

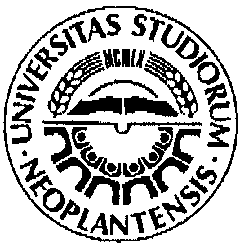
**UNIVERZITET U NOVOM SADU**

**PRIRODNO-MATEMATIČKI**

**FAKULTET**

**DEPARTMAN ZA MATEMATIKU**

**I INFORMATIKU**



Stefan Kalafatic, 113/19

Sistem za podršku vremenskog planiranja u zdravstvenoj ustanovi

- seminarski rad iz predmeta Skript jezici-

Novi Sad, 2020.

# Sadržaj

Sadržaj

[Sadržaj 2](#_Toc49405870)

[Uvod 2](#_Toc49405871)

[Opis programa 3](#_Toc49405872)

[Class Admin 3](#_Toc49405873)

[Class Lekar 8](#_Toc49405874)

[Class Pacijent 13](#_Toc49405875)

[Main() 13](#_Toc49405876)

[Menu1() 14](#_Toc49405877)

[Menu2() 15](#_Toc49405878)

[Menu3() 16](#_Toc49405879)

[Login() 17](#_Toc49405880)

[Zaključak 18](#_Toc49405881)

[Literatura 19](#_Toc49405882)

# Uvod

Sistem za podršku vremenskog planiranja u zdravstvenoj ustanovi je jednostavna aplikacija napisana u programskom jeziku Python kao seminarski rad u okviru teme Skript jezici. Napisan je sa ciljem da olakša i ubrza organizaciju vremena u zdravstvenoj ustanovi.

# Opis programa

U programskom jeziku Python napisan je program pod nazivom “Sistem za podršku vremenskog planiranja unutar zdravstvene ustanove”.

U ovakvom softverskom sistemu učesnici su

-Administrator

-Lekar

-Pacijent

Svaki korisnik sistema, bez obzira na ulogu opisan je sledećim podacima:

-Korisničko ime

-Lozinka

-Ime

-Prezime

Pored osnovnih poodataka u zavisnosti od uloge, neki korisnici su opisani dodatnim podacima.

Lekar:

-Spisak zakazanih pregleda

-Lista pacijenata

Pacijent:

-JMBG

-Lista pregleda

U nastavku ćemo detaljnije opisati svaku od datih funkcionalnosti.

## Class Admin

Administrator sistema rukuje mnogim funkcionalnostima kao što su: dodavanje/zapošljavanje novih lekara, štampanje svih lekara, brisanje lekara/otpuštanje, štampanje liste lekara sa šiframa i iscrtavanje grafika. Krenućemo sa opisom prve (Figure 1.).



Figure 1 dodavanje\_lekara()

Funkcija dodavanje\_lekara() može da registruje novog korisnike sistema (Lekara). Prilikom registracije novih korisnika administrator unosi: ime i prezime koji su ujedno i jedinstveno korisničko ime dok se automatski generiše lozinka.

Prilikom „zapošljavanja“ novog lekara, njemu se otvaraju 2 fajla: prvi u kojima ce čuvati listu svojih pacijenata, drugi u kom će imati uvid u svoje preglede.

Sledeća funkcija je stampanje\_liste\_lekara() (Figure 2.) u kojoj prolazimo kroz listu lekara i stampamo svaki element te liste.

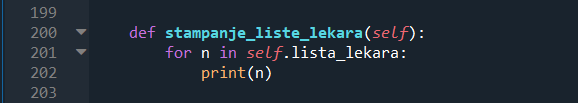


Figure 2. stampanje\_liste\_lekara()

U funkciji brisanje\_lekara()(Figure 3.) unosimo ime lekara kog zelimo da otpustimo. Pošto su u listi lekara oni zapisani u obliku IME-PREZIME, sa funkcijom split() uzimamo ime i prezime tog lekara da bi mogli da obrisemo njegove fajlove sa listom pregleda i listom pacijenata. Datog lekara brisemo i iz liste i iz fajla Lista\_lekara.txt.

