

SVEUČILIŠTE U ZAGREBU  
PRIRODOSLOVNO MATEMATIČKI FAKULTET

BANKOMAT  
DRUGA ZADAĆA IZ KOLEGIJA SOFTVERSKO INŽENJERSTVO

MIRJANA JUKIĆ-BRAČULJ

Prof. dr. sc. ROBERT MANGER

Zagreb, 2.2.2017.

## **1. Predgovor**

Dokument predstavlja rješenje druge domaće zadaće iz kolegija Softversko inženjerstvo na drugoj godini Diplomskog studija Računarstvo i matematika. Cilj zadaće je napraviti oblikovanje modela Bankomata prikazanog u prvoj zadaći.

## **2. Uvod**

Oblikovanje modela Bankomata provedeno je profinjavanjem modela strukture (class dijagrama) ostvarenim dodavanjem tipova atributa, povratnih vrijednosti i argumenata metoda. Profinjeni su i sequence dijagrami opisani u detaljnim specifikacijama use case-ova u prvoj zadaći. Osim toga, sustav je podjeljen na komponente, a prikaz se vidi na pripadnom component dijagramu.

### 3. Model strukture

Klase u modelu bankomata raspoređene su u tri komponente s pripadnim traženim i ponuđenim sučeljima. Komponente su *GrafickoSucelje*, *SredišnjiSustav* i *KorisnickiDetalji*.

U nastavku sljedi detaljan opis klasa dobivenih profinjavanjem asocijacija između klasa i dodavanjem tipova atributa, povratnih vrijednosti te argumenata metoda u klasama.

Dobivene su veze kompozicije i agregacije koje se implementiraju tako da klasa “cjelina” sadrži referencu na klasu “dio”.

Komponenta *GrafickoSucelje* obuhvaća klase zadužene za komunikaciju s korisnikom. Tu spadaju klase: *DozerNovca*, *Ekran*, *Printer*, *CitacKartice*.

<b>Ime klase:</b> DozerNovca	
<b>Opis klase:</b> Klasa je zadužena za isplatu i prihvrat novca od korisnika.	
<b>Atributi:</b>	
private int stanjeSpremnika	Privatni argument koji pamti količinu novca u spremniku bankomata. Argument ima pripadna get i set svojstva.
<b>Metode:</b>	
public DozerNovca()	Javni konstruktor koji ne prima niti jedan parametar.
public void isplatiNovac(int kolicinaNovca)	Javna metoda koja korisniku isplaćuje količnu novca dobivenu u parametru količina novca. Poziva se kada banka odobri i obavi transakciju isplate novca na korisnikovom računu.
public void prihvatiNovac()	Javna metoda kojom se DozerNovca prebacuje u stanje prihvata novca. Poziva se ako korisnik odabere transakciju UplataNovca.
public int prebrojiNovac()	Javna metoda koja se poziva kada je korisnik odabrao prekid uplate novca.

