

Пројекти у Python и R програмским језицима са применом постојећих Excel табела и база података

Миљан Јеремић, Милан Гоцић

Циљеви предавања:

- Упознати се са основним окружењем програмског језика R, његовом синтаксом основним могућностима језика R у области статистичке и графичке анализе података, и коришћењем података записаних у Excel табелама и базама података.
- Упознати се са основним окружењем програмског језика Python, његовом синтаксом и његовим додатним библиотекама за инсталирање и коришћењем података записаних у базама података.

Ученицима осмих разреда и других разреда средњих школа се на часовима објашњава детаљно принцип креирања Excel табела, како на свом рачунару, тако сада и помоћу апликација које су на интернету где вам је потребна само конекција са мрежом и без инсталирања било каквог пакета програма као што је Microsoft Office, Open Office или Libre Office.

План је да се у интерактивном раду са наставницима они упознају са софтверима које могу да искористе у настави, и да на један креативан начин прикажу ученицима принцип програмирања и приступа Excel документима и повезивање са базама података у Access-у и SQLite3 или неком другом програмском окружењу у коме је база података већ направљена.

У другом делу предавања, биће показано како применом програмских језика Python и R могу да се искористе претходно пројектоване базе података и да се постигне интеракција у програмирању између програмских језика и Excel докумената и база података.

Планирано је да се ураде неколико апликација које приказују рад са подацима из Excel табела и база података у конзолном и ГУИ окружењу у програмском језику Python, и да се на основу података у бази изврши детаљна анализа података применом R програмског језика, који од ове школске године може да се користи у редовној настави за други разред у гимназији.

R (програмски језик)

<http://www.r-project.org/>

Иако R постоји отприлике 20 година, у рачунарству, R је програмски језик и програмско окружење за статистичка израчунавања и омогућава да манипулишете и графички прикажете податке. Он је изведен из S програмског језика са лексичком семантиком инспирисаном програмским језиком Scheme. R су створили Ros Ihaka и Robert Džentlmen на Aukland универзитету (University of Auckland), Нови Зеланд, а сад га развија R Development Core Team. Назван је делимично према именима аутора.

R језик је постао стандард међу статистичарима који развијају статистички софтвер, и широко је коришћен за развој статистичког софтвера и анализу података. R је статистички оријентисан језик популаран међу „рударима“ података.

R је део GNU пројекта. Његов изворни код је слободан и под условима које даје GNU-ова општа јавна лиценца, а прекомпилиране бинарне верзије су обезбеђене за различите оперативне системе. R користи интерфејс командне линије, кроз више графичких корисничких окружења.

Python

Python је програмски језик опште намене, интерпретиран и високог нивоа који је створио Guido van Rossum 1990. године. Python допушта програмерима коришћење неколико стилова програмирања. Објектно оријентисано и структурно програмирање стилови су допуштени коришћењем Python-а и ова флексибилност чини програмски језик Python све популарнијим. Python се највише користи на Linux-у, али постоје и верзије за друге оперативне системе.

Иако се R и SAS најчешће називају "велика два" у свету аналитике, Python је недавно постао кандидат. Једна од његових главних карактеристика је широк спектар библиотека (нпр. Pandas, NumPy, SciPi, итд.) и статистичке функције.

Будући да је Python (као R) језик отвореног кода, ажурирања се брзо додају.

Други фактор који треба узети у обзир јесте да је Python можда најлакши за учење, због једноставности и широке доступности курсева и ресурса.

SQL

До сада смо гледали језике који су у истој породици и (више или мање) имају исте функције. SQL, који означава "Структурни језик упита", нема никакве везе са статистиком; фокусира се на руковање информацијама у релационим базама података.

То је најраспрострањенији језик базе података и отворен је извор.

Учење SQL-а треба да вам омогући креирање SQL база података, управљање подацима унутар њих и коришћење релевантних функција.

Основне SQL команде су дате без детаљних објашњења:

CREATE TABLE - креирање табеле у бази података,
DROP TABLE - уклањање табеле из базе података,
ALTER TABLE - измена дефиниције табеле,
CREATE INDEX - креирање индекса,
DROP INDEX - уклањање индекса,
CREATE VIEW - креирање погледа,
DROP VIEW - уклањање погледа,
SELECT - приказ жељеног садржаја из једне или више табела,
UPDATE - измена вредности колона табеле,
DELETE - брисање редова из табеле,
INSERT - додавање редова у табелу,
GRANT - додељивање овлашћења над објектима базе другим корисницима од стране власника,
REVOKE - укидање овлашћења додељених помоћу функције GRANT.

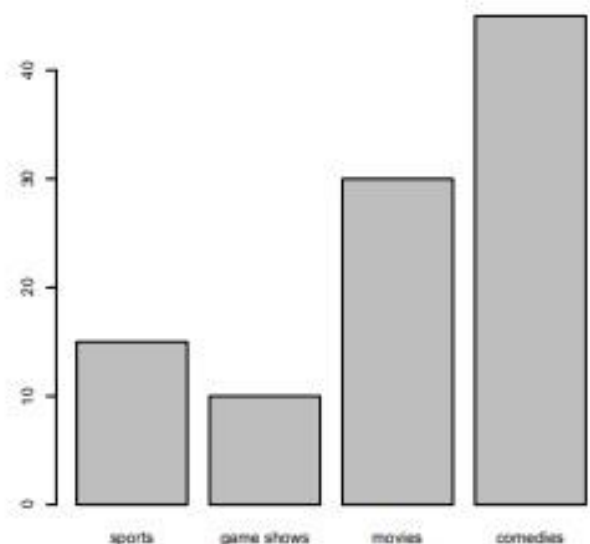
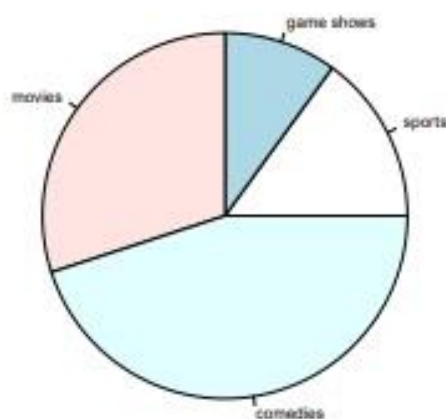
Увод у визуелизацију података у R програмском језику

Графички подаци

Следе примери кружних и стубичастих дијаграма. У првом примеру, измерено је време које је особа провела гледајући 4 различите врсте ТВ емисија. 15% времена је потрошено на спорт, 10% на емисије, 30% на филмове и 45% на комедије.

