



UNIVERZITET U NOVOM SADU  
PRIRODNO-MATEMATIČKI  
FAKULTET  
DEPARTMAN ZA MATEMATIKU  
I INFORMATIKU



Dejan Grujić, 9/19

Aplikacija za vođenje evidencije u hotelu

- seminarski rad iz predmeta Skript jezici-

Novi Sad, 2020.

## Sadržaj:

1. Uvod.....	3
<i>O programskom jeziku "Python"</i> .....	3
<i>Opis programa</i> .....	3
2. Opis programa .....	4
2.1 Klasa "Recepcija" .....	4
2.2 Klasa „Gosti“ .....	5
2.3 Klasa „GlavniProgram“ .....	15
2.3 Kako sve to izgleda ? .....	19
3. Zaključak .....	23
4. Literatura .....	24

# 1. Uvod

## *O programskom jeziku "Python"*

Python je Objektivno orijentisani programski jezik koji je stvorio Gvido van Rosum, sa Univerziteta Stičing u Holandiji, decembra 1989. godine. Poslednja verzija Pythona je 3.7.4. Programski jezik Python ima široku primenu. Upotrebljava se za kreiranje video-igara i razvoj web aplikacija. Python je softver otvorenog koda sa dobrom podrškom, dokumentacijom i bibliotekama i omogućava korisnicima pisanje preglednih programa.

## *Opis programa*

U seminarskom radu biće predstavljena aplikacija koja simulira vođenje evidencije u hotelu. Glavni korisnici su radnici koji rade za recepcijom. Oni dobijaju pristup tako što se pri pokretanju programa prijave unoseći korisničko ime i šifru. Oni imaju mogućnost dodavanja gosta, prikazivanja svih gostiju, produženja boravka određenog gosta, izmenene informacija gosta, pronalaženje gosta, prikazivanja računa, nalaženja gosta koji je najmanje ili najviše boravio u hotelu i prikazivanje grafika zavisnosti broja gostiju koji su se uselili odnosno iselili po mesecu

Tekstualni fajl koji se koristi u programu sadrži informacije o svakom gostu, a to su: broj sobe gosta, ime, prezime, e-mail, broj telefona, broj dana boravka, datum useljenja, datum useljenja i cena boravka. Postoji još jedan tekstualni fajl u kom se čuvaju podaci o radnicima gde se nalazi ime, prezime, korisničko ime i šifra svakog od radnika.

## 2. Opis programa

### 2.1 Klasa "Recepcija"

Ova klasa ima tri funkcije koje omogućavaju radniku na recepciji da se prijavi i koristi sve opcije koje se nalaze u glavnoj klasi programa.

Funkcija „stringToRec“ (slika 1.) služi za učitavanje informacija recepcije (korisničko ime i lozinku) i učitani string razdvaja na delove koji su u stringu spojeni znakom „|“. Zatim se dobijene vrednosti smeštaju u rečnik.

```
def stringToRec(line):  
    ime, prezime, username, password = line.split("|")  
    rec = {  
        "ime": ime,  
        "prezime": prezime,  
        "username": username,  
        "password": password  
    }  
    return rec
```

Slika 1. Metoda „stringToRec(...)“

Funkcija „login“ (slika 2.) proverava da li uneto korisničko ime i šifra postoje. Ako uneto korisničko ime i šifra postoje i ako se podudaraju sa korisničkim imenom i šifrom nekog od recepcionara iz tekstualnog fajla gde se oni nalaze, funkcija vraća vrednost true.

```
def login(username, password):  
    for rec in recepcionari:  
        if rec["username"] == username and rec["password"] == password:  
            return True  
    return False
```

Slika 2. Funkcija „login(...)“

Funkcija „loadRec“ (Slika 3.) učitava redove iz tekstualnog fajla gde se nalaze recepcionari.

```
def loadRec():
    for line in open("recepcija.txt", "r").readlines():
        if len(line) > 1:
            rec = stringToRec(line[:-1])
            recepcionari.append(rec)
```