<b>Ime klase:</b> Ekran	
<b>Opis klase:</b> klasa omogućava komunikaciju s korisnikom ispisujući mu poruke i mogućnosti odabira usluga koje pruža bankomat.	
<b>Metode:</b>	
public Ekran()	Javni konstruktor bez parametara.
public int traziPIN()	Javna metoda koja se poziva kada korisnik ubaci karticu. Crta se ekran o upitu PIN-a. Funkcija vraća unsemi PIN.
public int ponudiIzborTransakcije()	Javna metoda koja se poziva ako je provjera PIN-a uspješno obavljena. Crta se ekran s listom usluga koje banomat nudi: uplata novca, isplata novca, promjena PIN-a i provjera stanja na računu. Vraća cjelobroju vrijednost koja označava tip transakcije.
public boolean pitajZaPotvrdu()	Javna metoda koja se poziva prije obavljanja transakcije. Korisnika se pita želi li potvrdu o obavljanoj transakciji. Vraća <i>true</i> ako je odgovor da, inače vraća <i>false</i> .
public void informirajOUspjehu()	Javna metoda koja na ekran ispisuje poruku o uspjehu obavljanja transakcije.
public void informirajONeuspjehu()	Javna metoda koja na ekran ispisuje poruku o neuspjehu obavljanja transakcije.
public int traziIznosTransakcije()	Javna metoda koja crta ekran s upitom o iznosu količine novca koja će korisniku biti isplaćena prilikom obavljanja transakcije. Vraća unesenu vrijednost količine novca za isplatu.
public void traziUplatuNovca()	Javna metoda koja crta na ekran obavijest da je bankomat spreman za prihvrat novca od strane korisnika.
public int pitajPromjenuPINa()	Javna metoda koja prikazuje na ekranu upit o unosu novog PIN-a. Vraća unesenu vrijednost za PIN.
public void unosPINa(int PIN)	Javna metoda koja se poziva prilikom korisnikova unosa novog PIN-a.
public void odabirTransakcije(int odabir)	Javna metoda koja se poziva prilikom korisnikova odabira transakcije.
public void unosIznosaTransakcije(int iznos)	Javna metoda koja se poziva prilikom korisnikova unosa vrijednosti količine novca za isplatu.
public void odabirPotvrde(boolean odabir)	Javna metoda koja se poziva prilikom korisnikova odgovora na upit o ispisivanju potvrde.
public void krajUplateNovca()	Javna metoda koja se poziva kada korisnik označi kraj uplate novca.

<b>Ime klase:</b> Printer	
<b>Opis klase:</b> Klasa sadrži argumente koji čuvaju opće informacije o bankomatu koje se ispisuju na svakoj potvrdi i metode za ispis potvrde o transakciji.	
<b>Atributi:</b>	
private string nazivBanke	Privatni argument koji čuva ime banke kojoj bankomat pripada. Ima pripadna get i set svojstva.
private string adresa	Privatni argument koji čuva adresu bankomata. Ima pripadna get i set svojstva.
private string porukaZahvaleIInfo	Privatni argument koji čuva poruku zahvale za korištenje bankomata i kontakt na koji se korisnici mogu obratiti ako trebaju dodatnu informaciju. Ima pripadna get i set svojstva.
private int stanjeSpremnikaPapir	Privatni argument koji označava broj potvrda koji se može isprintati s trenutnom količinom papira u spremniku.
<b>Metode:</b>	
public Printer()	Javni konstruktor bez parametara.
public boolean ispisiPotvrdu(Transakcija transakcija, Racun racun)	Javna metoda koja ispisuje potvrdu o transakciji. Na potvrdi se treba nalaziti ime banke, datum, vrijeme izdavanja potvrde, adresa bankomata, dopušteni dio broja kartice, broj računa, iznos transakcije, stanje na računu i poruka sa zahvalom i dodatnim informacijama.

<b>Ime klase:</b> CitacKartice	
<b>Opis klase:</b> Klasa je zadužena za čitanje podataka s korisnikove kartice.	
<b>Atributi:</b>	
private Kartica kartica	Privatna varijabla koja čuva podatke pročitane s korisnikove kartice. Ima pripadna get i set svojstva.
<b>Metode:</b>	
public CitacKartice()	Javni konstruktor bez parametara.
public void izbaciKarticu()	Javna metoda kojom se korisniku vraća kartica po završetku uspješnog obavljanja transakcije.
public void prihvatiKarticu()	Javna metoda koja se pokreće kada korisnik ubaci karticu u bankomat. Tada se čitaju podaci s kartice i konstruira se instanca klase Kartica koja se pridružuje atributu klase kartica.

Komponenta *SredisniSustav* obuhvaća klase zadužene za obavljanje glavnih funkcionalnosti rada bankomata. To su: *Bankomat*, *BankaDB*, *Dezurstvo* i *Administrator*.