Прво је потребно да поставите објекат који садржи информације које ће се исцртати. Овде се објекту званом tv додељују потребне информације

```
tv <- c("sports" = 15, "game shows" = 10, "movies" = 30, "comedies" = 45)
pie(tv)
```

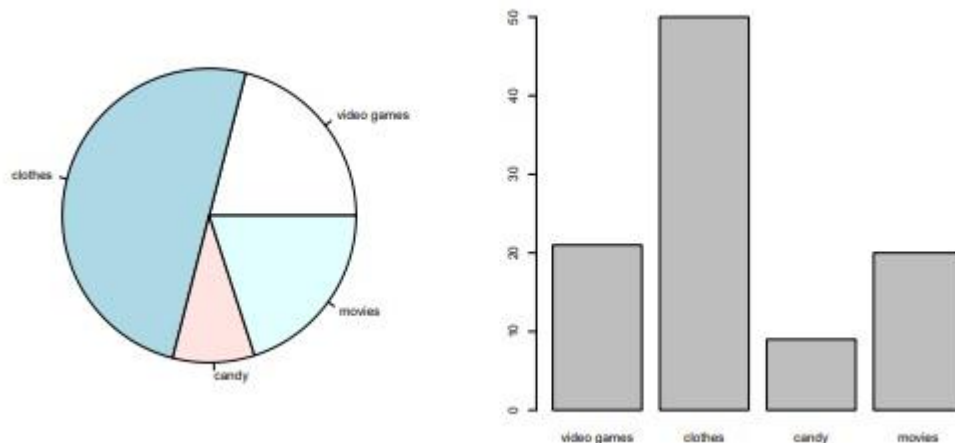


Функција barplot се користи за креирање стубичастог графика:

```
barplot(tv)
```

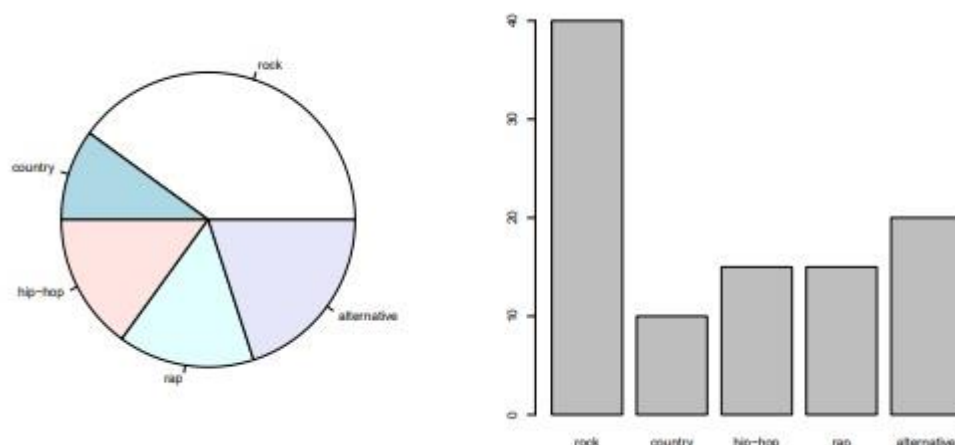
Други пример: Претпоставимо да вам је на располагању буџет од 100 долара који ћете потрошити на четири различите ствари: 21 долар се троши на видео игре, 50 долара на одећу, 9 на слаткише и 20 на филмове. Кружни и стубичасти графикон су приказани у наставку.

```
budget <- c("video games" = 21, "clothes" = 50, "candy" = 9, "movies" = 20)
pie(budget)
barplot(budget)
```



Следећи пример показује рашчлањивање музичких преференција за разред ученика седмог разреда: 40% преферира рок, 10% кантри, 15% хип-хоп, а остали алтернативу.

```
music <- c("rock" = 40, "country" = 10, "hip-hop" = 15, "rap" = 15, "alternative" = 20)
pie(music)
barplot(music)
```

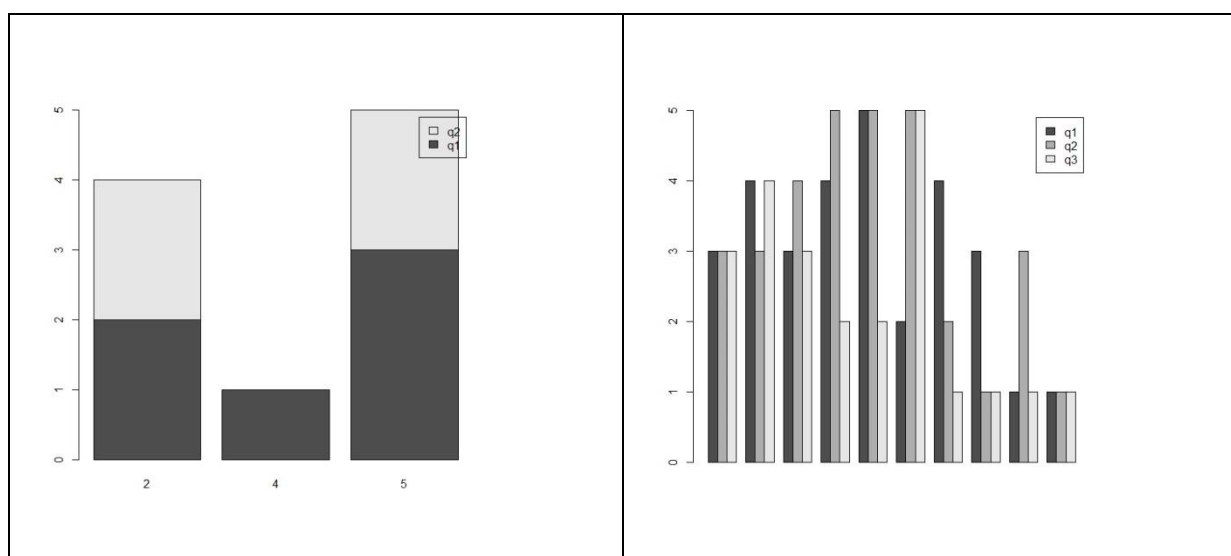


Пример са цртањем стубичастих графикона, где су на слици дате оцене којима су ученици оценили професоре на скали од 1 до 5. На спојеном стубичастом графикону представити одговоре на прво и друго питање, а на графикону са одвојеним ступцима представити одговоре на сва три питања.

```
q1 = c(3, 3, 3, 4, 3, 4, 3, 4, 3, 4)
q2 = c(5, 2, 5, 5, 2, 2, 5, 5, 4, 2)
q3 = c(1, 3, 1, 1, 1, 3, 1, 1, 1, 1)
```

```
barplot(table(q1, q2), legend.text = c("q1", "q2"))
barplot(matrix(data = c(q1, q2, q3), nrow = 3), beside = T,
legend.text = c("q1", "q2", "q3"))
```