Slika 3. Funkcija „loadRec()“

## 2.2 Klasa „Gosti“

Funkcija „gostToString“ (Slika 4.) podatke unete od strane recepcionara unosi u tekstualni fajl „gosti“

```
def gostToString(gost):
    return "|".join((gost["brSobe"],
                    gost["ime"],
                    gost["prezime"],
                    gost["email"],
                    gost["brTel"],
                    gost["brDana"],
                    gost["datum1"],
                    gost["datum2"],
                    gost["cenaS"]))
```

Slika 4. Funkcija „gostToString(...)“

Funkcija „StringToGost“ (Slika 5.) učitava podatke iz tekstualnog fajla i unosi ih u rečnik. Time se dobijaju informacije za svakog gosta.

```
def stringToGost(line):
    if line[-1] == '\n':
        line = line[:-1]
    brSobe, ime, prezime, mail, brTel, brDana, datum1, datum2, cenaS = line.split("|")
    gost = {
        "brSobe": brSobe,
        "ime": ime,
        "prezime": prezime,
        "email": mail,
        "brTel": brTel,
        "brDana": brDana,
        "datum1": datum1,
        "datum2": datum2,
        "cenaS": cenaS
    }
    return gost
```

Slika 5. Funkcija „StringToGost(...)“

Funkcija „loadGosti“ (Slika 6.) učitava redove iz tekstualnog fajla gde se čuvaju informacije od svakog gosta, zatim poziva funkciju „stringToGost“ i rečnik koji ova funkcija vraća se dodaje u listu gde se čuvaju podaci svakog gosta.

```
def loadGosti():  
    for line in open("gosti.txt", "r").readlines():  
        if len(line) > 1:  
            gost = stringToGost(line)  
            gosti.append(gost)
```

Slika 6. Funkcija „loadGosti()“

Funkcija „SaveGost“ (Slika 7.) čuva izmene koje je radnik uneo.

```
def saveGost():  
    file = open("gosti.txt", "w")  
    for gost in gosti:  
        file.write(gostToString(gost))  
        file.write("\n")  
    file.close
```

Slika 7. Funkcija „SaveGost()“

Funkcija „formatHeader“ (Slika 8.) formira zaglavlje koje se koristi pri prikazivanju liste gostiju. Zaglavlje sadrži broj sobe, ime gosta, prezime gosta, e-mail, broj telefona, dane boravka, datum useljenja, datum iseljenja i cena boravka.

```
def formatHeader():  
    print("Broj sobe |Ime      |Prezime  |E-mail      |Broj telefona |Dana boravka|Od dana   |Do dana   |Dugovanje € \n"  
          "-----+-----+-----+-----+-----+-----+-----+-----+-----")
```

Slika 8. Funkcija „formatHeader“

Funkcija „formatGost“ (Slika 9.) formira red u kojem će biti predstavljeni svi podaci o gostu i koji je u liniji sa zaglavljem iz funkcije „formatHeader“.

```
def formatGost(gost):  
    return "{0:10}|{1:12}|{2:12}|{3:30}|{4:15}|{5:12}|{6:12}|{7:12}|{8:12}".format(  
        gost["brSobe"],  
        gost["ime"],  
        gost["prezime"],  
        gost["email"],  
        gost["brTel"],  
        gost["brDana"],  
        gost["datum1"],  
        gost["datum2"],  
        gost["cenaS"]  
    )
```

Slika 9. Funkcija „formatGost (...)“

Funkcija „FormatGosti“ (Slika 10.) formira listu za prikazivanje.

```
def formatGosti(gosti):  
    lista = ""  
    for gost in gosti:  
        lista += formatGost(gost) + "\n"  
    return lista
```

Slika 10. Funkcija „FormatGosti(...)“

Funkcija „listGost“ (Slika 11.) izlistava sve goste u hotelu tako što vraća prethodno formatiranu listu.

```
def listGost():  
    return formatGosti(gosti)
```

Slika 11. Funkcija „listGost()“

Funkcija „addGost“ (Slika 12.) služi za dodavanje gosta u listu.

```
def addGost(gost):  
    gosti.append(gost)
```