<b>Ime klase:</b> Bankomat	
<b>Opis klase:</b> Glavna klasa koja upravlja cijelim sustavom. Uz argumente navedene u tablici čuva i reference na klase iz komponente <i>GafickoSucelje</i> i preko njih ostvaruje komunikaciju s korisnikom. Pomoću reference klase <i>BankaDB</i> provodi transakcije na najnižem sloju – bazi podataka. Za slanje podataka o transakciji i korisnikovu računu koristi strogo propisana pravila o kriptiranju podataka. Za svaku obavljanu transakciju zapisuje log o detaljima transakcije. Sadrži metode za obavljanje komunikacije s administratorima.	
<b>Argumenti:</b>	
private int id	Privatni argument koji predstavlja identifikacijski broj bankomata. Ima pripadna get i set svojstva.
private string adresa	Privatni argument, čuva adresu bankomata. Ima pripadna get i set svojstva.
<b>Metode:</b>	
public Bankomat(int id, string adresa)	Javni konstruktor.
public Racun dekriptirajPodatkeORacunu(string podaci)	Javna metoda za čitanje podataka o računu iz kriptirane poruke u obliku stringa.
public string enkriptirajPodatkeORacunu(Racun racun)	Javna metoda za kriptiranje podataka o racunu.
public Transacija dekriptirajPodatkeOTransakciji(string transakcija)	Javna metoda za čitanje podataka o transakciji iz kriptirane poruke u obliku stringa.
public string enkriptirajPorukuOTransakciji(Transacija transakcija)	Javna metoda za kriptiranje podataka o transakciji.
public int dekriptirajPodatkePIN(int PIN)	Javna metoda za čitanje podataka o PIN-u iz kriptirane poruke u obliku cijelog broja.
public int enkriptirajPodatkePIN(int PIN)	Javna metoda za kriptiranje PIN-a.
public void kreirajLog()	Javna metoda za zapisivanje informacija o obavljenoj transakciji u log datoteku. Potrebno je zapisati datum, vrijeme transakcije te detalje vezane uz transakciju i korisnikov račun. Ako se dogodi greška prilikom rada bankomata treba ju zapisati u log datoteku.
public void ukljuciSe()	Javna metoda kojom se bankomat stavlja u pogon. Poziva ju administrator.
public void iskljuciSe()	Javna metoda kojom se bankomat isključuje. Poziva ju administrator.
public void resetirajSe()	Javna metoda za resetiranje bankomata. Poziva ju

	adminstrator.
public void obradiNovogKorisnika()	Javna metoda koja se poziva kada CitacKartice učita novu karticu.
public void prekidUplateNovca()	Javna metoda koja se poziva kada je korisnik naznačio da je gotov s uplatom novca.
public Bankomat vratiStanjeBankomata()	Javna metoda koju poziva administrator kako bi utvrdio trenutno stanje bankomata.

<b>Ime klase:</b> BankaDB	
<b>Opis klase:</b> Klase čuva podatke i metode za spajanje i rad s bazom podatka.	
<b>Atributi:</b>	
private int id	Privatni argument koji predstavlja identifikacijski broj. Ima pripadna get i set svojstva.
private string ime	Privatni argument koji predstavlja naziv banke. Ima pripadna get i set svojstva.
private string veza	Privatni argument koji služi za povezivanje s bazom podataka. Ima pripadna get i set svojstva.
<b>Metode:</b>	
public BankaDB(int id, string ime, string veze)	Javni konstruktor.
public int provjeriPIN(int PIN)	Javna metoda koja služi za dobivanje dodatnog podatka o PIN kartice iz baze podataka. Pomoću tog podatka koji se vraća klasi Bankomat provjerava se točnost PIN-a kojeg je unio korisnik.
public int obradiTransakciju(string transakcija)	Javna metoda čiji argument je string koji prestavlja kriptirane podatke o transakciji. Podatke je prvo potrebno dekriptirati, a zatim obaviti promjene na bazi. Nakon toga, potrebno je učitati nove podatke iz baze i poslati ih u kriptiranom obliku klasi Bankomat.
public provjeriKarticu(string kartica)	Javna metoda koja služi za provjeru valjanosti podataka na kartici. Podaci se šalju u kriptiranom obliku.