Student	Ques. 1	Ques. 2	Ques. 3
1	3	5	1
2	3	2	3
3	3	5	1
4	4	5	1
5	3	2	1
6	4	2	3
7	3	5	1
8	4	5	1
9	3	4	1
10	4	2	1



R и Excel табеле

CSV датотеке

- Употребом функције **getwd()** могуће је добити информацију о радном директоријуму на који показује R окружење.
- Употребом функције **setwd()** могуће је променити радни директоријум који R окружење користи.

```
data <- read.csv("input.csv")
print(data)
```

	id,	name,	salary,	start_date,	dept
1	1	Rick	623.30	2012-01-01	IT
2	2	Dan	515.20	2013-09-23	Operations
3	3	Michelle	611.00	2014-11-15	IT
4	4	Ryan	729.00	2014-05-11	HR
5	NA	Gary	843.25	2015-03-27	Finance
6	6	Nina	578.00	2013-05-21	IT
7	7	Simon	632.80	2013-07-30	Operations
8	8	Guru	722.50	2014-06-17	Finance

```
data <- read.csv("input.csv")
print(is.data.frame(data))
print(ncol(data))
print(nrow(data))
```

```
[1] TRUE
[1] 5
[1] 8
```

Обрада података

Пример како наћи минималну и максималну плату у предузећу.

```
# Kreiranje okvira podataka.
data <- read.csv("input.csv")

# Get the max salary from data frame.
sal <- min(data$salary)
print(sal)
```

```
[1] 515.2
```

```
# Kreiranje okvira podataka iz CSV datoteke.
data <- read.csv("input.csv")

# Pretraga za osobom koja ima najveću platu.
retval <- subset(data, salary == max(salary))
print(retval)
```

	id	name	salary	start_date	dept
5	NA	Gary	843.25	2015-03-27	Finance

Снимање CSV датотеке

```
# Kreiranje okvira podataka.  
data <- read.csv("input.csv")  
retval <- subset(data, as.Date(start_date) > as.Date("2014-01-01"))  
  
# Pisanje u CSV datoteku filtriranih podataka.  
write.csv(retval, "output.csv")  
newdata <- read.csv("output.csv")  
print(newdata)
```

	X	id	name	salary	start_date	dept
1	3	3	Michelle	611.00	2014-11-15	IT
2	4	4	Ryan	729.00	2014-05-11	HR
3	5	NA	Gary	843.25	2015-03-27	Finance
4	8	8	Guru	722.50	2014-06-17	Finance

R и базе података

R може директно да манипулише и са базама података, са Excel фајловима итд.

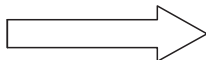
С обзиром на то да се у економским анализама често полази од великог броја података, самостално генерисање фајла са подацима понекад може бити компликовано и временски захтевно. Зато је препоручљиво уредити скуп података у програму Excel. Зависно од верзије Excel и R програма, постоји неколико одговарајућих начина уноса података у R.

Примера ради, корисници 32-битног R програма могу унети податке у R применом пакета RODBC. Следећи низ наредби инсталираће потребни пакет и учитати податке из Excel датотеке „Biti zdrav.xlsx” (лист под називом I razred) у променљиву која чува податке podaci_No1.

```
install.packages("RODBC")  
library(RODBC)  
channel<-odbcConnectExcel("Biti zdrav.xlsx")  
podaci_No1<-sqlFetch(channel,"list1")  
odbcClose(channel)
```

Дакле, наредба odbcConnectExcel покреће протокол ODBC (engl. Open Database Connectivity) који повезује програм R са Excel датотеком од интереса (у овом случају Biti_zdrav.xlsx). Наредба sqlFetch повезује програм са траженом листом у одабраној Excel датотеци (у овом случају I razred), док odbcClose(channel) прекида везу са ODBC протоколом.

	A	B
1	x	y
2	34	7
3	24	8



```
  x y  
1 34 7  
2 24 8  
> |
```

Пример повезивања Access базе података са R програмским језиком: База података се односи на евиденцију ученика и професора у школи. Треба напоменути да се овај програм покреће под 32-битном верзијом програма R.

```
library(RODBC)
```

```
db <- "C:/Users/PavleiMila/Documents/baza IV3.accdb"  
con2 <- odbcConnectAccess2007(db)
```

```
sqlTables(con2, tableType = "Profesor")
```

```
school <- sqlFetch(con2, "Profesor1")
```

```
str(school)
```

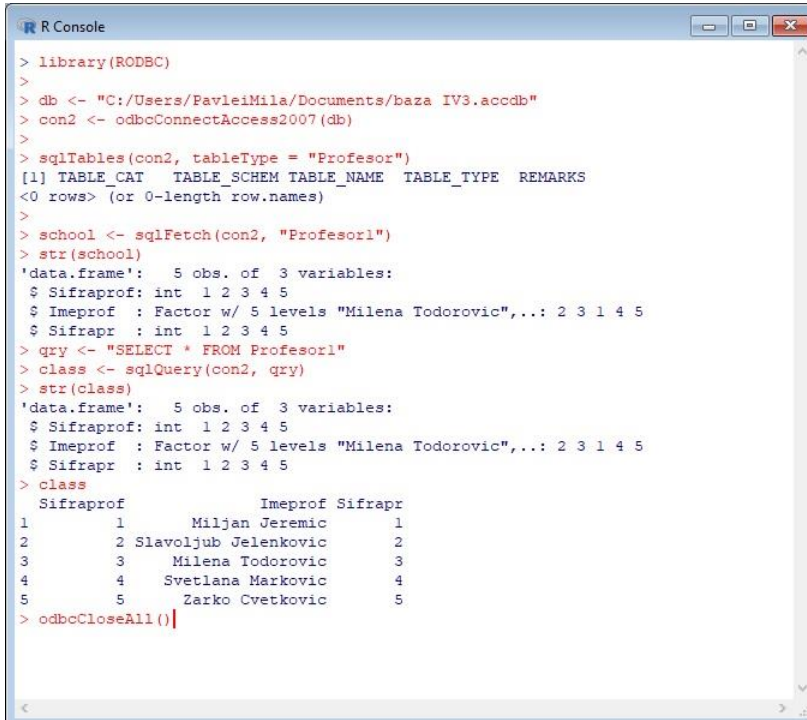
```
qry <- "SELECT * FROM Profesor1"
```

```
class <- sqlQuery(con2, qry)
```

```
str(class)
```

```
class
```

```
odbcCloseAll()
```