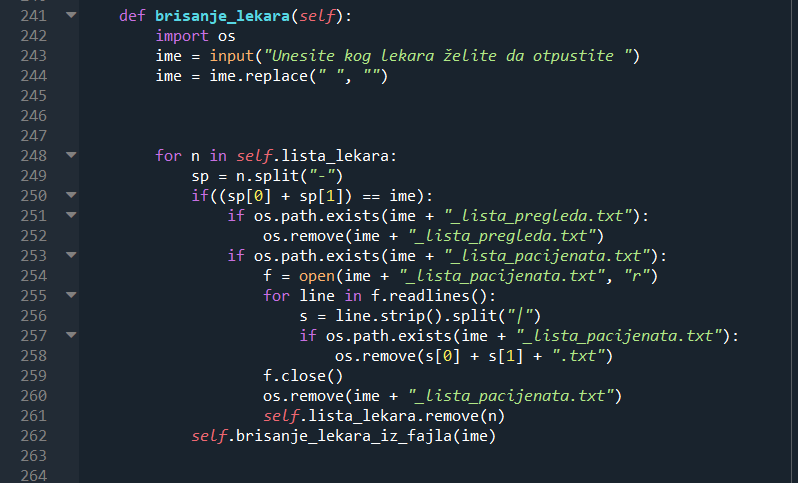


Figure 3. brisanje\_lekara()

Fukcija brisanje\_lekara\_iz\_fajla() je pomoćna funkcija koja se koristi u gore navedenoj brisanje\_lekara()koja brise datog lekara iz fajla, tako što ispisuje novi (ustvari isti taj fajl) bez tog lekara.(Figure 4.)

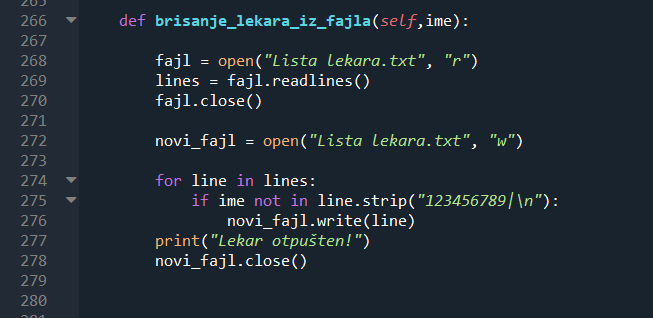


Figure 4. brisanje\_lekara\_iz\_fajla()

Sledeca nešto jednostavnija funkcija je stampanje\_liste\_lekara\_sa\_siframa(). Radi na principu otvaranja fajla "Lista lekara.txt" i štampa sve lekare sa njihovim šiframa. (Figure 5.).

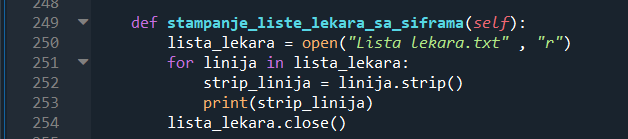


Figure 5. stampanje\_liste\_lekara\_sa\_siframa()

Takođe jos jedna funkcionalnost koju poseduje klasa Admin, koja je ujedno i poslednja, je iscrtavanje stubičastog grafika- funkcija grafik(). Iscrtava odnos lekara i njegov broj pacijenata. (Figure 6.)

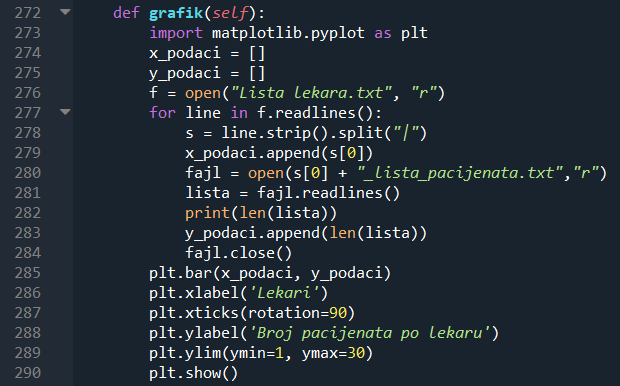
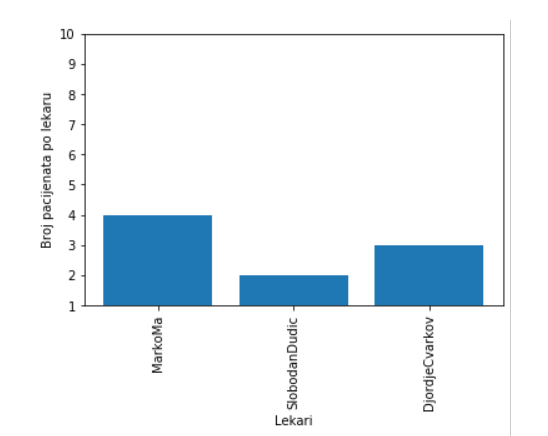


Figure 6. grafik()

Podatke za iscrtavanje grafika dobija prvo otvaranjem fajla „Lista lekara.txt“, u kom uzima ime prvog lekara, zatim otvara fajl tog lekara i uzima sve pacijente kojima je on lekar. I tako za svakog lekara koji je zaposlen u bolnici.



Primer Grafika

Funkcija koja nam pomaže da učitamo sve prethodno unete informacije, kada započinjemo ponovo korišćenje programa, je loadAdin(). (Figure 7.)

