

# 1 Tema

Napisati program za upravljanje radom banke.

## 2 Tehnički zahtjevi

1. Omogućiti operacije unosa, izmjene, pregleda i brisanja: korisnika, radnika, tekućih i štednih računa, kredita i transakcija.
2. Unos novih stavki, te izmjenu postojećih vršiti sa tastature.
3. Željenu radnju (unos, izmjena, pregled, brisanje) korisnik unosi preko tastature na osnovu ponuđenih opcija u terminalu.
4. Za novi unos definirati potrebne kolone i ograničiti unos ukoliko se ne unesu potrebni podaci.
5. Za povezivanje tabela koristit će se ključevi, koji predstavljaju jedinstveni identifikator određene stavke, kao npr:
  - radnik/korisnik: tekstualni zapis JMBG od 13 cifara,
  - tekući/štedni račun: tekstualni zapis od 16 cifara;

## 3 Formalni zahtjevi

1. Programski kôd (*backend*) pisati u C++ programskom jeziku.
2. Koristiti što više elemenata C++17 standarda.
3. Programsko sučelje (*frontend*) bit će terminal (*Command Prompt* ili *UNIX terminal*).
4. Strogo se pridržavati principa objektno – orijentiranog programiranja.
5. Jasno definirati potrebne klase.
6. Koristiti naslijeđivanje gdje je potrebno.
7. Maksimalno ograničiti vidljivost i pristup članovima klase.
8. Implementirati sve konstruktore i destruktore.
9. Napraviti odgovarajuće verzije konstruktora ovisno o očekivanom unosu.
10. Koristiti STL kontejnere, a gdje to nije moguće koristiti dinamički alociranu memoriju.
11. Smanjiti ponavljanje kôda kroz implementaciju funkcija.
12. Gdje je moguće koristiti metode umjesto funkcija.
13. Deklaracije izdvojiti u .h datoteke, definicije u .cpp datoteke, a napraviti i main.cpp datoteku koja predstavlja glavni program.
14. Paziti na zauzimanje resursa radne memorije.
15. Objekte, klase, članove klase i njene metode imenovati razumljivo.
16. Nastojati na jedinstven način imenovati metode koji obavljaju iste funkcije u više klasa.