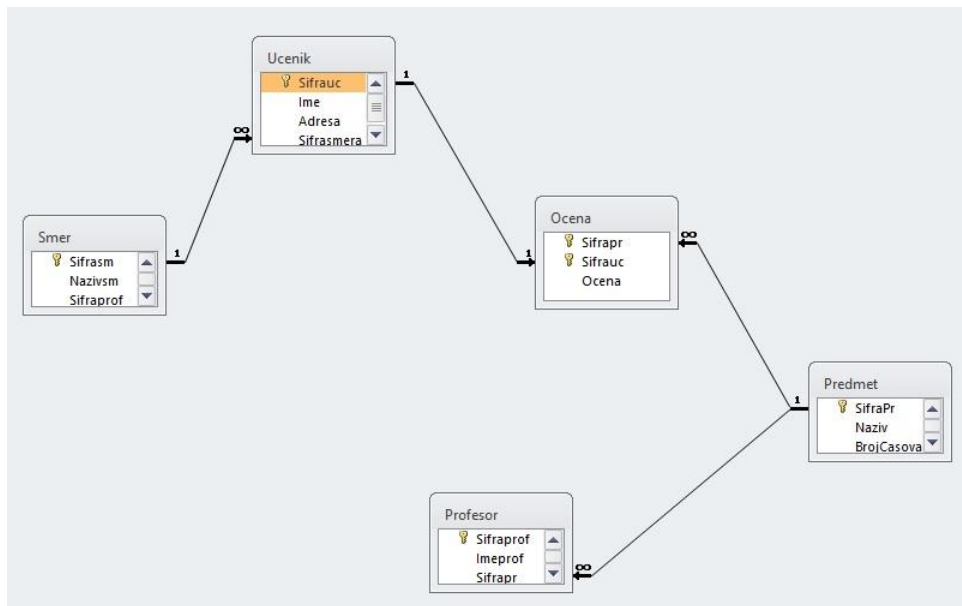


```
> library(RODBC)  
>  
> db <- "C:/Users/PavleiMila/Documents/baza IV3.accdb"  
> con2 <- odbcConnectAccess2007(db)  
>  
> sqlTables(con2, tableType = "Profesor")  
[1] TABLE_CAT TABLE_SCHEM TABLE_NAME TABLE_TYPE REMARKS  
<0 rows> (or 0-length row.names)  
>  
> school <- sqlFetch(con2, "Profesor1")  
> str(school)  
'data.frame': 5 obs. of 3 variables:  
 $ Sifraprof: int 1 2 3 4 5  
 $ Imeprof : Factor w/ 5 levels "Milena Todorovic",...: 2 3 1 4 5  
 $ Sifrapr : int 1 2 3 4 5  
> qry <- "SELECT * FROM Profesor1"  
> class <- sqlQuery(con2, qry)  
> str(class)  
'data.frame': 5 obs. of 3 variables:  
 $ Sifraprof: int 1 2 3 4 5  
 $ Imeprof : Factor w/ 5 levels "Milena Todorovic",...: 2 3 1 4 5  
 $ Sifrapr : int 1 2 3 4 5  
> class  
 Sifraprof Imeprof Sifrapr  
1 1 Miljan Jeremic 1  
2 2 Slavoljub Jelenkovic 2  
3 3 Milena Todorovic 3  
4 4 Svetlana Markovic 4  
5 5 Zarko Cvetkovic 5  
> odbcCloseAll()
```

Пример повезивања MySQL базе података користећи програмски језик R где је потребно креирати конекцију на базу података (пакети DBI и RMySQL), и учитати табелу ученици у свој data frame. За овако учитане податке потребно је:

- Прегледати сваки data frameи утврдити информације које би могле бити корисне
- Идентификовати податке који се налазе у табели ученици.

Шема базе података је приказана на слици.



```
install.packages("RMySQL")
library(RMySQL)
```

```
mydb = dbConnect(MySQL(), user='root', password='', dbname='ednevnikmgj',
host='localhost')
```

```
dbListTables(mydb)
```

```
rs = dbSendQuery(mydb, 'SELECT * FROM ucenici')
```

```
data = fetch(rs, n=-1)
```

```

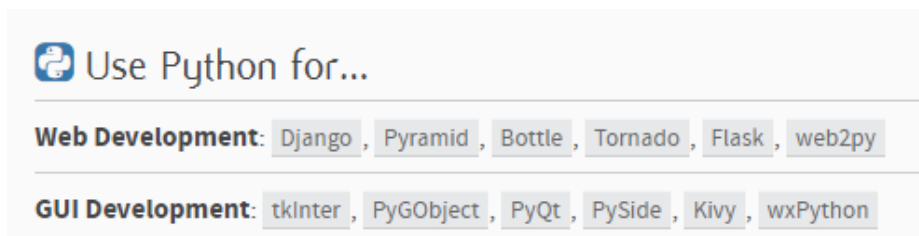
R Console
> install.packages("RMySQL")
Installing package into 'C:/Users/PavleiMila/Documents/R/win-library/3.6'
(as 'lib' is unspecified)
Warning: package 'RMySQL' is in use and will not be installed
> library(RMySQL)
>
> mydb = dbConnect(MySQL(), user='root', password='', dbname='ednevnikmgj', hos$
>
> dbListTables(mydb)
[1] "podaci" "ucenici"
>
> rs = dbSendQuery(mydb, 'SELECT * FROM ucenici')
>
> data = fetch(rs, n=-1)
> data
  id FirstName      LastName Srpski Engleski Nemacki Francuski Latinski Istorija
1 30 Miljan        Jeremic      5        5        5         0         5         5
2 1 Nikola Djurdjanovic 4        4        3         4         4         4
  Geografija Biologija Matematika Fizika Hemija Informatika Likovno Fizicko
1      5      5      5      5      5      5      5      5
2      4      4      4      4      4      4      4      4
  Muzicko Gradjansko Verska Vladanje
1      5      vu      5
2      5      vu      5
> |

```

Графички кориснички интерфејс и програмирање у Python-у

Python пружа различите могућности за развој графичког корисничког интерфејса (GUI). Постоји више алата за рад у графичком окружењу.