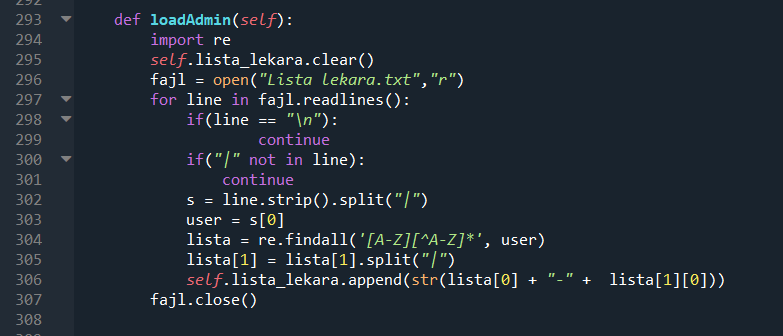


Figure 7. loadAdin()

## Class Lekar

Takođe, kao i Administrator, lekar ima različite ugrađene funkcionalnosti. Zapošljavanjem novog lekara otvara se lista pacijenata, i lista pregleda (pregledi se deklarišu kao rečnici).

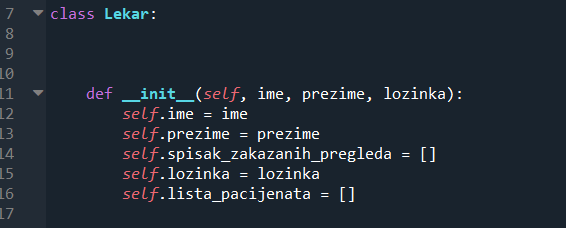


Figure 1 Prvi deo dodavanje\_pacijenta()

Dodavanje pacijenta, zakazivanje pregleda, štampanje pregleda za neki datum, štampanje svih pregleda, postavi dijagnozu pacijenta, štampaj pacijente sa dijagnozama su sve mogucnosti kojima može da se koristi zaposleni lekar.

Prva funkcija pod imenom dodaj\_pacijenta() je komplikovanija od ostalih, stoga ćemo je podeliti u tri dela

U prvom delu od korisnika se traže ime i prezime pacijenta, kao i njegov jmbg. Sledeći korak je dodavanje pacijenta u lekarov fajl (fajl pacijenta koji je memorisan kao njegovo ime i prezime). Proverava se da li fajl postoji, ako postoji upisuje se ime i prezime pacijenta kao i njegov jmbg.(Figure 1)

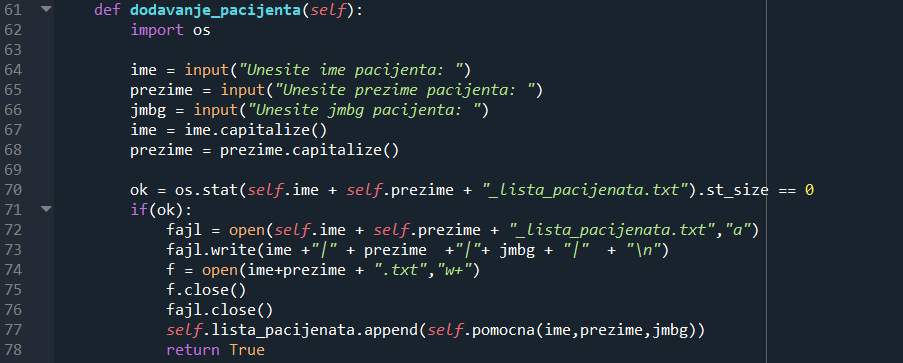


Figure 1 Prvi deo dodavanje\_pacijenta()

U drugom delu proveravamo da li taj pacijent postoji. Ponovo otvaramo fajl sa listom pacijenata za datog lekara i za svaku liniju fajla ispitujemo da li se poklapa sa novim imenom.(Figure 2)

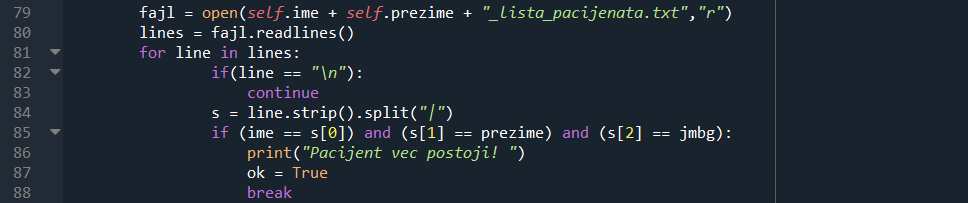


Figure 2 Drugi deo dodavanje\_pacijenta()

U trećem, ujedno i poslednjem delu funkcije dodavanje\_pacijenta(), pošto smo prošli proveru jedinstvenosti pacijenta, otvaramo mu fajl u kojem će kasnije biti smešteno: ime lekara, vreme, datum i cena pregleda. Takođe, pojavljuje se dodatna funkcija pomocna() koju ćemo kasnije detaljnije opisati(Figure 3.)