<b>Ime klase:</b> Administrator	
<b>Opis klase:</b> Klase čuva podatke o administratoru i omogućava dohvat podataka o bankomatu.	
<b>Atributi:</b>	
private int id	Privatni atribut koji predstavlja identifikacijski broj administratora. Ima pripadna get i set svojstva.
private string ime	Privatni atribut koji predstavlja ime administratora. Ima pripadna get i set svojstva.
private string prezime	Privatni atribut koji predstavlja prezime administratora. Ima pripadna get i set svojstva.
private int lokacija	Privatni atribut koji označava lokaciju administratora. Ima pripadna get i set svojstva.
<b>Metode:</b>	
public Administrator()	Javni konstruktor.
public void posaljiInfoAdminu(int idBankomata)	Javna metoda, poziva se kada čovjek – administrator želi provjeriti stanje bankomata.

Klasa *Dežurstvo* nastala je profinjavanjem asocijacijske veze mnogo naprama mnogo. Ona predstavlja par (bankomat, administrator), služi za čuvanje podataka o tome koji bankomat je pod nadležnosti kojeg administratora.

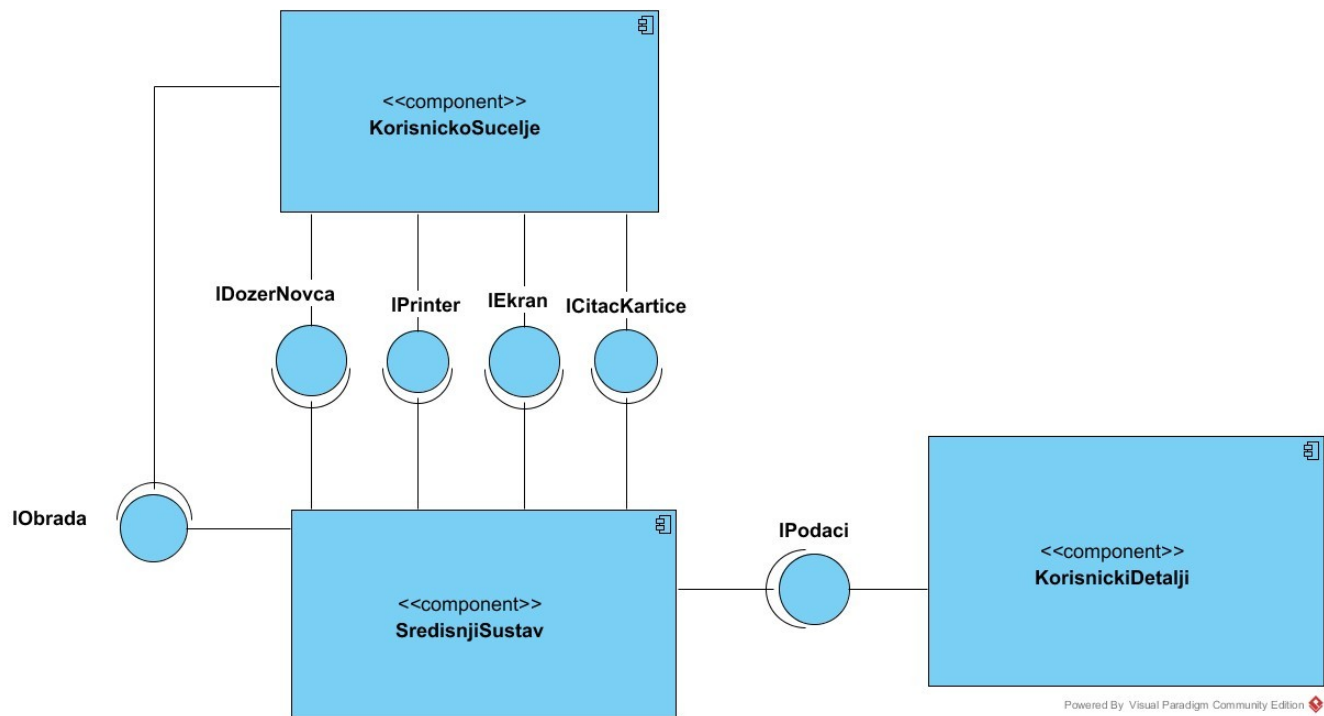
Komponenti *KorisnickiDetalji* pripadaju klase *Transakcija* koju nasljeđuju klase *UplataNovca*, *IsplataNovca*, *PromjenaPINa*, *ProvjeraStanjaRacuna* te klase *Racun* i *Kartica*. Klase *VektorTransakcija* je koleksijska klase nastala profinjavanjem asocijacijske veze jedan-naprema-mnogo. Klase su jednostavne. Imaju samo atribute zadužene za čuvanje podataka o trenutnom korisniku s pripadnim get i set svojstvima pa zbog toga nećemo navoditi njihov detaljan opis.