Slika 12. Funkcija „addGost(...)“

Funkcija „ukupnaCena“ (Slika 13.) određuje cenu boravka u hotelu. Broj dana se množi sa cenom sobe, koja zavisi od broja sobe, tj. tipa sobe u kojoj gost boravi.

```
def ukupnaCena(broj, brDana):  
    if eval(broj) <= 9:  
        cena = 10  
    elif eval(broj) >= 10 and eval(broj) <=16:  
        cena = 15  
    else:  
        cena = 20  
    ukupno = cena * brDana  
    return ukupno
```

Slika 13. Funkcija „ukupnaCena(...)“

Funkcija „printRacun“ (Slika 14.) ispisuje cenu boravka za gosta koji se nalazi u sobi koju je recepcionar uneo. Ako u toj sobi ne boravi ni jedan gost, ispisuje se poruka koja to govori.

```
def printRacun():  
    brSobe = input("Unesite broj sobe za koju zelite da izdate racun >> ")  
    for gost in gosti:  
        if gost["brSobe"] == brSobe:  
            return print("Racun gosta " + gost["ime"] + " " + gost["prezime"] + " je " + gost["cenaS"] + "€")  
    print("Ne postoji gost u toj sobi.")  
    return printRacun()
```

Slika 14. Funkcija „printRacun“

Funkcija „addDan()“ (Slika 15.) omogućava recepcionaru da unese broj sobe gosta i da mu produži boravak za određen broj dana koji takođe može da unese. Prilikom produženja boravka, menja se cena boravka, broj dana boravka i datum iseljenja.

```
def addDan():  
    brSobe = input("Unesite broj sobe u kom se nalazi gost >> ")  
    brDana = input("Unesite za koliko dana zelite da produzite boravak >> ")  
    for gost in gosti:  
        if gost["brSobe"].upper() == brSobe.upper():  
            cenaS = int(int(gost["cenaS"]) / int(gost["brDana"]))  
            gost["cenaS"] = str(int(gost["cenaS"]) + int(cenaS) * int(brDana))  
            gost["brDana"] = str(int(gost["brDana"]) + int(brDana))  
            gost["datum2"] = unesiDatum2(gost["datum1"], int(gost["brDana"]))  
            return saveGost()
```

Slika 15. Funkcija „addDan()“



Funkcija „updateInfo“ (Slika 16.) omogućava recepcionaru da promeni informacije o gostu. Moguće je menjati broj telefona ili e-mail adresu. Recepcionar ima mogućnost izbora šta će menjati. Nakon promene informacije, potrebno je sačuvati promenu pomoću funkcije „saveGost“

```
def updateInfo(brSobe, key, value):
    for gost in gosti:
        if gost["brSobe"].upper() == brSobe.upper():
            gost[key] = value
            return saveGost()
    print("Ne postoji gost u toj sobi.")
```

Slika 16. Funkcija „updateInfo()“

Funkcija „unesiDatum1“ (Slika 17.) služi da se unese datum useljenja tipa string (u formatu d.m.g). Uneti datum se dodeljuje promenljivama dan, mesec i godina. Zatim se proverava da li je datum ispravno unet. Odnosno mora se uneti broj dana koji je veći od 0 i manji od 31 odnosno 30 u zavisnosti od meseca. U slučaju da je unet drugi mesec, broj dana ne sme biti veći od 28 dana, odnosno 29 kad je godina prestupna. Takođe meseci se mogu numerisati brojevima od 1 do 12, a godina ne sme biti manja od 2020.

```
def unesiDatum1():
    datum = input("Unesite datum useljenja >> ")
    dan, mesec, godina = datum.split(".")
    if dan < '1':
        print("Niste uneli ispravan datum.")
        return unesiDatum1()
    if mesec in ("1", "3", "5", "7", "8", "10", "12"):
        if int(dan) > 31:
            print("Niste uneli ispravan datum.")
            return unesiDatum1()
    elif int(mesec) == 2:
        if int(godina) % 4 == 0 and (int(godina) % 100 != 0 or int(godina) % 400 == 0):
            if int(dan) > 28:
                print("Niste uneli ispravan datum.")
                return unesiDatum1()
        else:
            if int(dan) > 27:
                print("Niste uneli ispravan datum.")
                return unesiDatum1()
    else:
        if int(dan) > 30:
            print("Niste uneli ispravan datum.")
            return unesiDatum1()
    if int(mesec) > 12 or int(mesec) < 0:
        print("Niste uneli ispravan datum.")
        return unesiDatum1()
    if int(godina) < 2020:
        print("Niste uneli ispravan datum.")
        return unesiDatum1()
    return datum
```

