

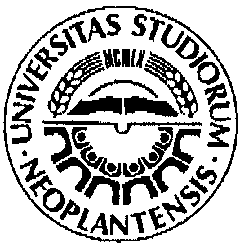
**UNIVERZITET U NOVOM SADU**

**PRIRODNO-MATEMATIČKI**

**FAKULTET**

**DEPARTMAN ZA MATEMATIKU**

**I INFORMATIKU**



Marko Mirković, 351/21

Aplikacija za vođenje evidencije i prodaju u prodavnici gitara

(odnosno, Stringy)

- seminarski rad iz predmeta Skript jezici-

Novi Sad, 2022.

# Sadržaj

Sadržaj

[Sadržaj 2](#_TOC000025)

[1. Uvod 3](#_TOC000026)

[2. Opis programa 5](#_TOC000027)

[2.1 main.py 5](#_TOC000028)

[2.2 Kupac.py 7](#_TOC000029)

[Importovanje biblioteka i globalne promenljive 7](#_TOC000030)

[pomoc() 7](#_TOC000031)

[menikupac() 7](#_TOC000032)

[sviproizvodi() 9](#_TOC000033)

[korpasadrzaj() 9](#_TOC000034)

[korpadodaj() 10](#_TOC000035)

[korpaobrisi() 10](#_TOC000036)

[promenistanje() 11](#_TOC000037)

[belezistatistiku() 12](#_TOC000038)

[kupi() 12](#_TOC000039)

[pretrazibrend() 13](#_TOC000040)

[pretrazibrojzica() 14](#_TOC000041)

[pretrazibrojpragova() 14](#_TOC000042)

[pretrazicena() 15](#_TOC000043)

[slikaproizvoda() 15](#_TOC000044)

[2.3 Radnik.py 17](#_TOC000045)

[Importovanje biblioteka 17](#_TOC000046)

[pomoc() 17](#_TOC000047)

[meniradnik() 17](#_TOC000048)

[prikazikupce() 18](#_TOC000049)

[promenicenu() 19](#_TOC000050)

[promenistanje() 19](#_TOC000051)

[dodajproizvod() 20](#_TOC000052)

[prikazigraf() 21](#_TOC000053)

[slikaproizvoda() 22](#_TOC000054)

[3. Zaključak 23](#_TOC000055)

[4. Literatura 24](#_TOC000056)

[4. Literatura 11](#__RefHeading___Toc3347_662000703)

# 1. Uvod

Svrha ovog programa je da upravlja sa poslovanjem prodavnice gitara. Program je fokusiran na rad sa korisnikom. Postoje dve vrste korisnika i na osnovu login-a određujemo ko je korisnik. Jedan tip korisnika je kupac, a drugi prodavac, odnosno radnik u prodavnici. Svaki tip korisnika ima svoje funkcionalnosti koje su odvojene od funkcionalnosti drugog korisnika. Funkcionalnosti kupca su orijentisane prema pregledu proizvoda i kupovini, dok su funkcionalnosti radnika orijentisane prema samom radu prodavnice i upravljanje proizvodima.

Funkcionalnosti kupca su:

* Pregled proizvoda, to uključuje prikaz detalja o proizvodu, i posebnu funkcionalnost pregleda slika;
* Pretraga proizvoda po različitim kriterijumima;
* Upravljanje korpom, sto uključuje dodavanje, brisanje i kupovinu.

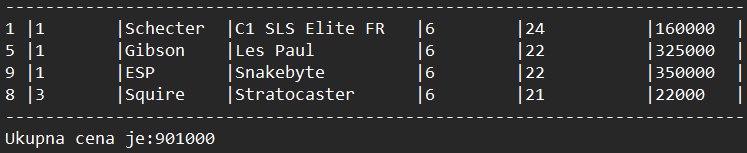
Funkcionalnosti radnika su:

* Pregled proizvoda kao i kod kupca, uz to da radnik moze videti i brojno stanje proizvoda;
* Pregled korisnika;
* Mogućnost promene stanja i cene proizvoda;
* Dodavanje novog proizvoda u prodaju;
* Pregled grafa koji prikazuje procenat prodaje proizvoda.

Podaci za rad programa su čuvani u vidu 4 tekstualna dokumenta i jednog direktorijuma koji sadrži slike proizvoda. Dokumenti su „Korisnici.txt“, „Racuni.txt“, „Roba.txt“, „Statistika.txt“ i direktorijum pod nazivom „Slike“. Struktura tekstualnih dokumenata je osmišljena da jedan red predstavlja jedan proizvod(u slučaju dokumenta „Roba.txt“, jedini izuzetak tog formata je dokument „Racuni.txt“ koji ima dodatnih redova kako bi olakšao preglednost korisnika), a karakteristike proizvoda u jednom redu su odvojene simbolom „|“. „Korisnici.txt“ sadrži podatke o korisnicima i njihove login informacije. „Racuni.txt“ sadrži odštampane račune koji nastaju prilikom kupovine. „Roba.txt“ sadrži podatke o proizvodima u prodaji. „Statistika.txt“ sadrži podatke o prodaji proizvoda koji se koriste za štampanje grafa.

