Grupa 6 (Bankari) 

Analiza sistema

# Uvod

Sistem za banku koji omogućava:

* korisnicima banke da urade stvari za koje bi inače morali otići u poslovnicu, kao što je vršenje uplata na drugi račun
* bankarima da vode evidenciju, upisuju nove klijente, potvrđuju zahtjeve za kredite i slično
* administratoru da nadzire čitav spektar radnji vezanih za banku, kao i da dodaje novosti, uređuje objekte na mapi (bankomate i filijale)

# Osnovne Klase (Modeli)

1. Korisnik – apstraktna klasa koja opisuje korisnika koji se može logovati na sistem

* Atributi:

-id (Int)

-ime (String)

-prezime (String)

-korisnickoIme (String)

-lozinka (String)

* Metode

-getteri i setteri

1. Adresa – klasa koja opisuje mjesto na mapi

* Atributi:

-id (Int)

-latitude (Float)

-longitude (Float)

-naziv (String)

* Metode

-konstruktor

-getteri i setteri

1. Klijent extends Korisnik implements IKlijentPodnosilacZahtjevaZaKredit, IKlijentKD, IKlijentUplatitelj – klasa koja opisuje klijenta banke

* Atributi:

-datumRodjenja (Date)

-JMBG (String)

-brojTelefona (String)

-brojLicneKarte (String)

-adresa (Adresa)

-zanimanje (String)

-racuni (List<Racun>)

* Metode

-konstruktor

-getteri i setteri

1. Racun – opisuje bankovni račun

* Atributi:

-id (Int)

-stanjeRacuna (Float)

-vrstaRacuna (VrstaRacuna)

* Metode

-konstruktor

-getteri i setteri

1. Transakcija

* Atributi:

-id (Int)

-vrijeme (DateTime)

-saRacuna (Racun)

-naRacun (Racun)

-iznos (Float)

-statusTransakcije (StatusTransakcije)

-vrstaTransakcije (VrstaTransakcije)

* Metode

-konstruktor

-getteri i setteri

1. KreditBaza – apstraktna klasa koja opisuje zahtjev za kredit ili kredit koji je u toku ili završen

* Atributi:

-id (Int)

-klijent (Klijent)

-racun (Racun)

-iznos (Float)

-kamatnaStopa (Float)

-rokOtplate (Date)

* Metode

-getteri i setteri

1. ZahtjevZaKredit extends KreditBaza

* Atributi:

-namjenaKredita (String)

-mjesecniPrihodi (Float)

-prosjecniTroskoviDomacinstva (Float)

-nazivRadnogMjesta (String)

-nazivPoslodavca (String)

-radniStaz (Int)

-brojNekretnina (Int)

-iznosNeplacenihDugova (Float)

* Metode

-konstruktor

-getteri i setter

-izracunajKreditnuSposobnost(): Int

1. StatusKredita (enum)

-Aktivan

-Zavrsen

1. VrstaTransakcije (enum)

-UobicajenoPlacanje

-IndividualnoPlacanje

-Kupnja

-IndividualniDohodak

-UobicajeniDohodak

1. StatusTransakcije (enum)

-UTijeku

-Obradjen

1. VrstaRacuna (enum)

-Tekuci

-Ziro

-Devizni

-Stedni

1. Kredit extends KreditBaza

* Atributi:

-isplaceno (Float)

-podignut (Date)

-status (StatusKredita)

* Metode

-konstruktor

-getteri i setteri

1. Bankar extends Korisnik implements IBankarKreditar, IBankarRacunovodja, IBankarUpravljacKlijenata

* Atributi:

- mjestoZaposlenja (Filijala)

* Metode

-konstruktor

-getteri i setteri

1. Administrator extends Korisnik implements IAdministratorUpravljacNovostima, IAdministratorUpravljacBankarima, IAdministratorUpravljacBankomatima, IAdministratorUpravljacFilijalama

* Metode

-konstruktor

-getteri i setteri

1. Filijala – klasa koja opisuje jednu poslovnicu

* Atributi:

- id (Int)

- ime (String)

- adresa (Adresa)

- brojTelefona (String)

* Metode

-konstruktor

-getteri i setteri

1. Bankomat

* Atributi:

- id (Int)

- adresa (Adresa)

- trenutnoStanje (Float)

* Metode

-konstruktor

-getteri i setteri

1. Novost – klasa koja predstavlja jednu vijest na oglasnoj ploči

* Atributi:

- id (Int)

- dodano (DateTime)

- naslov (String)

- sadrzaj (String)

- prikazana (Bool)

* Metode

-konstruktor

-getteri i setteri

# Interface

1. IAdministratorUpravljacNovostima

* Metode

-dodajNovost(novost: Novost)

-urediNovost(novost: Novost)

-ukloniNovost(novost: Novost)

1. IAdministratorUpravljacBankarima

* Metode

-dodajBankara(bankar: Bankar)

-urediBankara(bankar: Bankar)

-ukloniBankara(bankar: Bankar)

1. IAdministratorUpravljacBankomatima

* Metode

-dodajBankomat(bankomat: Bankomat)

-urediBankomat(bankomat: Bankomat)

-ukloniBankomat(bankomat: Bankomat)

1. IAdministratorUpravljacFilijalama

* Metode

-dodajFilijalu(filijala: Filijala)

-urediFilijalu(filijala: Filijala)

-ukloniFilijalu(filijala: Filijala)

1. IBankarKreditar

* Metode

-procesuirajZahtjevZaKredit(zahtjevZaKredit: ZahtjevZaKredit, odobren: Bool)

1. IBankarRacunovodja

* Metode

-otvoriRacun(racun: Racun)

-zatvoriRacun(racun: Racun)

1. IBankarUpravljacKlijenata

* Metode

-dodajKlijenta(klijent: Klijent)

-urediKlijenta(klijent: Klijent)

-ukloniKlijenta(klijent: Klijent)

1. IKlijentPodnosilacZahtjevaZaKredit

* Metode

-podnesiZahtjevZaKredit(zahtjevZaKredit: ZahtjevZaKredit)

1. IKlijentKD

* Metode

-konvertujDevizu(iznos: Float, izValute: String, uValutu: String): Float

1. IKlijentUplatitelj

* Metode

-uplati(transakcija: Transakcija)

# Baza podataka

Biti će neophodna baza podataka koja će čuvati sve tabele slične klasama iznad.

# APi

API će se koristiti za dobijanje informacija o trenutnom stanju deviza pri konvertovanju istih.