Slika 16. Funkcija „unesiDatum1()“

Funkcija „unesiDatum2“ (Slika 17.) služi da se gostu odredi datum iseljenja. Funkcioniše tako što se na datum useljenja doda broj dana koliko će gost boraviti u hotelu, vodeći računa o tome koji je mesec i da li datum iseljenja prelazi u sledeći mesec ili sledeću godinu.

```
def unesiDatum2(datum1, brDana):
    dan1, mesec1, godina1 = datum1.split(".")
    dan2 = str(int(dan1) + brDana)
    mesec2 = mesec1
    godina2 = godina1
    if mesec2 in ("1", "3", "5", "7", "8", "10", "12"):
        if int(dan2) > 31:
            mesec2 = str(int(mesec2) + 1)
            dan2 = str(int(dan2) - 31)
    elif int(mesec2) == 2:
        if (int(godina2) % 4 == 0 and (int(godina2) % 100 != 0 or int(godina2) % 400 == 0)):
            if int(dan2) > 29:
                mesec2 = str(int(mesec2) + 1)
                dan2 = str(int(dan2) - 28)
        else:
            if int(dan2) > 28:
                mesec2 = str(int(mesec2) + 1)
                dan2 = str(int(dan2) - 28)
    else:
        if int(dan2) > 30:
            mesec2 = str(int(mesec2) + 1)
            dan2 = str(int(dan2) - 30)
    if int(mesec2) > 12:
        mesec2 = str(int(mesec2) - 12)
        godina2 = str(int(godina2) + 1)
    datum2 = dan2 + "." + mesec2 + "." + godina2
    return datum2
```

Slika 17. Funkcija „unesiDatum2(...)“

Funkcija „printGrafik“ (Slika 18.) uzima broj useljenja i broj iseljenja koristeći funkciju „gostPoMesecu“ i prikazuje ih na Y-osi. X-osa se deli na 12 podeoka i svaki predstavlja po jedan mesec u godini. Postoji po dva stuba za svaki mesec, gde jedan prikazuje broj useljenja, drugi broj iseljenja. Imenuju se x i y osa i prikazuje legenda koja govori koji stub šta prikazuje. Za realizaciju ove funkcije bilo je potrebno prethodno importovati biblioteku „matplotlib“.

```
def printGrafik():
    x1 = np.arange(12)
    x2 = [x + 0.35/2 for x in x1]

    y1,y2 = gostPoMesecu()

    plt.bar(x1 - 0.35/2, y1, color='blue', width=0.35, edgecolor='white', label='Useljenje')
    plt.bar(x2, y2, color='red', width=0.35, edgecolor='white', label='Iseljenje')
    plt.xticks([r for r in range(12)], ["januar", "februar", "mart", "april", "maj", "jun", "jul", "avgust", "septembar", "oktobar", "novembar", "decembar"])
    plt.xlabel('meseci')
    plt.xticks(rotation=90)
    plt.ylabel('broj gostiju')
    if max(y1) > max(y2):
        maks = max(y1)
    else:
        maks = max(y2)
    plt.ylim(ymin=0, ymax=maks+1)
    plt.legend()
    plt.show()
```