Formati tekstualnih dokumenata:

* Korisnici.txt
  + Format: Id|Titula|Ime|Prezime|Broj telefona|Korisničko ime|Lozinka
  + Titula 0 predstavlja radnika, a titula 1 predstavlja kupca
  + Primer: 1|0|Goran|Darazanovic|0616789123|usr1|pass1
* Racuni.txt
  + Format: Id|Količina|Proizvod|Model|Broj žica|Broj pragova|Cena
  + Primer:

Slika 1. Primer računa u dokumentu „Racuni.txt“

* Roba.txt
  + Format: Id|Brend|Model|Broj žica|Broj pragova|Cena|Broj modela na stanju|Putanja do slike proizvoda
  + Primer: 1|Schecter|C1 SLS Elite FR|6|24|160000|6|podaci\Slike\c1sls.jpg
* Statistika.txt
  + Format: Id|Brend|Model|Broj prodanih primeraka
  + Primer: 1|Schecter|C1 SLS Elite FR|1

# 2. Opis programa

## 2.1 main.py

Kao glavni fajl, on sadrži funkciju koja pokreće sam program. On je odgovoran za login i raspoređivanje korisnika u odgovarajuće fajlove.

from Radnik import meniradnik

from Kupac import menikupac

korisnici=[]

Ovde importujemo funkcije meniradnik() i menikupac() iz fajlova Radnik.py i Kupac.py. Kada program odredi titulu on na osnovu nje kod nastavlja tako sto poziva meniradnik() ili menikupac(). main() ćemo pokazati iz više delova. Lista korisnici služi da u sebi čuva sve korisnike koji su učitani iz fajla.

def main():

  titula=-1

  imekorisnika=''

  while titula==-1:

    korisnickoime=input('Korisnicko ime: ')

    lozinka=input('Lozinka: ')

    with open('podaci\Korisnici.txt') as fajl:

      podaci=fajl.readlines()

      for red in podaci:

        korisnik=red.strip().split('|')

        korisnici.append(korisnik)

      for red in korisnici:

        if korisnickoime==red[5] and lozinka==red[6]:

          titula=red[1]

          imekorisnika=red[2]

    if titula==-1:

      print('Uneli ste pogresno korisnicno ime ili lozinku.')

      print('Pokusajte ponovo:')

    else:

      print('Login uspesan! Dobro dosao/la ' + imekorisnika)

Ovaj deo main() funkcije traži od korisnika da unese svoje korisničko ime i svoju lozinku, a potom iz fajla u memoriju učita sve korisnike i traži koji korisnik odgovara unetim podacima. Ako pronađe korisnika, čuva njegov id i ime. Ispod toga se nalazi provera, koja ispituje da li se promenila početna vrednost promenljive titula, jer ako nije to znači da program nije uspeo da pronađe korisnika. Sve to se nalazi unutar while petlje koja nam omogućava da od korisnika tražimo login dok se ne unese neki odgovarajući. Kada korisnik unese neki postojeći login onda prelazi na sledeći deo koda.

  if titula=='0':

    meniradnik()

  elif titula=='1':

    menikupac()

  else:

    print('Doslo je do greske. Nepoznata titula.')

Ovde proveravamo koju titulu ima korisnik koji se ulogovao. Ukoliko je titula „0“ onda izvršavanje programa prosleđujemo u fajl Radnik.py i pokrećemo funkciju meniradnik(), a ukoliko je titula „1“ onda izvršavanje programa prosleđujemo u fajl Kupac.py i pokrećemo funkciju menikupac().

if \_\_name\_\_=='\_\_main\_\_':

  main()

Ovde proveravamo da li promenljiva \_\_name\_\_ ima vrednost \_\_main\_\_, što označava da ovaj modul(fajl) sadrži main funkciju koju treba pokrenuti odmah po otvaranju fajla main.py.

## 2.2 Kupac.py

### Importovanje biblioteka i globalne promenljive

from matplotlib import pyplot as plt

from matplotlib import image as mpimg

korpa=[]

U ovom fajlu koristimo biblioteku pod nazivom matplotlib. Nju koristimo u funkciji slikaproizvoda() za prikaz slike na ekranu. To ćemo kasnije detaljnije objasniti. Globalna promenljiva korpa je lista, koja u sebi čuva proizvode u korpi.

### pomoc()

Ovo je funkcija koja se poziva kroz menikupac() i služi da korisniku prikaže dostupne komande za koje on može da zahteva od programa da se izvše. Funkcija je sačinjena od print() naredbi koje ispisuju dostupne komande.

def pomoc():

  print(' sviproizvodi   - Ispisuje listu svih proizvoda u prodavnici')

  print(' slikaproizvoda  - Prikazuje sliku zeljenog proizvoda')

  print(' pretrazi     - Omogucava vise razlicitih nacina pretrage')

  print(' korpasadrzaj   - Prikazuje koje proizvode imate u korpi')

  print(' korpadodaj    - Dodaje proizvod u korpu')

  print(' korpaobrisi   - Brise proizvod iz korpe')

  print(' kupi       - Izvrsava kupovinu i stampa racun')

  print(' pomoc      - Izbacuje listu dozvoljenih komandi')

  print(' zatvori     - Zatvara program\n')