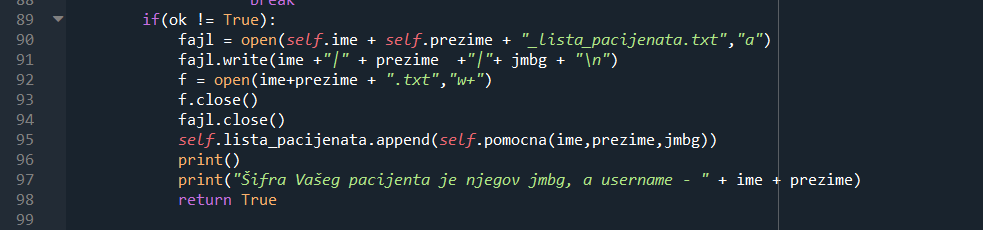


Figure 3 Treći deo dodavanje\_pacijenta()

Funkciju zakazivanje\_pregleda() smo spomenuli u prethodnoj. Ona omogućava lekaru da fajlu (kartonu) pacijenta dodeli: ime lekara, datum, vreme i cenu pregleda.

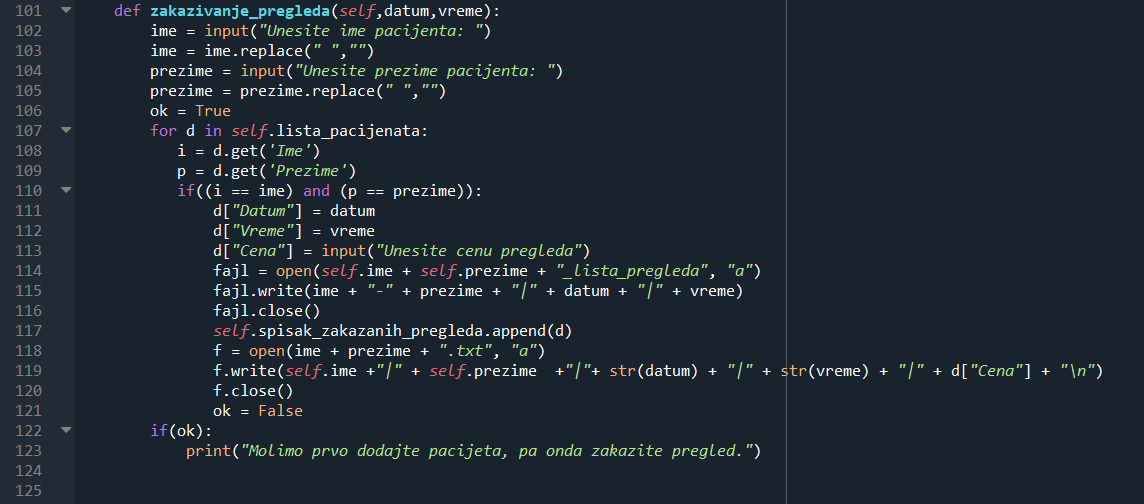


Figure 4 zakazivanje\_pregleda()

Slede dve veoma slične funkcije koje cemo ovde opisati.

Prva, stampaj\_spisakk\_pregleda(), stampa spisak pregleda za zadati datum. Importujemo modul date time sa kojim smo memorisali datum i kroz listu rečnika (spisak\_zakazanih\_pregleda) proveravamo sve preglede koje imaju zajednički ključ (datum) pa ih štampamo.(Figure 5)

Dok druga štampa svaki pregled (rečnik) iz liste spisak\_zakazanih\_pregleda.