Na sljedećem class dijagramu nalaze se opisane klase i veze između njih.



## 4. Component diagram

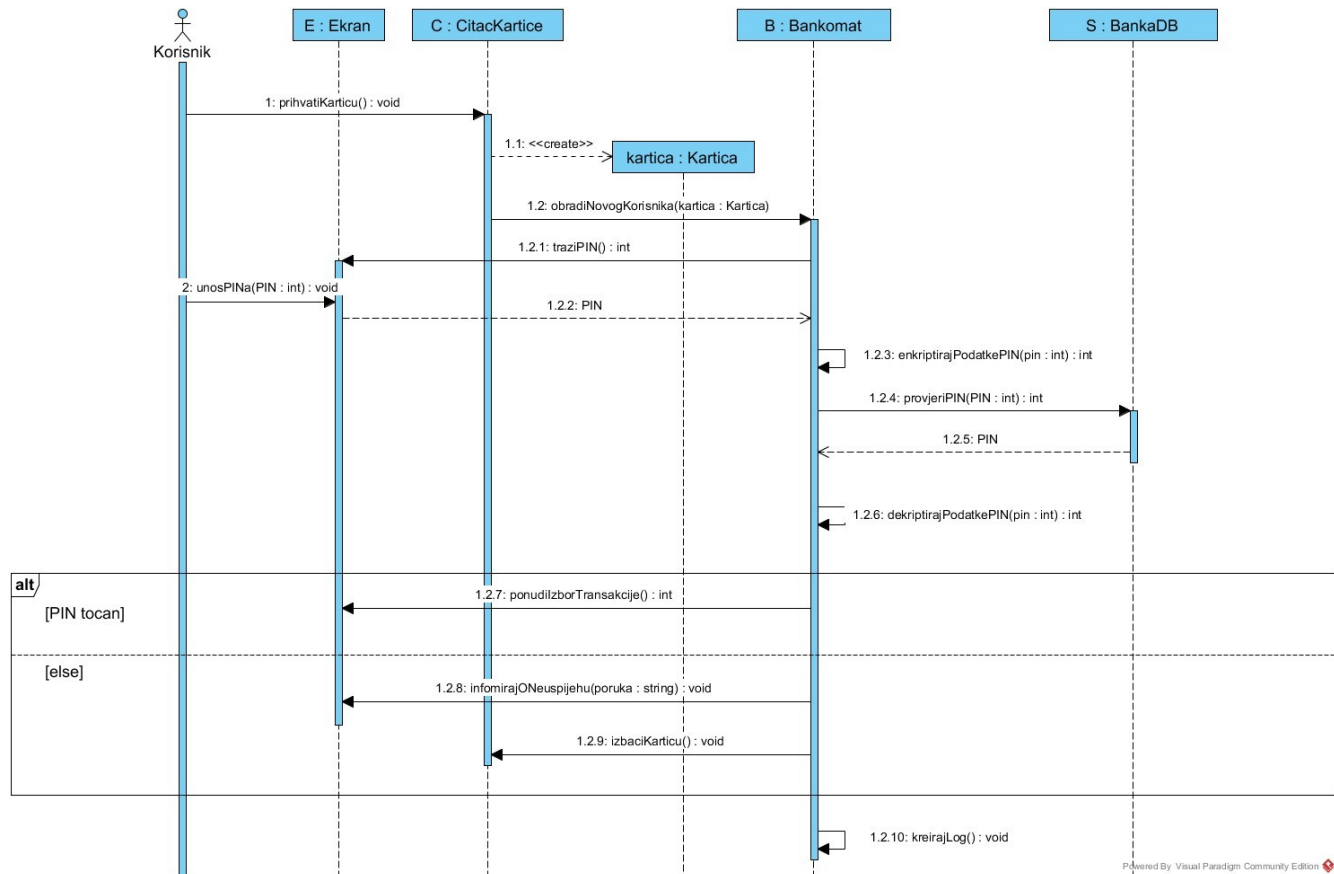
Opisana podjela klase po komponentama prikazana je na sljedećem dijagramu.