### menikupac()

Kada se korisnik uloguje i program prepozna da je korisnik kupac, program nastavlja svoje izvrsavanje u fajlu Kupac.py tako sto pokrene funkciju menikupac(). Ova funkcija je odgovorna za pozivanje funkcija koje su dostupne korisniku. Kada se pokrene, ona traži unos od korisnika, i posle na osnovu tog unosa dalje zaključuje koju funkciju je potrebnu pokrenuti. Ako korisnik unese nepostojeću komandu, onda će tražiti ponovni unos zato što su provere smeštene unutar while petlje. Izvršavanje će se ponavljati sve dok korisnik ne unese „zatvori“.

def menikupac():

  unos=input('Odaberite komandu("pomoc" za listu komandi): ')

  while unos!='zatvori':

    if unos=='pomoc':

      pomoc()

    elif unos=='sviproizvodi':

      sviproizvodi()

    elif unos=='pretrazi':

      unos1=input('Pretraga po brendu, broju zica, broju pragova ili ceni? ')

      if unos1=='brendu':

        pretrazibrend()

      elif unos1=='broju zica':

        pretrazibrojzica()

      elif unos1=='broju pragova':

        pretrazibrojpragova()

      elif unos1=='ceni':

        pretrazicena()

      else:

        print('Uneli ste pogresnu opciju, vracamo vas na odabir komandi. ')

        pass

    elif unos=='slikaproizvoda':

      slikaproizvoda()

    elif unos=='korpasadrzaj':

      korpasadrzaj()

    elif unos=='korpadodaj':

      korpadodaj()

    elif unos=='korpaobrisi':

      korpaobrisi()

    elif unos=='kupi':

      kupi()

    elif unos=='':

      pass

    else:

      print('Uneta komanda ne postoji')

    unos=input('Odaberite komandu("pomoc" za listu komandi): ')

Ukoliko korisnik unese „pretrazi“ tada se taj deo koda grana i daje korisniku opciju da bira koji vid pretrage želi. Na kraju same funkcije se traži ponovni unos koji će se ispisati nakon svakog izlaska iz neke funkcije.

### sviproizvodi()

* + 1. def sviproizvodi():

  with open('podaci\Roba.txt') as roba:

    proizvodi=roba.readlines()

    print(f'{"id":2}|{"Proizvod":10}|{"Model":18}|{"Broj zica":9}|{"Broj pragova":12}|{"Cena":8}|')

    for red in proizvodi:

      proizvod=red.strip().split('|')

      print(f'{proizvod[0]:2}|{proizvod[1]:10}|{proizvod[2]:18}|{proizvod[3]:9}|{proizvod[4]:12}|{proizvod[5]:8}|')

    print(' ')

Ova funkcija služi da korisniku prikaže sve dostupne proizvode koji se nalaze u prodavnici. Funkcija prvo učita sve proizvode iz fajla u memoriju putem naredbe open(ovde koristimo open sa naredbom with što ne menja samu naredbu već nam dozvoljava da po završetku sa naredbom open ne moramo da zatvorimo fajl, već će se on sam zatvoriti). Naredba print štampa prvi red „tabele“ u kome ispisuje korisniku šta koja kolona predstavlja. Unutar naredbe print postoji i naredba f koja je zaslužna za formatiranje teksta kako bi sve bilo poravnato i uređeno. Nakon toga funkcija pokreće for petlju koja prolazi kroz sve proizvode u memoriji. Svaki proizvod će biti predstavljen kao lista svojih karakteristika i to pomoću funkcija strip()(koja briše whitespace karaktere sa početka i kraja reda) i split()(koja element deli na podelemente, gledajući od navedenog znaka do navedenog znaka). Na kraju preostaje ispisivanje svakog proizvoda pojedinačno, takođe koristeći naredbu f za formatiranje.

### korpasadrzaj()

def korpasadrzaj():

  if len(korpa)==0:

    print('Korpa je prazna.')

  else:

    print(f'{"id":2}|{"Kolicina":8}|{"Proizvod":10}|{"Model":18}|{"Broj zica":9}|{"Broj pragova":12}|{"Cena":8}|')

    cena=0

    for i in korpa:

      print(f'{i[0]:2}|{i[8]:8}|{i[1]:10}|{i[2]:18}|{i[3]:9}|{i[4]:12}|{i[5]:8}|')

      cena=cena+int(i[5])\*int(i[8])

    print('Ukupna cena je:', cena)

  print(' ')

Ova funkcija služi da prikaže korisniku šta se nalazi u korpi. Sastoji se od provere da li je korpa prazna, ako nije prazna štampa prvi red, nakon toga kreira promenljivu „cena“ koja će da sadrži vrednost korpe i pomoću for petlje štampa sav sadržaj korpe. I na kraju štampa ukupnu cenu korpe.