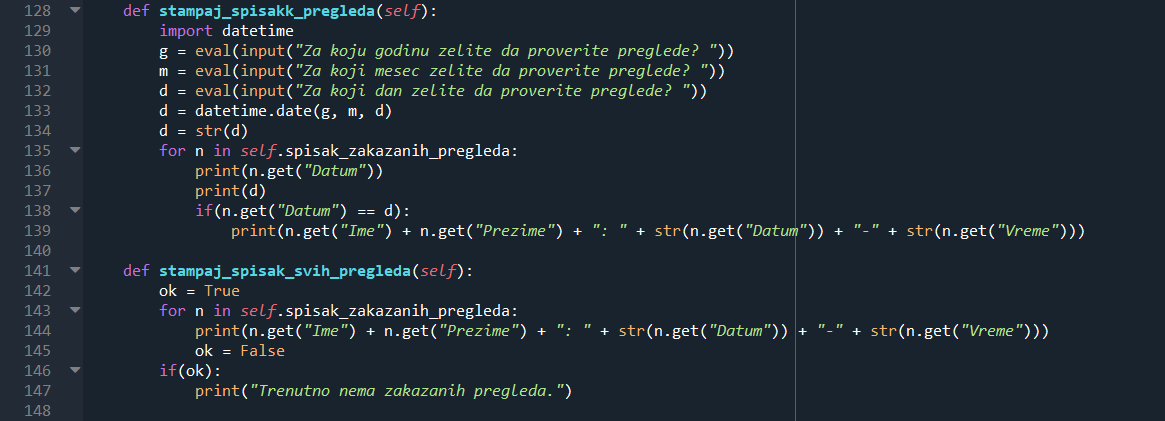


Figure 5 stampaj\_spisakk\_pregleda()

Sistem ne bi imao smisla ako se pacijentu ne mogu dodeljivati dijagnoze. To je rešeno funkcijom postavi\_dijagnozu(), koja “prolazi” kroz listu pregleda (lista rečnika) pronalazi dato ime i prezime pacijenta kojem zelimo postaviti dijagnozu i dodeljuje mu je.(Figure 6)

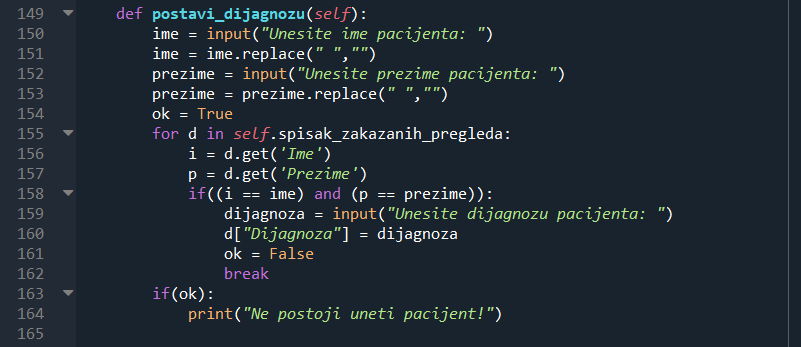


Figure 6 postavi\_dijagnozu()

Takodje funkcijom stampaj\_pacijente\_sa\_dijagnozama(), možemo videti sve pacijente I njihove dijagnoze.(Figure 7)

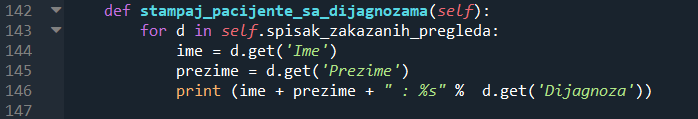


Figure 7 stampaj\_pacijente\_sa\_dijagnozama()

Poslednja funkcija koja je veoma bitna je loadLekar() . Prvo proveravamo da li taj fajl uopšte ima pacijenta, ako ima učitavamo ih. Učitavamo listu pacijenata iz fajla tog lekara, zatim pomoću imena tog pacijenta pronalazimo njegov fajl sa njegovim pregledima i učitavamo ih u listu spisak\_zakazanih\_pregleda.(Figure 8)

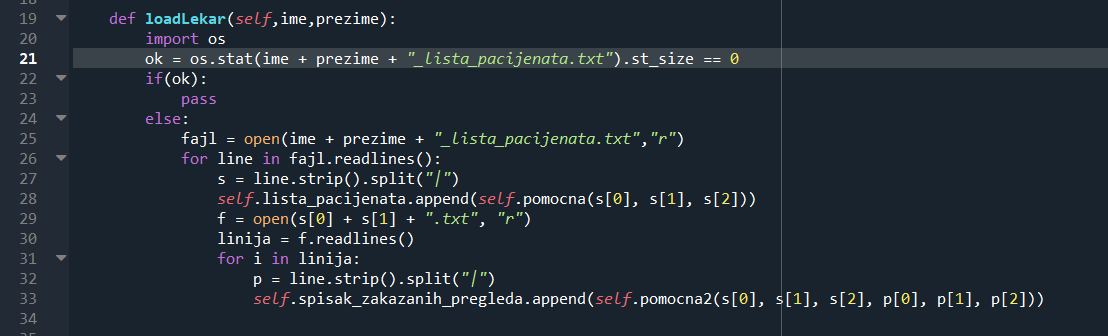


Figure 8 loadLekar()

## Class Pacijent

Klasa Pacijent nije složena kao ostale.Pacijent ima svoju listu pregleda kao i jednu funkciju koja otvara fajl pacijenta, i stampa sve preglede za datog pacijenta. (Figure 1)