Slika 18. Funkcija „printGrafik()“

Funkcija „gostPoMesecu“ (Slika 19.) za svaki mesec dodaje broj useljenja i broj iseljenja i sve to prosleđuje u dve liste.

```
def gostPoMesecu():
    januar = februar = mart = april = maj = jun = jul = avgust = septembar = oktobar = novembar = decembar = 0
    januar2 = februar2 = mart2 = april2 = maj2 = jun2 = jul2 = avgust2 = septembar2 = oktobar2 = novembar2 = decembar2 = 0
    for gost in gosti:
        dan, mesec, godina = gost["datum1"].split(".")
        dan2, mesec2, godina2 = gost["datum2"].split(".")
        if mesec == "1":
            januar += 1
        elif mesec == "2":
            februar += 1
        elif mesec == "3":
            mart += 1
        elif mesec == "4":
            april += 1
        elif mesec == "5":
            maj += 1
        elif mesec == "6":
            jun += 1
        elif mesec == "7":
            jul += 1
        elif mesec == "8":
            avgust += 1
        elif mesec == "9":
            septembar += 1
        elif mesec == "10":
            oktobar += 1
        elif mesec == "11":
            novembar += 1
        elif mesec == "12":
            decembar += 1

        if mesec2 == "1":
            januar2 += 1
        elif mesec2 == "2":
            februar2 += 1
        elif mesec2 == "3":
            mart2 += 1
        elif mesec2 == "4":
            april2 += 1
        elif mesec2 == "5":
            maj2 += 1
        elif mesec2 == "6":
            jun2 += 1
        elif mesec2 == "7":
            jul2 += 1
        elif mesec2 == "8":
            avgust2 += 1
        elif mesec2 == "9":
            septembar2 += 1
        elif mesec2 == "10":
            oktobar2 += 1
        elif mesec2 == "11":
            novembar2 += 1
        elif mesec2 == "12":
            decembar2 += 1

    gPmU = [januar, februar, mart, april, maj, jun, jul, avgust, septembar, oktobar, novembar, decembar]
    gPmI = [januar2, februar2, mart2, april2, maj2, jun2, jul2, avgust2, septembar2, oktobar2, novembar2, decembar2]
    return gPmU, gPmI
```

Slika 19. Funkcija „gostPoMesecu“

Funkcija „mostDays“ (Slika 20.) proverava koji gost je boravio najduže u hotelu. Ako ima više gostiju koji su boravili najduže u hotelu, ispisuje ih sve.

```
def mostDays():
    maxD = 0
    for gost in gosti:
        if int(gost["brDana"]) > maxD:
            maxD = int(gost["brDana"])
            g = gost
    g = []
    for gost in gosti:
        if int(gost["brDana"]) == maxD:
            g.append(gost)
    if len(g) == 1:
        for gost in g:
            print("Gost " + gost["ime"] + " " + gost["prezime"] + " je boravio najvise dana u hotelu")
    else:
        print("Gosti koji su boravili najduze u hotelu su:")
        for gost in g:
            print(gost["ime"] + " " + gost["prezime"])
```

Slika 20. Funkcija „mostDays()“

Funkcija „leastDays“ (Slika 21.) proverava koji gost je boravio najkraće u hotelu. Ako ima više gostiju koji su boravili najkraće u hotelu, ispisuje ih sve.

```
def leastDays():
    minD = 100
    for gost in gosti:
        if int(gost["brDana"]) < minD:
            minD = int(gost["brDana"])
    g = []
    for gost in gosti:
        if int(gost["brDana"]) == minD:
            g.append(gost)
    if len(g) == 1:
        for gost in g:
            print("Gost " + gost["ime"] + " " + gost["prezime"] + " je boravio najmanje dana u hotelu")
    else:
        print("Gosti koji su boravili najmanje u hotelu su:")
        for gost in g:
            print(gost["ime"] + " " + gost["prezime"])
```

Slika 21. Funkcija „leastDays()“

Funkcija „dodeliSobu“ (Slika 22.) dodeljuje nasumično sobu u zavisnosti da li gost traži jednokrevetnu, dvokrevetnu ili trokrevetnu sobu. Za realizaciju ove funkcije bilo je potrebno prethodno importovati biblioteku „random“.

```
def dodeliSobu():  
    print("Izaberite sobu: ")  
    print("1 - jednokrevetna")  
    print("2 - dvokrevetna")  
    print("3 - trokrevetna")  
    soba = input("Izbor >> ")  
    if soba == '1':  
        broj = random.randint(1, 9)  
    elif soba == '2':  
        broj = random.randint(9,16)  
    else:  
        broj = random.randint(16, 20)  
    return str(broj)
```

Slika 22. Funkcija „dodeliSobu()“

## 2.3 Klasa „GlavniProgram“

U glavnoj klasi se pozivaju funkcije iz prethodnih klasa (Slika 23.). Na samom početku klase importujemo klasu „Recepcija“ i klasu „Gosti“ kako bi mogli da pozivamo sve metode iz ovih klasa.

```
import Recepcija
import Gosti
```