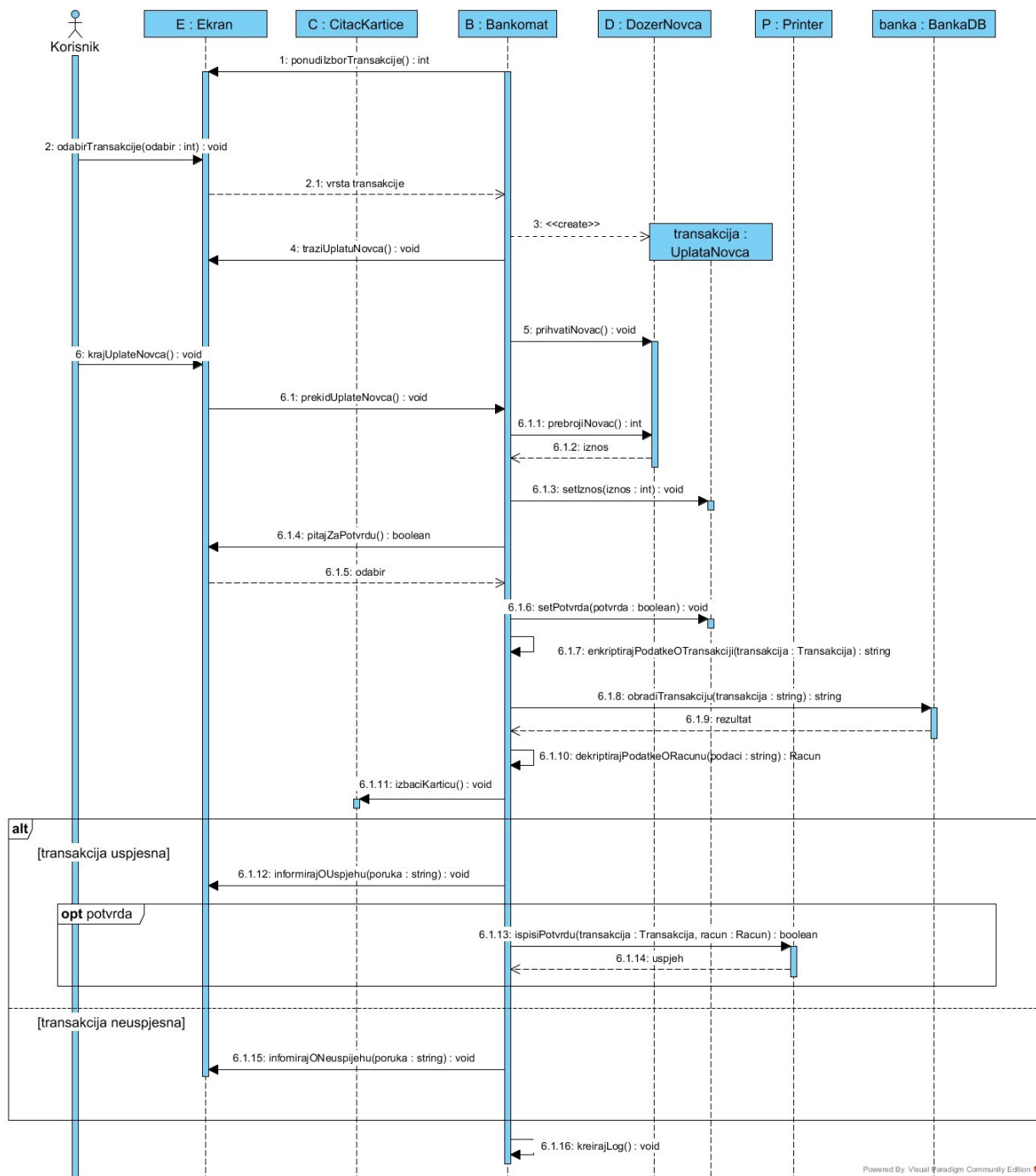
## 5. Sequence dijagrami

Slijedi prikaz profinjenih sequence dijagrama navedenih u prvoj zadaći.