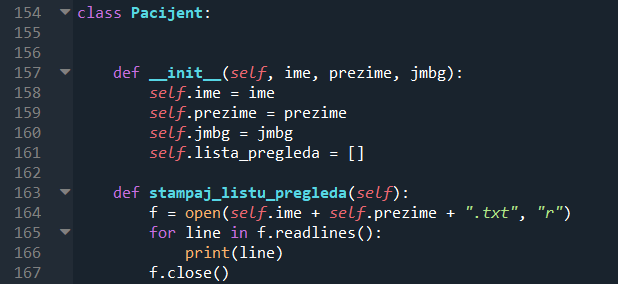


Figure 1 stampaj\_listu\_pregleda()

## Main()

Počinjemo program pozivom funkcije main(). U funkciji main() postoje 3 opcije: prijava kao admin, prijava kao lekar i prijava kao pacijent. Prvo se unosi odredjeni karakter u zavisnosti od opcije koju želimo. Zatim se poziva pomocna funkcija login() , koju cemo detaljnije opisati u nastavku, koja nam vraca vrednost tipa tuple(). “Raspakivanjem” tupla dobijamo 3 vrednosti, da li je prijavljivanje uspesno, ime i prezime korisnika. Proverom prvobinto unetog karaktera kao i vrednosti koje smo raspakovali tupla funkcija nas salje na odgovarajući meni.(Figure 1)

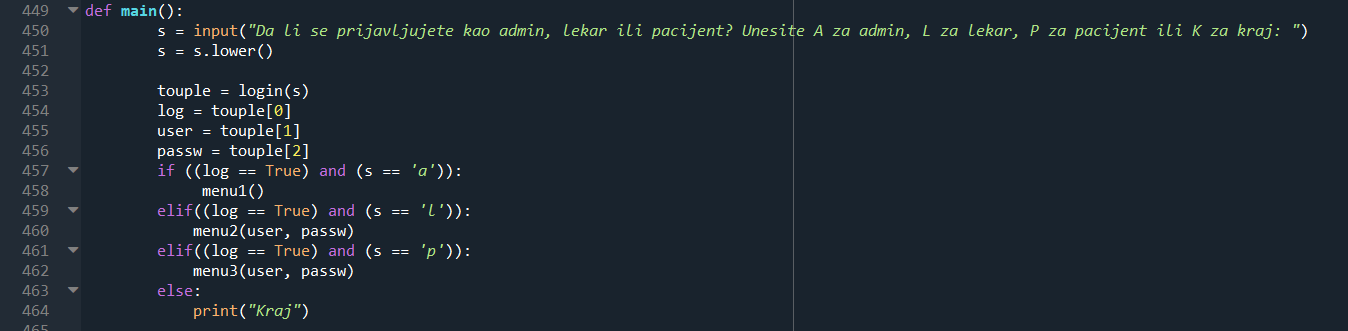


Figure 1 main()

## Menu1()

Meni Admina:

Ukoliko se korisnik prijavi na meni1(), u mogućnosti je da koristi sve funkcionalnosti kao Admin. Ako odabere neku od opcija: ‘1’, ‘2’, ‘3’, ‘4’, ‘5’, ‘6’ izvršiće se jedan od sledećih metoda:

Opcija ‘1’ – Pozivanje funkcije dodavanje lekara.

Opcija ‘2’ – Štampanje liste lekara

Opcija ‘3’ – Brisanje lekara

Opcija ‘4’ – Štampanje liste lekara sa šiframa

Opcija ‘5’ – Iscrtavanje grafika

Opcija ‘6’ – Pozivanje funkcije logout() i vraćanje na početni meni.