### korpadodaj()

def korpadodaj():

  print('Lista proizvoda: ')

  sviproizvodi()

  idpredmeta=input('Unesite id proizvoda koji zelite da dodate: ')

  kolicina=input('Unesite kolicinu proizvoda:')

  with open('podaci\Roba.txt') as roba:

    proizvodi=roba.readlines()

    for red in proizvodi:

      proizvod=red.strip().split('|')

      if int(idpredmeta)==int(proizvod[0]) and int(proizvod[6])>0:

        if int(proizvod[6])>=int(kolicina):

          proizvod.append(kolicina)

          korpa.append(proizvod)

          print('Uspesno ste dodali ' + kolicina + ' x ' + proizvod[1] + ' ' + proizvod[2] + '.')

        else:

          print('Nema dovoljno proizvoda na stanju.')

  print(' ')

Ova funkcija služi da korisniku omogući da odabrani proizvod doda u korpu radi kupovine. Prvo poziva funkciju sviproizvodi() kako bi korisniku dala uvid u trenutne proizvode i njihov id, jer se pomoću id-a proizvoda vrši odabir. Nakon toga se traži da korisnik unese id i količinu(koliko primeraka istog proizvoda želi). Potom program učitava proizvode iz fajle u memoriju, pravi od njih listu, traži proizvod i proverava da li ga ima na stanju. Ako proizvoda ima dovoljno na stanju koliko je korisnih tražio, proizvodu se dodaje karakteristika koliko je korisnik proizvoda tražio, proizvod se dodaje u korpu i korisnik se obaveštava da je proizvod dodat, u suprotnom ispisuje poruku da nema dovoljno proizvoda na stanju.

### korpaobrisi()

Služi za uklanjanje proizvoda iz korpe. Sam odabir proizvoda za brisanje je isti kao i u funkciji za dodavanje. Brisanje se vrši tako što for petlja prolazi kroz korpu dok ne nadje proizvod sa odgovarajućim id-om, ako ga nađe obrisaće ga pozivom na metod remove().

def korpaobrisi():

  print('Koji proizvod zelite da obrisete?')

  korpasadrzaj()

  brisi=input('Unesite id proizvoda koji zelite da obrisete iz korpe:')

  for i in korpa:

    if(i[0]==brisi):

      korpa.remove(i)

      print('Proizvod je uspesno izbacen iz korpe.')

  print(' ')

### promenistanje()

Ova funkcija se koristi kao pomoćna funkcija prilikom kupovine proizvoda i njoj korisnik ne može da pristupi. Kada se izvršava kupovina potrebno je da se promeni stanje proizvoda u fajlu. Ona kao prosleđene vrednosti uzima id proizvoda i količinu.

def promenistanje(idx, kolicinax):

  postoji=False

  noviproizvodi=[]

  with open('podaci\Roba.txt') as roba:

    proizvodi=roba.readlines()

    for red in proizvodi:

      proizvod=red.strip().split('|')

      if int(idx)==int(proizvod[0]):

        postoji=True

        proizvod[6]=int(proizvod[6])-int(kolicinax)

      noviproizvod=proizvod[0]+'|'+proizvod[1]+'|'+proizvod[2]+'|'+proizvod[3]+'|'+proizvod[4]+'|'+proizvod[5]+'|'+str(proizvod[6])+'|'+proizvod[7]+'\n'

      noviproizvodi.append(noviproizvod)

  if postoji==True:

    with open('podaci\Roba.txt', 'w') as roba:

      for red in noviproizvodi:

        roba.write(red)

Učitavamo proizvode u memoriju, zatim pomoću prosleđenog id-a nađemo proizvod, izračunamo novu količinu i upišemo je u memoriju. Onda iz memorije taj proizvod koji je u obliku liste sastavimo u string i taj string čuvamo u novoj listi, na kraju ako je zadovoljen uslov da je nađen proizvod sa prosleđenim id-em, upisujemo iz memorije u fajl pomoću metoda write. Pošto je fajl otvoren u modu „w“ to znači da će prethodni sadržaj fajla biti obrisan i upisan novi.

### belezistatistiku()

Funkcija služi da nakon svake prodaje u fajl upiše koliko je proizvoda prodato i ti podaci se kasnije koriste za štampanje grafa. Funkcija radi na potpuno istom principu kao i promenistanje(). Jedina razlika je da je string drugog formata i da se umesto oduzimanja sa stanja sada dodaje.

def belezistatistiku(idx, kolicinax):

  postoji=False

  noviproizvodi=[]

  with open('podaci\Statistika.txt') as roba:

    proizvodi=roba.readlines()

    for red in proizvodi:

      proizvod=red.strip().split('|')

      if int(idx)==int(proizvod[0]):

        postoji=True

        proizvod[3]=int(proizvod[3])+int(kolicinax)

      noviproizvod=proizvod[0]+'|'+proizvod[1]+'|'+proizvod[2]+'|'+str(proizvod[3])+'\n'

      noviproizvodi.append(noviproizvod)

  if postoji==True:

    with open('podaci\Statistika.txt', 'w') as roba:

      for red in noviproizvodi:

        roba.write(red)