Slika 23. Importovanje klase „Recepcija“ i klase „Gosti“

Funkcija „main“ (Slika 24.) proverava da li rezultat funkcije „login“ daje rezultat true. Zatim proveravamo da li je komanda jednaka broju 1,2,3,4,5,6,7,8 ili 9 sve dok vrednost komande nije jednaka X. Svaki od brojeva poziva jednu funkciju. U slučaju da funkcija „login“ daje vrednost false, ponovo će biti omogućeno prijavljivanje recepcionara. Takođe, ukoliko se ne unese X ili neki broj od 1 do 9, biće ponovo omogućen unos.

```
def main():
    print()
    print( "Evidencija gostiju")
    print( "=====")
    print()
    if not login():
        print("\nNiste uneli dobro korisnicko ime ili lozinku.")
        print()
        main()
    komanda = '0'
    while komanda != 'X':
        komanda = menu()
        if komanda == '1':
            addGost()
        elif komanda == '2':
            listGost()
        elif komanda == '3':
            addDan()
        elif komanda == '4':
            updateInfo()
        elif komanda == '5':
            findGost()
        elif komanda == '6':
            printRacun()
        elif komanda == '7':
            mostDays()
        elif komanda == '8':
            leastDays()
        elif komanda == '9':
            printGrafik()
    return print( "Dovidjenja.")
```

Slika 24. Funkcija „main()“

Funkcija „login“ (Slika 25.) učitava korisničko ime i lozinku i poziva metod „login“ kome prosleđuje parametre koji su prethodno uneti.

```
def login():
    username = input("Korisnicko ime >> ")
    password = input("Lozinka >> ")
    return Recepcija.login(username, password)
```

Slika 25. Funkcija „login()“

Funkcija „menu“ (Slika 26.) poziva funkciju „printMenu“, zatim se unosi komanda i vrši provera da li je uneta odgovarajuća komanda. Ako jeste, komanda se prosleđuje dalje.

```
def menu():
    printMenu()
    command = input(">> ")
    while command.upper() not in ("1", "2", "3", "4", "5", "6", "7", "8", "9", "X"):
        print("\nUneli ste pogresnu komandu")
        printMenu()
        command = input(">> ")
    return command.upper()
```

Slika 26. Funkcija „menu()“

Funkcija „printMenu“ (Slika 27.) prikazuje opcije koje ima recepcioner.

```
def printMenu():
    print("\nIzaberite opciju: ")
    print("1 - Dodaj gosta: ")
    print("2 - Prikaz svih goostiju: ")
    print("3 - Produzenje boravka: ")
    print("4 - Izmena informacija: ")
    print("5 - Pronadji gosta: ")
    print("6 - Prikaz racuna: ")
    print("7 - Gost koji je najvise boravio u hotelu: ")
    print("8 - Gost koji je najmanje boravio u hotelu: ")
    print("9 - Broj gostiju po mesecu: ")
    print("x - Izlaz ")
```

Slika 27. Funkcija „printMenu()“



Funkcija „addGost“ (Slika 28.) služi za dodavanje novog gosta u listu.

```
def addGost():
    print("[1] Upis novog gosta\n")
    print()
    g = {}
    g["brSobe"] = str(len(open("gosti.txt", "r").readlines())+1)
    g["ime"] = input("Ime >> ")
    g["prezime"] = input("Prezime >> ")
    g["email"] = input("mail >> ")
    g["brTel"] = input("Broj telefona >> ")
    brDana = eval(input("Broj dana boravka >> "))
    g["brDana"] = str(brDana)
    datum1 = Gosti.unesiDatum1()
    g["datum1"] = datum1
    datum2 = Gosti.unesiDatum2(datum1, brDana)
    g["datum2"] = datum2
    g["brSobe"] = Gosti.dodeliSobu()
    g["cenaS"] = str(Gosti.ukupnaCena(g["brSobe"], brDana))
    Gosti.addGost(g)
    Gosti.saveGost()
```

Slika 28. Funkcija „addGost()“

Funkcija „listGost“ (Slika 29.) izlistava sve goste u hotelu.

```
def listGost():
    print("[2] Lista svih gostiju:\n")
    print(Gosti.foramtHeader())
    print(Gosti.listGost())
```