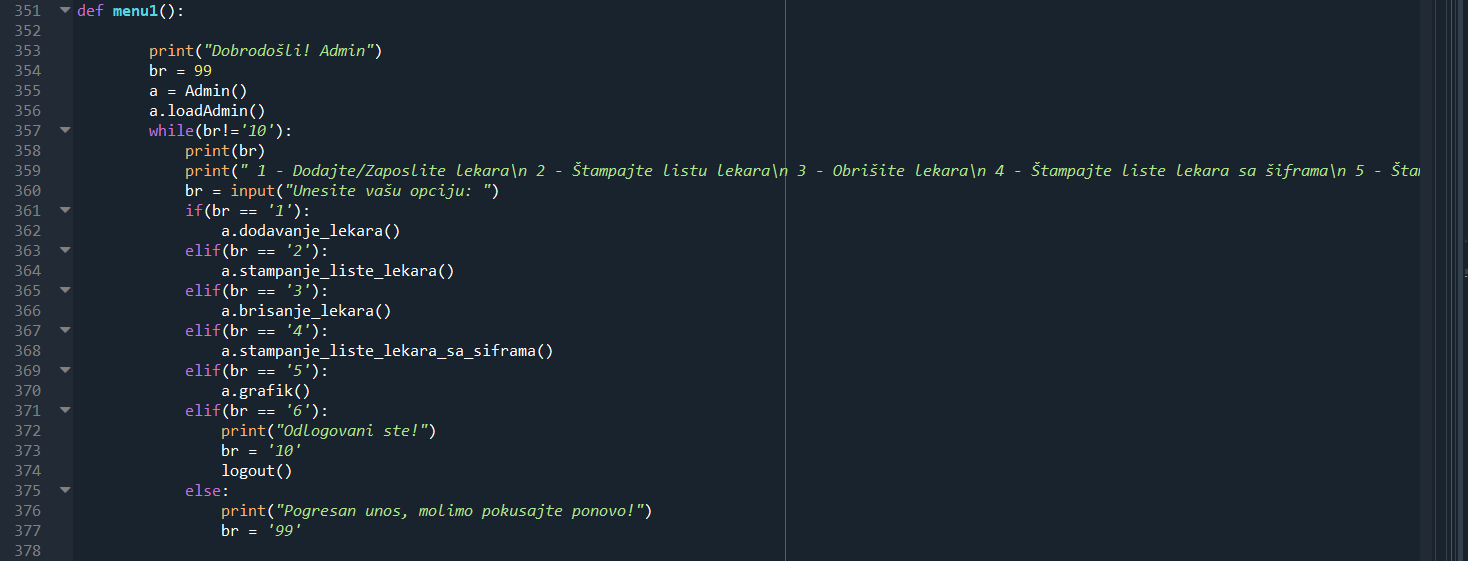


Figure 1 Menu1()

## Menu2()

Meni lekara:

Meni2() je nešto komplikovaniji od ostalih, zbog toga sto moramo znati ime i prezime lekara koji se prijavljuje. To smo rešili odvajamo string po velikim slovima. (Figure 1). Elemente koje dobijamo prosledjujemo klasi lekar. Nakon toga pozivamo funkciju loadLekar() koju smo gore opisali.

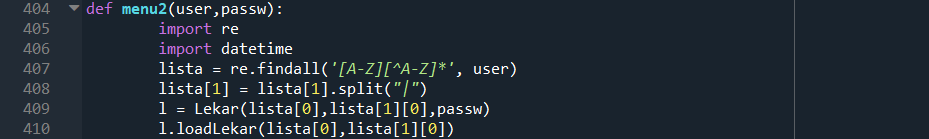


Figure 1 Menu2()

Nakon toga meni postaje sličan prošlom. Biramo neku od opcija koju program može da izvrši.(Figure 2)

