, već samo predstavlja skicu potencijalnog rešenja. 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 i biblioteke jezika Java, kao što su 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 komentarisanju 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 izborom u datom trenutku dostupnih opcija putem **JavaFX grafičkog korisničkog interfejsa**. Interakcija može da se vrši putem tastature ili miša. U zavisnosti od izabrane opcije i njenih parametara, program izvršava zadatu 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.

Pokretanje programa

Prilikom pokretanja programa korisniku se nudi da kreira novu bazu podataka ili da učita postojeću. Potrebno je definisati format koji bi podržao čuvanje baze podataka. **Format treba da odgovara formatu koji je korišćen u prvom projektnom zadatku.** Iz toga sledi da je tabele napravljene u prvom projektnom zadatku moguće otvoriti pomoću alata koji je predmet drugog projektnog zadatka i obrnutno. Pored toga, potrebno je obezbediti učitavanje baze podataka i iz SQL fajla u kome se nalaze komande za kreiranje i popunjavanje tabela (pogledati prvi projektni zadatak).

Po učitavanju postojeće ili kreiranju nove baze podataka korisnički interfejs za vreme izvršavanja programa prikazuje:

- spisak postojećih tabela
- podatke iz trenutno selektovane tabele
- tekstualno polje za unos upita i prostor u kome se prikazuje rezultat izvršavanja upita

Podatke iz selektovane tabele i rezultate izvršavanja upita potrebno je prikazati u vidu tabele. U prvom redu se prikazuju nazivi kolona, dok se u narednim redovima prikazuju sami podaci. Određivanje načina za prikaz tabele se ostavlja studentima pri čemu je potrebno voditi računa o lakoći korišćenja alata, a posebno o preglednosti.

Osnovna manipulacija

Korisniku treba dozvoliti sledeće opcije:

1. Opcije koje je potrebno implementirati isključivo na jeziku Java:
    - Brisanje postojeće tabele
    - Prikaz podataka iz postojeće tabele
2. Opcije za koje je potrebno koristiti rešenje iz prvog projektnog zadatka
    - Izvršavanje upita

Za izvršavanje upita potrebno je koristiti rešenje sa prvog projektnog zadatka. Koristeći JNI treba pozvati funkciju koja je deo C++ projekta, a koja izvršava zadati upit. Funkciji je neophodno proslediti upit i informaciju nad kojom bazom podataka da izvrši upit. Kao rezultat izvršavanja upita moguće je dobiti grešku, broj zapisa u bazi podataka na koje je ovaj upit delovao (broj kreiranih, ažuriranih ili obrisanih zapisa) ili podatke koji su selektovani (u slučaju SELECT naredbe). Format u kome će podaci biti vraćeni studenti sami treba da specificiraju.

Čuvanje podataka i kraj rada

Korisnik može da zahteva da se baza podataka sačuva pri čemu je potrebno da zada format i putanju destinacionog fajla. Potrebno je podržati čuvanje baze podataka u oba formata iz kojih je bazu moguće i učitati.

Korisnik može da zahteva kraj rada programa. Od korisnika se traži potvrda za napuštanje programa. Ukoliko korisnik nije sačuvao bazu podataka koja je trenutno otvorena potrebno mu je ponuditi da to uradi pre napuštanja programa.

Generisanje dokumentacije

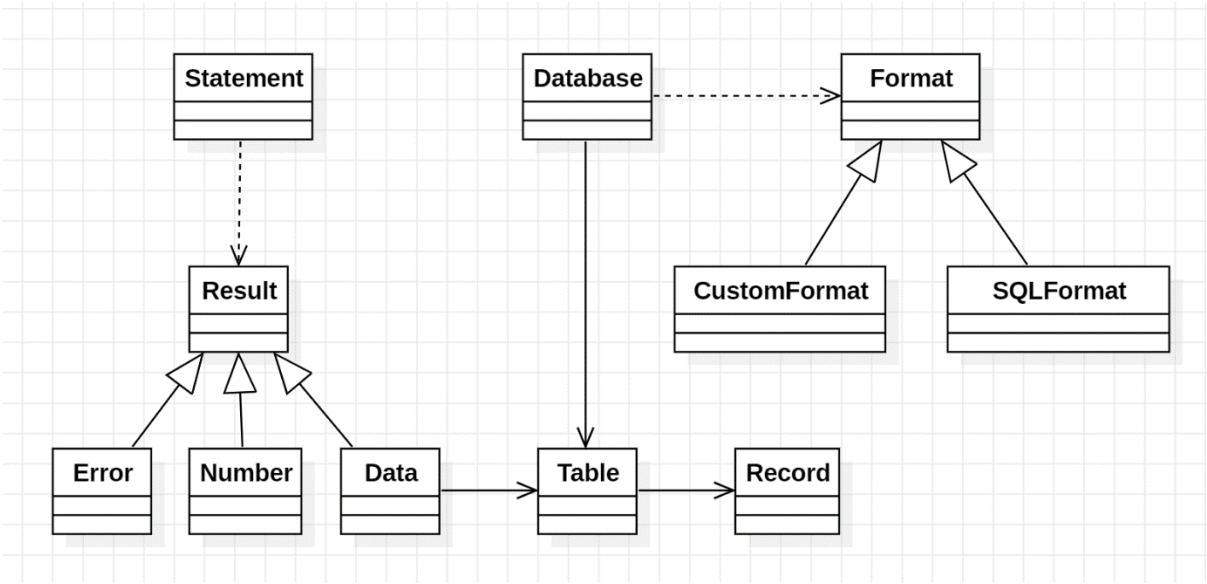
Potrebno je jednu klasu (koja ima veći broj metoda) detaljno opremiti dokumentacionim komentarima I genrisati dokumentaciju za ovu klasu korišćenjem alata Javadoc.

Testiranje rada programa

Prilikom testiranja rada programa moguće je koristiti postojeće alate za rad sa bazama podataka. Baza podataka napravljena u ovom alatu se može sačuvati u .sql fajlu. Naredbe iz ovog fajla je moguće izvršiti u većiti alata za rad sa bazama podataka i time se može proveriti ispravnost rada.

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.



Prilikom implementacije rešenja, obratiti pažnju na objektno orijentisani dizajn i intenzivno koristiti kolekcije i algoritme standardne biblioteke jezika Java i lambda funkcije gde god je to moguće.

Napomene

Studenti koji nisu radili prvi projektni zadatak celokupan projekat treba da implementiraju na programskom jeziku Java.

Na broj osvojenih poena utiče i:

- korišćenje mogućnosti koje pruža biblioteka jezika Java (kolekcije, algoritmi, regularni izrazi, iteratori, lambda izrazi)
- objektno orijentisani dizajn rešenja,
- čistoća, čitkost i komentarisanje programskog koda