Slika 29. Funkcija „listGost()“

Funkcija „addDan“ (Slika 30.) poziva metod „addDan“ iz klase „Gosti“.

```
def addDan():
    print("[5] Produzenje boravka:\n")
    Gosti.addDan()
```

Slika 30. Funkcija „addDan()“

Funkcija „updateInfo“ (Slika 31.) menja podatke gosta, u ovom slučaju broj telefona i e-mail adresu. Ovo se postiže pozivanjem funkcije „updateInfo“ iz klase „Gosti“.

```
def updateInfo():
    print("[4] Izmena informacija:\n")
    brSobe = input("Unesite broj sobe u kom se nalazi gost >> ")
    print("Da li zelite da menjate e-mail ili broj telefona (uneti [email/brTel]) ?")
    kljuc = input(">> ")
    while kljuc not in ("email", "brTel"):
        print("\nUneli ste pogresnu komandu")
        kljuc = input(">> ")
    vrednost = input("Unesite novu vrednost >> ")
    Gosti.updateInfo(brSobe, kljuc, vrednost)
```

Slika 31. Funkcija „updateInfo()“

Funkcija „findGost“ (Slika 32.) poziva metodu „findGost“ iz klase „Gosti“ pronalazi gosta po broju sobe i ispisuje ga.

```
def findGost():
    print("[5] Pronadji gosta po broju sobe:\n")
    brSobe = input("Unesite broj sobe >> ")
    gost = Gosti.findGost(brSobe)
    if len(gost) == 0:
        print("Nema gosta u toj sobi")
    else:
        print(Gosti.foramtHeader())
        print(Gosti.formatGost(gost))
```

Slika 32. Funkcija „findGost()“

Funkcija „printRacun“ (Slika 33.) poziva metodu „printRacun“ iz klase „Gosti“ i ispisuje vrednost koju metoda iz klase „Gosti“ prosleđuje.

```
def printRacun():
    print("[6] Prikazi racun:\n")
    print(Gosti.printRacun())
```

Slika 33. Funkcija „printRacun()“

Funkcija „printGrafik“ (Slika 34.) poziva metod „printGrafik“ iz klase „Gosti“ i prikazuje grafik zavisnosti useljenja/iseljenja u odnosu na vreme (na mesečnom nivou).

```
def printGrafik():
    print("[9] Broj gostiju po mesecu:\n")
    Gosti.printGrafik()
```

Slika 34. Funkcija „printGrafik“

Funkcije „mostDays“ i „leastDays“ (Slika 35.) pozivaju metode „printGrafik“ i „mostDays“ iz klase „Gosti“ i prikazuju ko je najduže boravio, odnosno najkraće u hotelu.

```
def mostDays():
    print("[7] Gost koji je najvise boravio u hotelu:\n")
    Gosti.mostDays()

def leastDays():
    print("[7] Gost koji je najmanje boravio u hotelu:\n")
    Gosti.leastDays()
```

Slika 35. Funkcije „mostDays“ i „leastDays“

## 2.3 Kako sve to izgleda ?

```
__main__
Evidencija gostiju
=====

Korisnicko ime >> dejan

Lozinka >> 123

Izaberite opciju:
1 - Dodaj gosta:
2 - Prikaz svih gostiju:
3 - Produzenje boravka:
4 - Izmena informacija:
5 - Pronadji gosta:
6 - Prikaz racuna:
7 - Gost koji je najvise boravio u hotelu:
8 - Gost koji je najmanje boravio u hotelu:
9 - Broj gostiju po mesecu:
x - Izlaz

>>
```

Slika 35. Login i prikaz svih operacija

```

Ime >> Marko

Prezime >> Markovic

mail >> marko.markovic@gmail.com

Broj telefona >> 065

Broj dana boravka >> 10

Unesite datum useljenja >> 10.10.2019
Izaberite sobu:
1 - jednokrevetna
2 - dvokrevetna
3 - trokrevetna

Izbor >> 2

```

Slika 36. Dodavanje gosta

[2] Lista svih gostiju:

Broj sobe	Ime	Prezime	E-mail	Broj telefona	Dana boravka	Od dana	Do dana	Dugovanje €
None								
11	Marko	Markovic	marko.markovic@gmail.com	065	10	10.10.2019	20.10.2019	150
1	Nikola	Nikolic	nikola.nikolic@gmail.com	061	13	28.1.2019	10.2.2019	130
16	Branko	Brankovic	branko.brankovic@gmail.com	060	7	29.12.2019	5.1.2020	105
4	Ana	Antic	ana.antic@gmail.com	066	23	20.2.2019	15.3.2019	230
9	Marina	Matic	marina.matic@gmail.com	065	15	17.5.2019	1.6.2019	150

Slika 37. Prikaz liste svih gostiju

```
Unesite broj sobe u kom se nalazi gost >> 4
```

```
Unesite za koliko dana zelite da produzite boravak >> 5
```

```

4      |Ana      |Antic      |ana.antic@gmail.com      |066      |28      |20.2.2019      |20.3.2019      |280

```

Slika 38. Produženje boravka

Unesite broj sobe u kom se nalazi gost >> 9  
Da li zelite da menjate e-mail ili broj telefona (uneti [email/brTel]) ?

>> brTel

Unesite novu vrednost >> 064

9	Marina	Matic	marina.matic@gmail.com	064	15	17.5.2019	1.6.2019	150
---	--------	-------	------------------------	-----	----	-----------	----------	-----

Slika 39. Izmena informacija

>> 5  
[5] Pronadji gosta po broju sobe:

Unesite broj sobe >> 16

Broj sobe	Ime	Prezime	E-mail	Broj telefona	Dana boravka	Od dana	Do dana	Dugovanje €
None								
16	Branko	Brankovic	branko.brankovic@gmail.com	060	7	29.12.2019	5.1.2020	105

Slika 40. Pronalaženje određene osobe po broju sobe

>> 6  
[6] Prikazi racun:

Unesite broj sobe za koju zelite da izdate racun >> 11  
Racun gosta Marko Markovic je 150€  
..

Slika 41. Prikaz računa

>> 7  
[7] Gost koji je najvise boravio u hotelu:

Gost Ana Antic je boravio najvise dana u hotelu

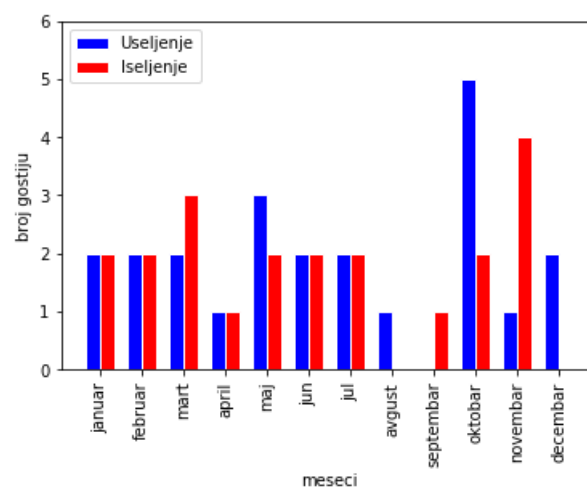
Slika 42. Prikaz gosta koji je najduže boravio u hotelu

```
>> 8
[7] Gost koji je najmanje boravio u hotelu:

Gosti koji su boravili najmanje u hotelu su:
Branko Brankovic
Petra Petric
```

Slika 43. Prikaz gostiju koji su najkraće boravili u hotelu

```
>> 9
[9] Broj gostiju po mesecu:
```



Slika 44. Prikaz grafika

### 3. Zaključak

U projektu predstavljena je aplikacija za vođenje evidencije u hotelu. Ovaj program uz nadogradnju se može primeniti u stvarnom životu zbog potrebe radnika da unose i menjaju informacije o gostima. Projekat je moguće unaprediti dodavanjem metoda za:

- proveru da li je neka soba slobodna u periodu kad se useljava novi gost
- brisanje/odjavu gostiju
- mogućnost povećanja cene u slučaju da gost želi dodatne usluge kao što su korišćenje namirnica iz frižidera i slično

## 4. Literatura

1. <https://www.w3schools.com/python/>
2. <https://python-graph-gallery.com/matplotlib/>