Tkinter



Бојење позадине прозора

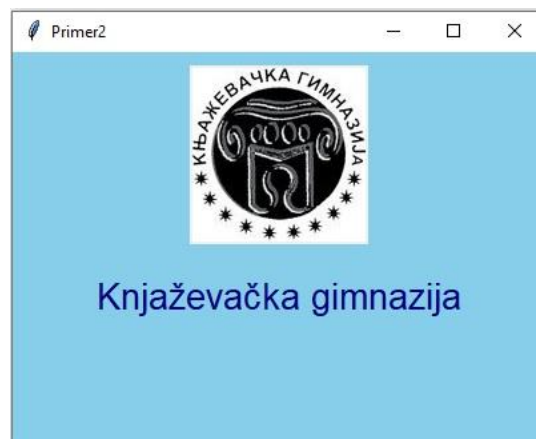
Постављање боје на позадину прозора врши се на следећи начин: `p.config(bg = 'blue')`
Методом `config()` можемо да мењамо параметре `widgeta`.

```
# Primer broj 1
import tkinter
p = tkinter.Tk()
p.geometry('400x300')
p.title('Naslov')
p.wm_iconbitmap('facebook.ico')
p.config(bg = 'blue')
p.mainloop()
```



Компонента Label - пример

```
# Primer broj 2
import tkinter
p = tkinter.Tk()
p.geometry('400x300')
p.title('Primer2')
p.config(bg = 'sky blue')
s = tkinter.PhotoImage(file = 'KG.png')
l = tkinter.Label(p, image = s)
l1 = tkinter.Label(text = 'Knjaževačka gimnazija', fg =
'navy',bg = 'sky blue', font = ('Arial','20'))
l.pack(padx = 10, pady = 10)
l1.pack(padx = 10, pady = 10)
p.mainloop()
```



Tkinter прихвата само слике које имају екстензију `png` или `gif`.

Компонента Entry - пример

```
import tkinter as tk

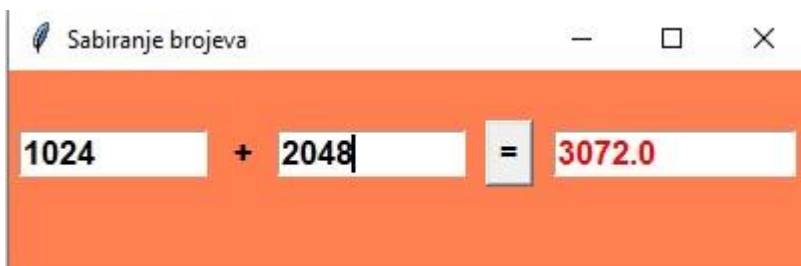
p = tk.Tk()
p.geometry('400x100')
p.title('Sabiranje brojeva')
p.config(bg = 'coral')

ff = font('Arial', 12, 'bold')

def racunaj():
    a = float(e1.get())
    b = float(e2.get())
    s = a + b
    e3.delete(0, len(e3.get()))
    e3.insert(0, str(s))

e1 = tk.Entry(p, font = ff, width = 10)
l1 = tk.Label (p, text = '+', bg = 'coral', font = ff)
e2 = tk.Entry(p, font = ff, width = 10)
b1 = tk.Button(p, text = '=', command = racunaj, font = ff)
e3 = tk.Entry(p, fg = 'red', font = ff, width = 13)

e1.grid(column = 1, row = 1, padx = 5, pady = 25)
l1.grid(column = 2, row = 1, padx = 5, pady = 25)
e2.grid(column = 3, row = 1, padx = 5, pady = 25)
b1.grid(column = 4, row = 1, padx = 5, pady = 25)
e3.grid(column = 5, row = 1, padx = 5, pady = 25)
p.mainloop()
```



Како повезати Python са MS Access базом података помоћу pyodbc-a

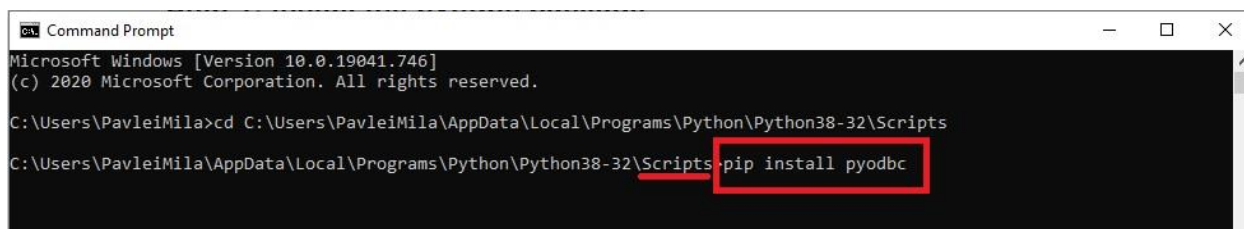
Овде желимо да прикажемо кораке за успостављање ове врсте везе од почетка.

Такође, објаснићемо како да решите уобичајене грешке приликом покушаја повезивања Pythona са Access-ом.

Кораци приликом повезивања Python-а и MS Access коришћењем pyodbc.

1: Инсталирање pyodbc package

На почетку, уз помоћу Command Prompt-а, инсталирати *pyodbc* пакет који ће се користити као конекција Python-а са Access базом података. Користити [PIP install method](#) како би инсталирали pyodbc пакет:

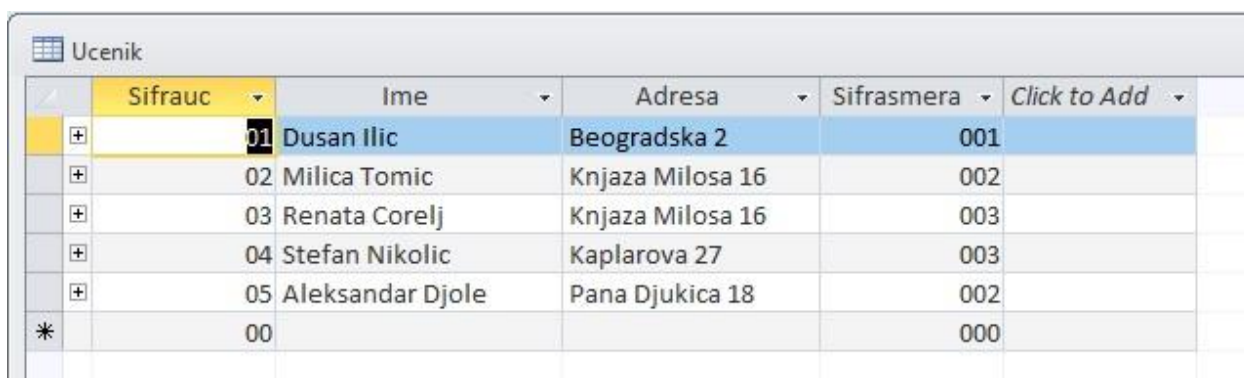


```
Microsoft Windows [Version 10.0.19041.746]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\PavleMila>cd C:\Users\PavleMila\AppData\Local\Programs\Python\Python38-32\Scripts
C:\Users\PavleMila\AppData\Local\Programs\Python\Python38-32\Scripts>pip install pyodbc
```