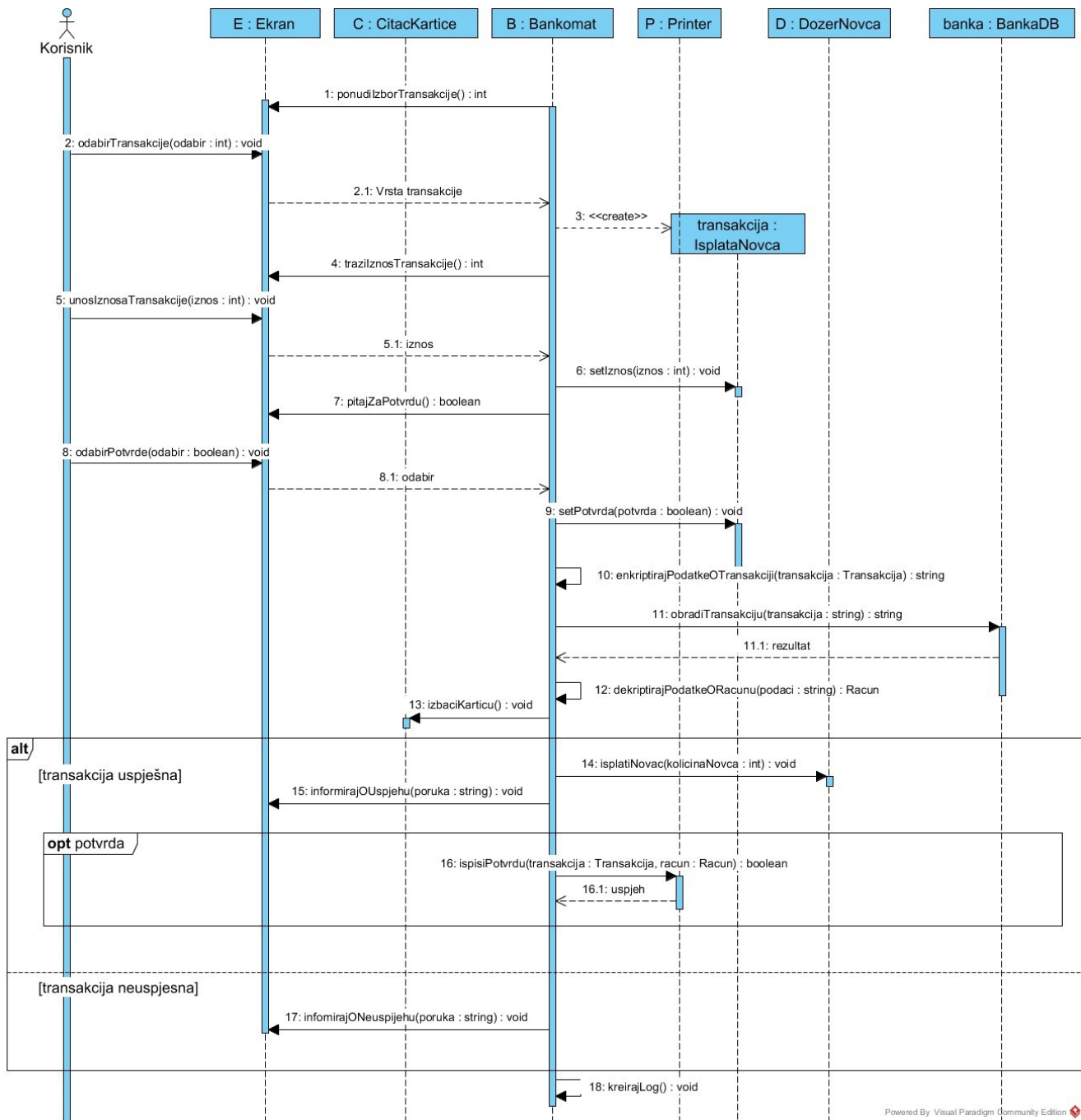
Sequence dijagram koji opisuje identifikaciju korisnika i provjeru točnosti unesenog PIN-a