### kupi()

Svrha ove funkcije je da izvrši kupovinu proizvoda iz korpe i da odštampa račun u odgovarajući fajl. Funkcioniše tako što formira string sa proizvodom i štampa ga u fajl. Nakon odštampanih proizvoda odštampa ukupnu cenu koju je izračunao. Sadrži i pisanje linija koje odvajaju račune radi preglednosti. Kada završi sa štamanjem računa, program poziva funkcije promenistanje() i belezistatistiku() za svaki proizvod iz korpe i nakon toga čisti korpu. Naravno, sve to se izvršava ukoliko u korpi postoji neki sadržaj.

def kupi():

  if len(korpa)!=0:

    cena=0

    with open('podaci\Racuni.txt', 'a') as racuni:

      racuni.write('--------------------------------------------------------------------------\n')

      for i in korpa:

        red1=f'{i[0]:2}|{i[8]:8}|{i[1]:10}|{i[2]:18}|{i[3]:9}|{i[4]:12}|{i[5]:8}|'

        racuni.write(red1)

        racuni.write('\n')

        cena=cena+int(i[5])\*int(i[8])

      racuni.write('--------------------------------------------------------------------------\n')

      red2='Ukupna cena je:' + str(cena)

      racuni.write(red2)

      racuni.write('\n')

    for i in korpa:

      promenistanje(i[0], i[8])

      belezistatistiku(i[0], i[8])

    korpa.clear()

    print('Hvala na kupovini, racun je odstampan i korpa ispraznjena.')

  else:

    print('Korpa je prazna.')

  print(' ')

### pretrazibrend()

def pretrazibrend():

  brend=input('Unesite brend koji zelite da pretrazite: ')

  postoji=False

  with open('podaci\Roba.txt') as roba:

    proizvodi=roba.readlines()

    print(f'{"id":2}|{"Proizvod":10}|{"Model":18}|{"Broj zica":9}|{"Broj pragova":12}|{"Cena":8}|')

    for red in proizvodi:

      proizvod=red.strip().split('|')

      if brend.lower()==proizvod[1].lower():

        postoji=True

        print(f'{proizvod[0]:2}|{proizvod[1]:10}|{proizvod[2]:18}|{proizvod[3]:9}|{proizvod[4]:12}|{proizvod[5]:8}|')

  if postoji==False:

    print('Nemamo proizvode sa navedenim brendom.')

  print(' ')

Sve funkcije za pretragu su zasnovane na istom principu, razlikuju se samo po kriterijumu pretrage. Funkcija traži od korisnika da unese traženi podatak. Iz fajla upisuje sve proizvode u memoriju i na ekran ispisuje samo proizvode koji zadovoljavaju uslov, odnosno imaju traženu karakteristiku.

### pretrazibrojzica()

def pretrazibrojzica():

  brzica=input('Unesite broj zica koji zelite: ')

  postoji=False

  with open('podaci\Roba.txt') as roba:

    proizvodi=roba.readlines()

    print(f'{"id":2}|{"Proizvod":10}|{"Model":18}|{"Broj zica":9}|{"Broj pragova":12}|{"Cena":8}|')

    for red in proizvodi:

      proizvod=red.strip().split('|')

      if int(brzica)==int(proizvod[3]):

        postoji=True

        print(f'{proizvod[0]:2}|{proizvod[1]:10}|{proizvod[2]:18}|{proizvod[3]:9}|{proizvod[4]:12}|{proizvod[5]:8}|')

  if postoji==False:

    print('Nemamo proizvode sa navedenim brojem zica.')

  print(' ')

### pretrazibrojpragova()

def pretrazibrojpragova():

  brpragova=input('Unesite broj pragova koji zelite: ')

  postoji=False

  with open('podaci\Roba.txt') as roba:

    proizvodi=roba.readlines()

    print(f'{"id":2}|{"Proizvod":10}|{"Model":18}|{"Broj zica":9}|{"Broj pragova":12}|{"Cena":8}|')

    for red in proizvodi:

      proizvod=red.strip().split('|')

      if int(brpragova)==int(proizvod[4]):

        postoji=True

        print(f'{proizvod[0]:2}|{proizvod[1]:10}|{proizvod[2]:18}|{proizvod[3]:9}|{proizvod[4]:12}|{proizvod[5]:8}|')

  if postoji==False:

    print('Nemamo proizvode sa navedenim brojem pragova.')

  print(' ')

### pretrazicena()

def pretrazicena():

  cena=input('Unesite maksimalnu cenu proizvoda: ')

  postoji=False

  with open('podaci\Roba.txt') as roba:

    proizvodi=roba.readlines()

    print(f'{"id":2}|{"Proizvod":10}|{"Model":18}|{"Broj zica":9}|{"Broj pragova":12}|{"Cena":8}|')

    for red in proizvodi:

      proizvod=red.strip().split('|')

      if int(cena)>=int(proizvod[5]):

        postoji=True

        print(f'{proizvod[0]:2}|{proizvod[1]:10}|{proizvod[2]:18}|{proizvod[3]:9}|{proizvod[4]:12}|{proizvod[5]:8}|')

  if postoji==False:

    print('Nemamo proizvode u tom cenovnom rangu.')

  print(' ')

Jedina razlika kod ove funkcije je to što će ispisati sve proizvode koji su navedene i cene niže od nje.