2: Креирање базе података и табела у MS Access

- Access база података се зове: **baza IV3.mdb**
- Табела се зове: **ucenik**
- Табела **ucenik** садржи следеће колоне и податке:



Sifrauc	Ime	Adresa	Sifrasmera	Click to Add
01	Dusan Ilic	Beogradska 2	001	
02	Milica Tomic	Knjaza Milosa 16	002	
03	Renata Corelj	Knjaza Milosa 16	003	
04	Stefan Nikolic	Kaplarova 27	003	
05	Aleksandar Djole	Pana Djukica 18	002	
00			000	

3: Повезивање Python-а са MS Access базом података

Повезивање Python-а са MS Access базом података:

(1) Покренути окружење Python IDLE

(2) Унесите следећи код у Python-у како бисте приказали садржај табеле **ucenik** у MS Access бази података. Зато је потребно да:

- Додате путању до базе података где је Access file (после **DBQ =**). Не заборавити да се дода екстензија базе података у MS Access-у на крају (**'accdB'**)

- Додате име табеле у **SELECT** наредби

```
conn = pyodbc.connect(r'Driver={Microsoft Access Driver (*.mdb, *.accdb)};DBQ=Baza
IV3.accdb;')
cursor.execute('SELECT * FROM ucenik')
```

Шта треба нагласити у овом примеру:

- Access база података се налази у фолдеру Documents:
C:\Users\PavleiMila\Documents\Baza IV3.accdb
- Табела коришћена у SELECT наредби је **ucenik**

Приказ целокупног кода у примеру:

```
import pyodbc

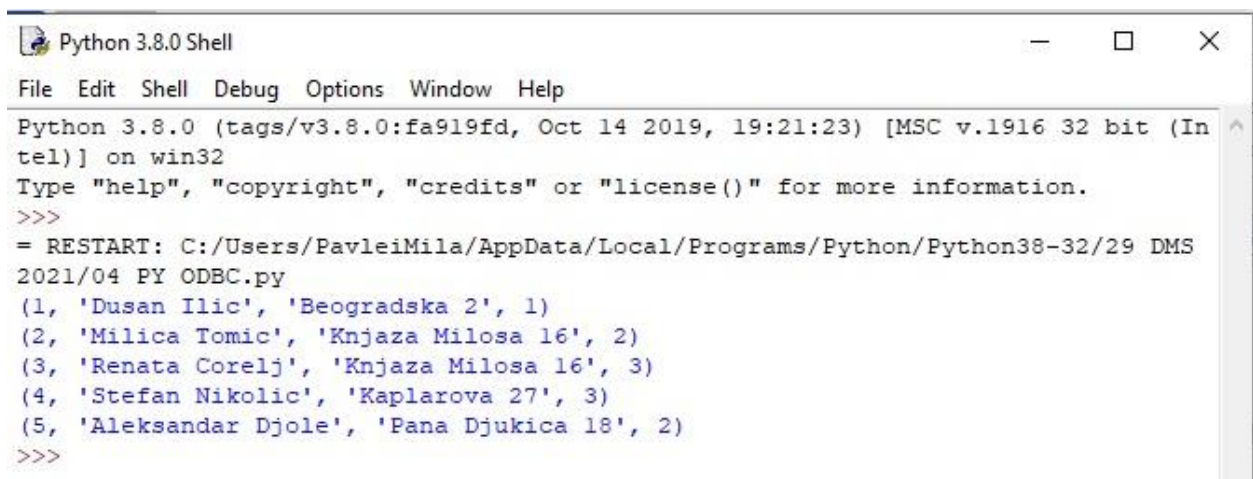
conn = pyodbc.connect(r'Driver={Microsoft Access Driver (*.mdb,
*.accdb)};DBQ=C:\Users\PavleiMila\Documents\Baza IV3.accdb;')

cursor = conn.cursor()
cursor.execute('select * from ucenik')

for row in cursor.fetchall():
    print (row)
```

4: Покренути код програма у Python-у

Покретањем програма добићете све слоге записане у табели **ucenik**.

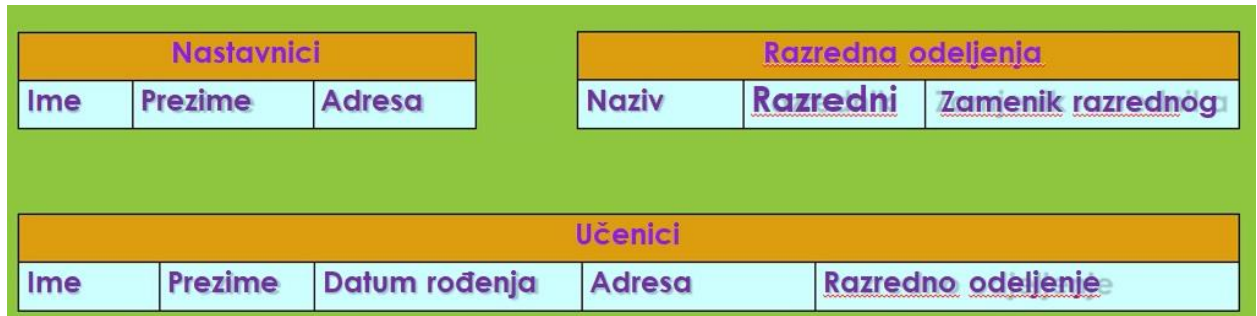


```
Python 3.8.0 Shell
File Edit Shell Debug Options Window Help
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:21:23) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/PavleiMila/AppData/Local/Programs/Python/Python38-32/29 DMS
2021/04 PY ODBC.py
(1, 'Dusan Ilic', 'Beogradska 2', 1)
(2, 'Milica Tomic', 'Knjaza Milosa 16', 2)
(3, 'Renata Corelj', 'Knjaza Milosa 16', 3)
(4, 'Stefan Nikolic', 'Kaplarova 27', 3)
(5, 'Aleksandar Djole', 'Pana Djukica 18', 2)
>>>
```

GUI апликација за рад са SQLite базом података

У овом делу представћемо кроз пример повезивање програмског језика Python са базом података у SQLite3 окружењу.