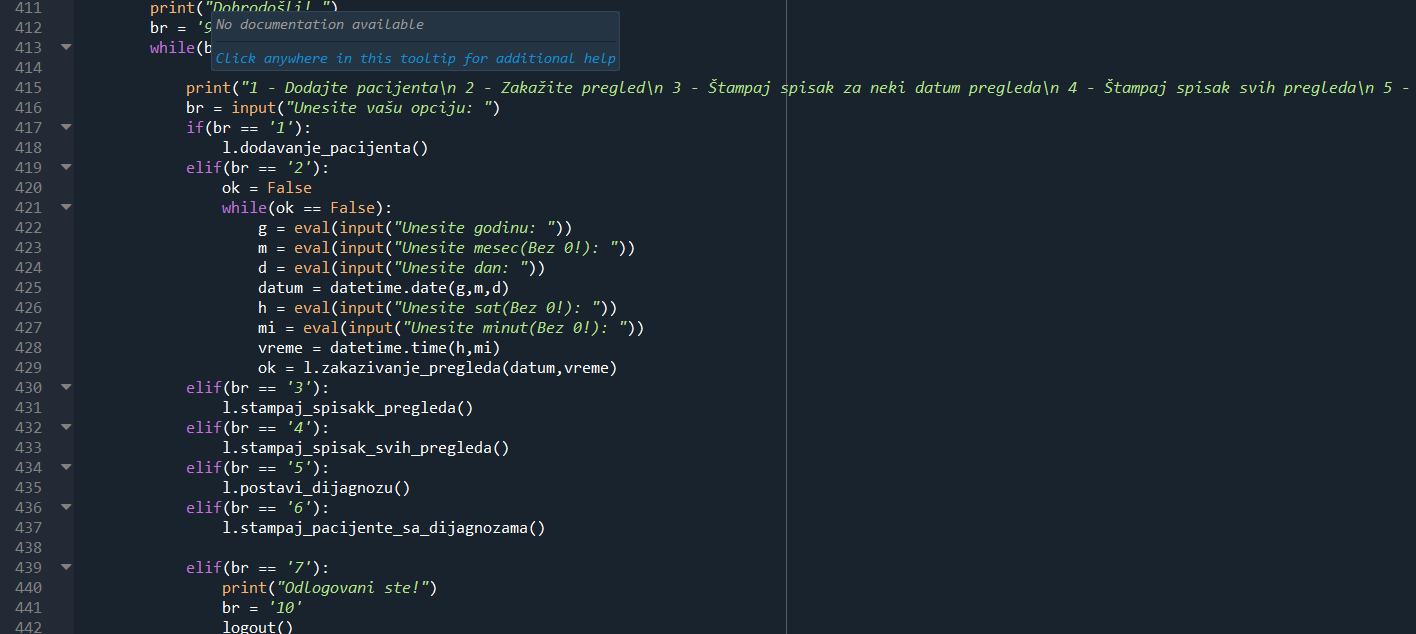


Figure 2 Menu2()

Opcija ‘1’ – Dodavanje novog pacijenta.

Opcija ‘2’ – Zakazivanje pregleda, biramo datum i vreme pregleda i zakazujemo pregled za datog pacijenta.

Opcija ‘3’ – Štampanje spiska pregleda za uneti datum.

Opcija ‘4’ – Štampanje liste svih pregleda za datog lekara.

Opcija ‘5’ – Postavljanje dijagnoze za unetog pacijenta

Opcija ‘6’ – Štampanje svih pacijenta sa njihovim dijagnozama

Opcija ‘7’ – Pozivanje funkcije logout() i vraćanje na prvobitni meni.

## Menu3()

Meni Pacijenta:

Analogno meniju2, razdvajamo ime i prezime pacijenta i prosledjujemo parametre klasi Pacijent.(Figure 1)

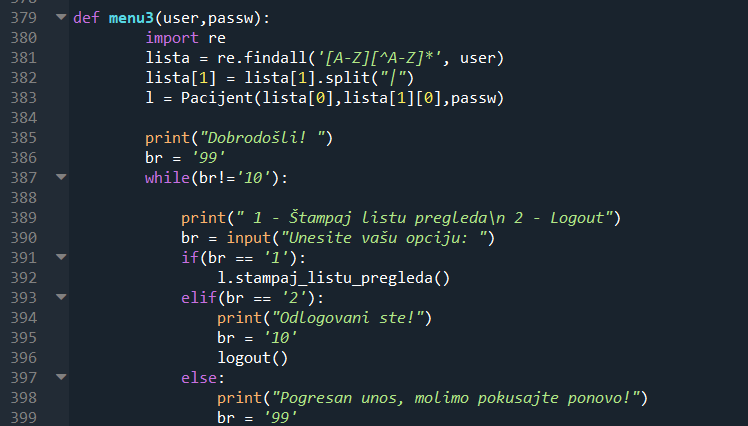


Figure 1 Menu3()

Ponovo korisnik bira neku od opcija:

Opcija ‘1’ – Štampanje liste pregleda pacijenta.

Opcija ‘2’- Pozivanje funkcije logout() i vraćanje na prvobitni meni.

## Login()

Funkcija login() je podeljena na tri dela. U zavisnosti od opcije koju je prethodno uneo proverava se dato korisničko ime i lozinka u odgovarajućim fajlovima:

Opcija ‘1’ – Korisnik se loguje kao admin:

Otvara se odgovarajući fajl u kojem je smeštena šifra kao i korisničko ime admina koje je definisano kao – KORISNIČKOIME|PASSWORD. Prolaskom kroz fajl, linija se deli na KORISNIČKOIME i PASSWORD koje se nakon toga proveravaju sa unetim vrednostima. Ukoliko je sve tačno funkcija vraća vrednost True.

Opcija ‘2’ – Korisnik se loguje kao lekar:

Analogno opciji 1.

Opcija ‘3’ – Korisnik se loguje kao pacijent:

Analogno opciji 1.

# Zaključak

Softver je baziran na principu objeknto-orijentisanog programiranja, pokriva mnoge mogućnosti rada i olakšava brzinu funkcionisanja zdravstvene ustanove. Poboljšanje programa moglo bi se naći u korišćenju baza podataka umesto fajlova kao i proširivanjem obima funkcionalnosti. Npr. Dodavanjem funkcija za rezervisanje sala, dodavanje novih lica i drugih zaposlenih pored lekara itd… Funkcionalnost ovog programa mogu pronaći razne drzavne pa i privatne zdravstvene ustanove koji bi u potpunosti olakšao rad zaposlenih.

# Literatura

1. <https://stackoverflow.com/>