### slikaproizvoda()

Ova funkcija služi za prikazivanje slike odabranog proizvoda koristeći biblioteku matplotlib. Funkcija ispiše korisniku listu proizvoda i zahteva da unese id proizvoda ciju sliku želi da vidi. Pronalazi traženi proizvod i pamti njegov brend, model i putanju do slike. Brend i model upisuje u naslov slike. Naredbe plt.xticks() i plt.yticks() služe da uklone brojeve sa prikaza slike. U promenljivu „slika” upisujemo sliku pomoću naše biblioteke i naredbama plt.imshow() i plt.show() štampamo sliku na ekran.

def slikaproizvoda():

  print('Lista proizvoda: ')

  sviproizvodi()

  idx=input('Unesite id proizvoda ciju sliku zelite da vidite: ')

  postoji=False

  with open('podaci\Roba.txt') as roba:

    proizvodi=roba.readlines()

    for red in proizvodi:

      proizvod=red.strip().split('|')

      if idx==proizvod[0]:

        postoji=True

        brend=proizvod[1]

        model=proizvod[2]

        putanja=proizvod[7]

  if postoji==True:

    plt.title(brend+' '+model)

    plt.xticks(color='w')

    plt.yticks(color='w')

    slika = mpimg.imread(putanja)

    plt.imshow(slika)

    plt.show()

  else:

    print('Proizvod ne postoji.')

## 2.3 Radnik.py

### Importovanje biblioteka

from matplotlib import pyplot as plt

from matplotlib import image as mpimg

Importovanje biblioteka je isto kao i u prethodnom slučaju, samo se sada koristi za štampanje grafa umesto štampanja slika.

### pomoc()

Ova funkcija je ista kao i u slučaju za kupca, samo se razlikuje po tome što štampa druge opcije.

def pomoc():

  print(' sviproizvodi   - Ispisuje listu svih proizvoda u prodavnici')

  print(' slikaproizvoda  - Prikazuje sliku zeljenog proizvoda')

  print(' prikazikupce   - Ispisuje listu registrovanih kupaca')

  print(' promenicenu   - Menja cenu proizvoda')

  print(' dodajproizvod  - Dodaje proizvod u izbor')

  print(' promenistanje  - Promeni stanje nekog proizvoda u magacinu')

  print(' prikazigraf   - Prikazuje graf prodaje')

  print(' pomoc      - Izbacuje listu dozvoljenih komandi')

  print(' zatvori     - Zavrsava kupovinu\n')

### meniradnik()

I ova funkcija je istog principa kao u slučaju za kupca, samo se razlikuje po opcijama koje nudi.

def meniradnik():

  unos=input('Odaberite komandu("pomoc" za listu komandi): ')

  while unos!='zatvori':

    if unos=='pomoc':

      pomoc()

    elif unos=='sviproizvodi':

      sviproizvodi()

    elif unos=='prikazikupce':

      prikazikupce()

    elif unos=='promenicenu':

      promenicenu()

    elif unos=='dodajproizvod':

      dodajproizvod()

    elif unos=='promenistanje':

      promenistanje()

    elif unos=='prikazigraf':

      prikazigraf()

    elif unos=='slikaproizvoda':

      slikaproizvoda()

    elif unos=='':

      pass

    else:

      print('Uneta komanda ne postoji')

    unos=input('Odaberite komandu("pomoc" za listu komandi): ')

### prikazikupce()

def prikazikupce():

  with open('podaci\Korisnici.txt') as nalozi:

    kupci=nalozi.readlines()

    print(f'{"Ime":12}|{"Prezime":15}|{"Broj telefona":14}|')

    for red in kupci:

      kupac=red.strip().split('|')

      if kupac[1]=='1':

        print(f'{kupac[2]:12}|{kupac[3]:15}|{kupac[4]:14}|')

      else:

        pass

  print(' ')

Funkcija služi da radniku ispiše listu svih kupaca koji su registrovani. Funkcioniše na principima koje smo već objasnili kroz druge primere.

### promenicenu()

def promenicenu():

  print('Lista proizvoda: ')

  sviproizvodi()

  idpredmeta=input('Unesite id proizvoda ciju cenu zelite da promenite: ')

  novacena=input('Unesite novu cenu proizvoda: ')

  postoji=False

  noviproizvodi=[]

  with open('podaci\Roba.txt') as roba:

    proizvodi=roba.readlines()

    for red in proizvodi:

      proizvod=red.strip().split('|')

      if int(idpredmeta)==int(proizvod[0]):

        postoji=True

        proizvod[5]=novacena

      noviproizvod=proizvod[0]+'|'+proizvod[1]+'|'+proizvod[2]+'|'+proizvod[3]+'|'+proizvod[4]+'|'+proizvod[5]+'|'+proizvod[6]+'|'+proizvod[7]+'\n'

      noviproizvodi.append(noviproizvod)

  if postoji==True:

    with open('podaci\Roba.txt', 'w') as roba:

      for red in noviproizvodi:

        roba.write(red)

    print('Uspesno ste promenili cenu.\n')

  else:

    print('Proizvod ne postoji.\n')

Ova funkcija omogućava radniku da promeni cenu proizvoda i funkcionise na istom principu kao i funkcija promenistanje() kod kupca, jedino se razlikuje u tome što ovoj funkciji radnik moze da pristupi i ovaj put ne prosleđujemo parametre već ih zahtevamo od korisnika.