Структура табела у бази података изгледа као на следећој слици:



Записи података у табелама приказани су на следећој слици:

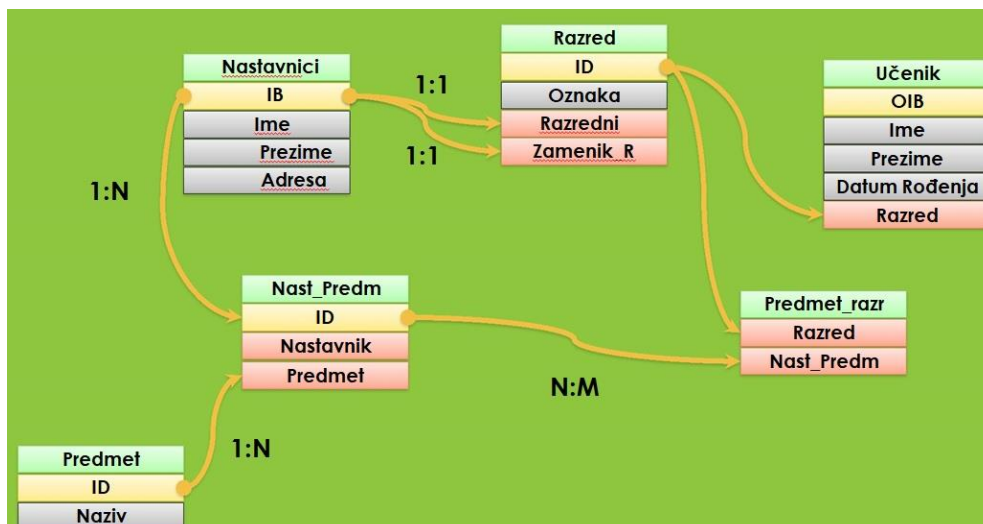
The screenshot displays the data records for the three tables:

Nastavnici		
Ime	Prezime	Adresa
Marko	Marković	Borska2
Stefan	Petrović	Niška 3
Marija	Ilić	Zetska 8
Mira	Minić	Petrina 6

Razredna odjeljenja		
Naziv	Razredni	Zamenik razrednog
2C	?	?
3F	?	?

Učenici				
Ime	Prezime	Datum rođenja	Adresa	Razredno odeljenje
Iva	Bukovac	13.4.1997	Borska 1	?
Maja	Marković	17.7.1997	Niška 13	?
Petar	Kodić	13.6.1996	Beogradska 2	?
Ivan	Petrić	13.6.1997	Terazije 4	?
Stefan	Tomić	14.2.1996	Tomina 17	?
Dragan	Ilić	4.7.1997	Požega 7	?

ER модел базе података приказан је на следећој слици:



Пример 1: Уз претпоставку да смо креирали табелу **Nastavnici** креирајмо табеле **Razredna odeljenja** и **Ucenici** са припадајућим примарним и страним кључевима.

Креирајмо табеле Nastavnici, Razredna_odeljenja и Ucenici, чија ће структура бити:

```
from sqlite3 import *

import sqlite3
conn = connect('d:\\skola.db')

c = conn.cursor()

s = 'CREATE TABLE Nastavnici ('
s += 'OIB TEXT PRIMARY KEY,'
s += 'Ime TEXT, '
s += 'Prezime TEXT, '
s += 'Adresa TEXT)'
c.execute(s)

s1 = 'CREATE TABLE Razredna_odeljenja ('
s1 += 'ID INTEGER PRIMARY KEY, '
s1 += 'Naziv TEXT, '
s1 += 'Razredni TEXT, '
s1 += 'Zamenik_razrednog TEXT, '
s1 += 'FOREIGN KEY(Razredni) REFERENCES Nastavnici (OIB), '
s1 += 'FOREIGN KEY(Zamenik_razrednog) REFERENCES Nastavnici (OIB))'
c.execute(s1)

s2 = 'CREATE TABLE Ucenici ('
s2 += 'JMBG TEXT PRIMARY KEY, '
s2 += 'Ime TEXT, '
s2 += 'Prezime TEXT, '
s2 += 'Datum_rodjenja TEXT, '
s2 += 'Adresa TEXT, '
s2 += 'Razredno_odeljenje INTEGER, '
s2 += 'FOREIGN KEY(Razredno_odeljenje) REFERENCES Razredna_odeljenja(ID))'
c.execute(s2)

s3 = 'INSERT INTO Nastavnici (OIB, '
s3 += 'Ime, Prezime, Adresa) '
s3 += 'VALUES ("2301978754128", '
s3 += '"Miljan", "Jeremić", "Kaplarova 37")'
c.execute(s3)

s4 = 'INSERT INTO Razredna_odeljenja (ID, '
s4 += 'Naziv, Razredni, Zamenik_razrednog) '
s4 += 'VALUES (1, "I/1", "Miljan G. Jeremic", '
s4 += '"Milan Lj. Gocic") '
c.execute(s4)
```

```
s5 = 'INSERT INTO Ucenici (JMBG, '
s5 += 'Ime, Prezime, Datum_rodjenja, Adresa, Razredno_odeljenje) '
s5 += 'VALUES ("2301984754121", '
s5 += '"Dusan", "Jeremić", "23.01.1984", "Kaplarova 37", 1)'
c.execute(s5)
```

```
s6 = 'INSERT INTO Ucenici (JMBG, '
s6 += 'Ime, Prezime, Datum_rodjenja, Adresa, Razredno_odeljenje) '
s6 += 'VALUES ("1502978750036", '
s6 += '"Jelena", "Jeremić", "15.02.1978", "Kaplarova 37", 1)'
c.execute(s6)
```