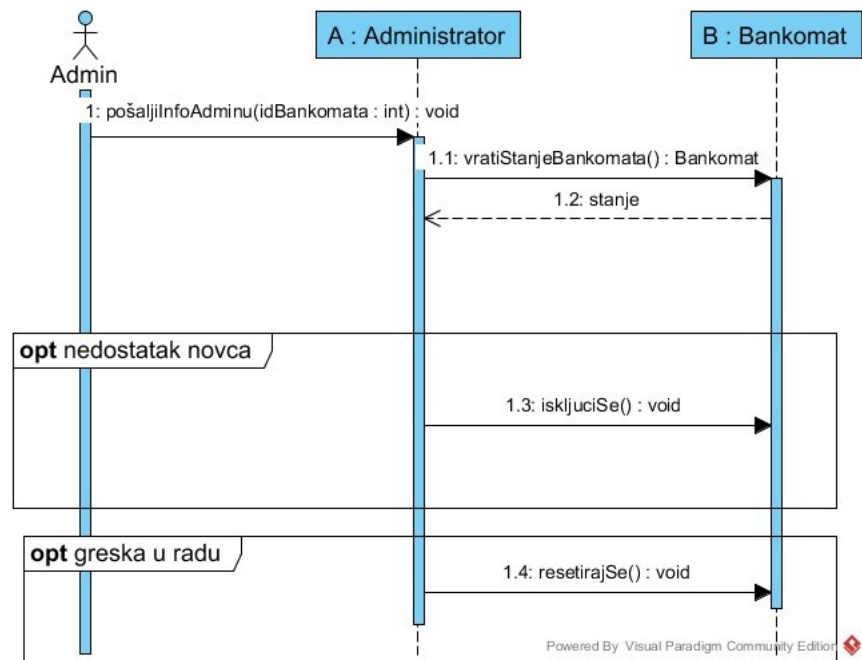
Sljedeći sequence diagram prikazuje uplatu novca na bankomatu.



Sljedeći sequence diagram prikazuje isplatu novca na bankomatu.



Sljedeći sequece dijagram prikazuje administratorovu provjeru ispravnosti rada bankomata i moguće događaje u slučaju neispravnosti.



## 6. State machine dijagram

Sljedeći state machine dijagram prikazuje stanja objekta tipa Bankomat i prelaske između pojedinih stanja objekta.