### promenistanje()

I ova funkcija je potpuno analogna istoimenoj funkciji od kupca, i takodje se razlikuje po tome što ovoj funkciji radnik moze da pristupi i ovaj put ne prosleđujemo parametre već ih zahtevamo od korisnika.

def promenistanje():

  print('Lista proizvoda: ')

  sviproizvodi()

  idpredmeta=input('Unesite id proizvoda cije stanje zelite da promenite: ')

  novostanje=input('Unesite novo stanje proizvoda: ')

  postoji=False

  noviproizvodi=[]

  with open('podaci\Roba.txt') as roba:

    proizvodi=roba.readlines()

    for red in proizvodi:

      proizvod=red.strip().split('|')

      if int(idpredmeta)==int(proizvod[0]):

        postoji=True

        proizvod[6]=novostanje

      noviproizvod=proizvod[0]+'|'+proizvod[1]+'|'+proizvod[2]+'|'+proizvod[3]+'|'+proizvod[4]+'|'+proizvod[5]+'|'+proizvod[6]+'|'+proizvod[7]+'\n'

      noviproizvodi.append(noviproizvod)

  if postoji==True:

    with open('podaci\Roba.txt', 'w') as roba:

      for red in noviproizvodi:

        roba.write(red)

    print('Uspesno ste promenili stanje.\n')

  else:

    print('Proizvod ne postoji.\n')

### dodajproizvod()

Ova funkcija služi za dodavanje novog proizvoda u ponudu. Zahteva od korisnika da unese sve parametre tog proizvoda. Upise postojeće proizvode u memoriju, dok to radi broji koliko ih ima kako bi mogla da novom proizvodu dodeli odgovarajući id i usput proverava da li postoji već taj proizvod. Ako proizvod ne postoji otvaramo fajl u modu „a“ što nam omogućava upisivanje bez brisanja, kreiramo string sa datim karakteristikama i upisujemo ga u fajl.

def dodajproizvod():

  brend=input('Unesite brend proizvoda: ')

  model=input('Unesite model proizvoda: ')

  brzica=input('Unesite broj zica proizvoda: ')

  brpragova=input('Unesite broj pragova proizvoda: ')

  cena=input('Unesite cenu proizvoda: ')

  kolicina=input('Unesite kolicinu proizvoda u magacinu: ')

  putanja=input('Unesite putanju do slike proizvoda: ')

  id=int(0)

  postoji=False

  with open('podaci\Roba.txt') as roba:

    proizvodi=roba.readlines()

    for red in proizvodi:

      id=id+1

      proizvod=red.strip().split('|')

      if brend==proizvod[1] and model==proizvod[2]:

        postoji=True

        print('Proizvod vec postoji.')

        return

  if postoji==False:

    with open('podaci\Roba.txt','a') as roba:

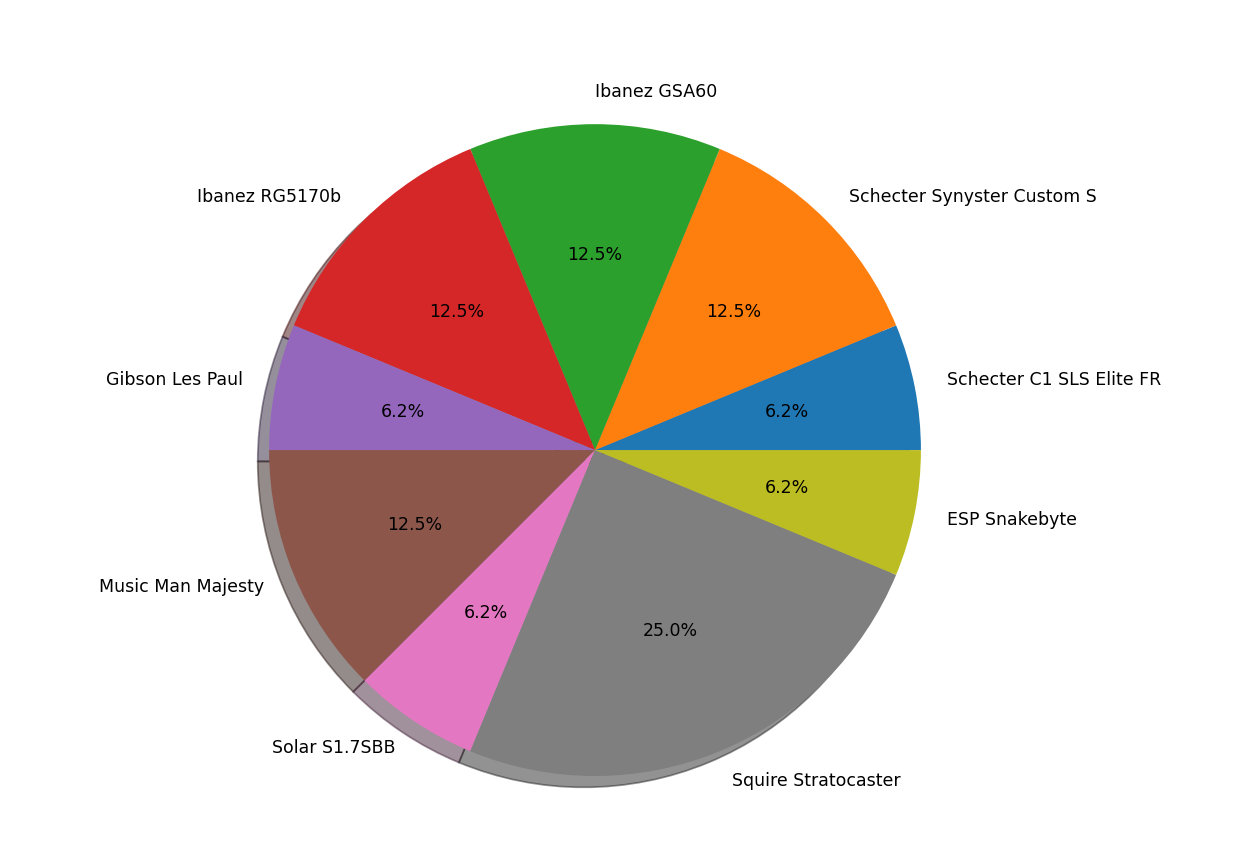
      id=id+1

      noviproizvod=str(id)+'|'+brend+'|'+model+'|'+brzica+'|'+brpragova+'|'+cena+'|'+kolicina+'|'+putanja+'\n'

      roba.write(noviproizvod)

      print('Uspesno ste dodali nov proizvod. \n')

### prikazigraf()

Za štampanje grafa koristimo biblioteku matplotlib i fajl „Satistika.txt“. Prvo kreiramo dve liste labels(koja čuva nazive proizvoda) i sizes(koja čuva vrednosti proizvoda). Te dve liste dobijaju svoje podatke iz fajla. Ostaje da podesimo sam graf. fig1 i ax1 postavimo kao subplot pomocu metoda iz biblioteke(to je nešto što sama biblioteka zahteva kako bi uspešno štampala graf). Nakon toga na ax1 pozovemo metod pie koji označava da će tip grafa biti pitasti graf i podesimo parametre(veličine, nazive, format ispisa i senka grafa). Ovo su korišćeni parametri u ovom grafu, ali postoji dosta više parametara za razne varijacije grafa. I na kraju štampamo graf na ekran.

def prikazigraf():

  labels=[]

  sizes=[]

  with open('podaci\Statistika.txt') as roba:

    proizvodi=roba.readlines()

    for red in proizvodi:

      proizvod=red.strip().split('|')

      if proizvod[3]!='0':

        labels.append(proizvod[1]+' '+proizvod[2])

        sizes.append(proizvod[3])

    fig1, ax1 = plt.subplots()

    ax1.pie(sizes, labels=labels, autopct='%1.1f%%', shadow=True)

    ax1.axis('equal')

    plt.show()

### slikaproizvoda()

Ova funkcija je identična funkciji slikaproizvoda() koja se nalazi kod kupca.

def slikaproizvoda():

  print('Lista proizvoda: ')

  sviproizvodi()

  idx=input('Unesite id proizvoda ciju sliku zelite da vidite: ')

  postoji=False

  with open('podaci\Roba.txt') as roba:

    proizvodi=roba.readlines()

    for red in proizvodi:

      proizvod=red.strip().split('|')

      if idx==proizvod[0]:

        postoji=True

        brend=proizvod[1]

        model=proizvod[2]

        putanja=proizvod[7]

  if postoji==True:

    plt.title(brend+' '+model)

    plt.xticks(color='w')

    plt.yticks(color='w')

    slika = mpimg.imread(putanja)

    plt.imshow(slika)

    plt.show()

  else:

    print('Proizvod ne postoji.')

# 3. Zaključak

Program koji smo demonstrirali se uz male promene moze prilagoditi i za druge tipove prodavnica ili poslovanja generalno. Zato što je kod podeljen u delove za kupca i za radnika postoji mogućnost da se i to proširi eventualno na poslodavca. To je takođe pomoglo jednostavnosti čitanja koda i jednostavnije je dodavanje novih funkcionalnosti. Postoje detalji koji mogu da se unaprede, na primer kada radnik doda nov proizvod i upiše putanju do slike, on mora ručno da doda sliku. To bi moglo da se automatizuje da se slika pronađe na interneru i preuzme na odgovarajuće mesto sa odgovarajucim nazivom.

# 4. Literatura

1. Python Software Foundation, <https://www.python.org/>
2. Python Tutorial, <https://www.w3schools.com/python/default.asp>
3. Matplotlib, [https://matplotlib.org/](https://matplotlib.org/stable/gallery/)
4. Stackoverflow, <https://stackoverflow.com>