Пример 2: Напишимо програм који ће креирати апликацију са графичким корисничким интерфејсом која ће омогућавати уношење података о ученицима у базу података.



```
from tkinter import *
from sqlite3 import *
from tkinter.messagebox import *
```

```
from tkinter import font
import tkinter as tk
```

```
import sqlite3
```

```
import os
if os.path.exists("skola.db"):
    os.remove("skola.db")
else:
    print("The file does not exist")
```

```
conn = sqlite3.connect('skola.db')
c = conn.cursor()
```



```

class Program(Frame):
    def __init__(self, root):
        self.R = root
        self.R.title('Novi učenik')
        super().__init__(self.R)
        self.grid(rows = 5, columns = 4)
        self.kreirajGUI()
        self.ucitajRazrede()
        return

    def kreirajGUI(self):
        f = ('Calibri', 16, 'bold')
        Label(self, text = 'JMBG', font = f, fg = 'blue').grid(row = 0, column = 0, sticky =
W)#.pack()
        self.O = Entry(self, font = f, bg = 'yellow', fg = 'red')
        self.O.grid(row = 0, column = 1)
        Label(self, text='Datum rođenja', font = f,fg = 'blue').grid(row = 0,column = 2,
sticky=W)#.pack()
        self.D = Entry(self, font = f, bg = 'yellow', fg = 'red')
        self.D.grid(row = 0, column = 3)
        Label(self, text = 'Ime', font = f, fg = 'blue').grid(row = 1, column = 0, sticky =
W)#.pack()
        self.I = Entry(self, font = f, bg = 'yellow', fg = 'red')
        self.I.grid(row = 1, column = 1)
        Label(self, text = 'Prezime', font = f, fg = 'blue').grid(row = 1, column = 2, sticky =
W)#.pack()
        self.P = Entry(self, font = f, bg = 'yellow', fg = 'red')
        self.P.grid(row = 1, column = 3)
        Label(self, text = 'Adresa', font = f, fg = 'blue').grid(row = 2, column = 0, sticky =
W)#.pack()
        self.A = Text(self, width = 10, height = 3, font = f, bg = 'yellow', fg = 'red')
        self.A.grid(row = 3, column = 0, columnspan = 3, sticky = W+E+N+S)
        Label(self, text = 'Razred', font = f, fg = 'blue').grid(row = 2, column = 3, sticky =
W)#.pack()
        self.RA = self.ucitajRazrede()
        self.L = Listbox(self, font = f, bg = 'yellow', fg = 'red')
        for t in self.RA:
            self.L.insert(END, t[1])
        self.L.grid(row = 3, column = 3)
        Button(self, text = 'Spremi', command = self.spremi, font = f, bg = 'green', fg
='white').grid(row = 4, column = 0, columnspan = 4)#.pack()
        return

    def ucitajRazrede(self):
        conn = sqlite3.connect('d:\\skola.db')
        c = conn.cursor()
        upit = 'SELECT * FROM Razredna_odeljenja ORDER BY Naziv ASC'
        c.execute(upit)
        conn.commit()

```

```

print("Prikazani podaci o razredu!!!")
r = []

for t in [1]:
    r.append(("1M"))
    r.append(("2M"))
    r.append(("3M"))
    r.append(("4M"))

for t in c.execute(upit):
    r.append((t[0], t[1]))
conn.close()
return r

def curselection(self):
    return self.lists[0].curselection()

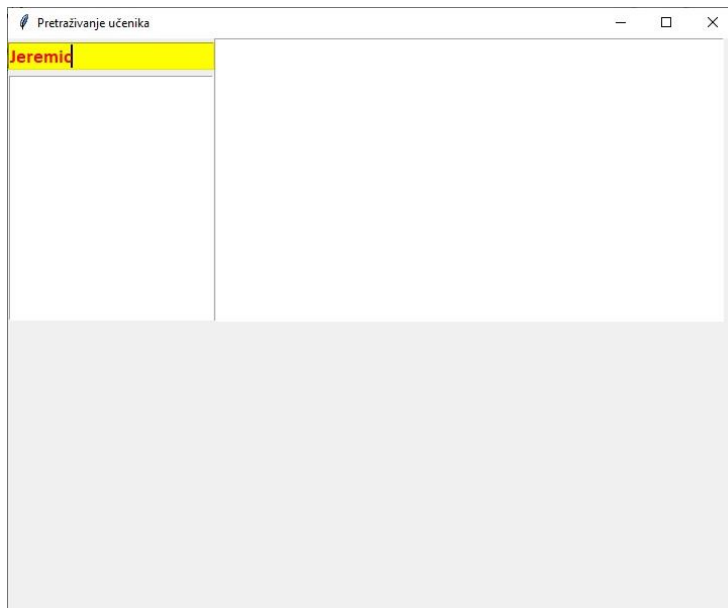
def spremi(self):
    conn = sqlite3.connect('d:\\skola.db')
    c = conn.cursor()
    t = self.L.curselection()
    print(t)
    if len(t) > 0:
        upit = 'INSERT INTO Ucenici (JMBG, Ime, Prezime, Datum_rodenja, Adresa, Razredno_odeljenje) '
        upit += 'VALUES ("{}", "{}", "{}", "{}", "{}", {})'
        .format(self.O.get(), self.I.get(), self.P.get(), self.D.get(), self.A.get(0.0, END), self.RA[t[0]][0])
        try:
            c.execute(upit)
            showinfo('Novi učenik', 'Učenik je dodat')
        except:
            showinfo('Novi učenik', 'Došlo je do greške')
    else:
        showinfo('Novi učenik', 'Niste odabrali razredno odeljenje')
    conn.commit()
    conn.close()
    return

def main():
    p = Program(Tk()) # p je objekt klase Program, a parametar mu je prozor glavnog programa (Tk)
    mainloop()

main() # poziv

```

Пример 3: Напишимо програм који ће креирати апликацију са графичким корисничким интерфејсом која ће омогућавати претраживање података о ученицима.



```
from tkinter import *
from sqlite3 import *
from tkinter.messagebox import *

from tkinter import font
import tkinter as tk

import sqlite3

con = sqlite3.connect('skola.db')

#from db import Database

# Instanciranje objekta baze podataka
#db = Database('skola.db')

class Nastavnik:
    def __init__(self, oib, ime, prezime, adresa):
        self.O = oib
        self.I = ime
        self.P = prezime
        self.A = adresa
        return

    def __repr__(self):
        return '{} {}'.format(self.I, self.P)

class Odeljenje:
    def __init__(self, ID, naziv, razredni, zamenik):
        self.ID = ID
        self.N = naziv
        self.R = razredni
```

```
self.Z = zamenik
return
```

```
class Ucenik:
```

```
def __init__(self, jmbg, ime, prezime, datR, adr, odel):
    self.O = jmbg
    self.I = ime
    self.P = prezime
    self.D = datR
    #self.M = mesT
    self.A = adr
    self.R = odel
    self.IP = self.I + ' ' + self.P
    return
```

```
def __repr__(self):
    s = 'Ime i prezime: { }({ })'.format(self.IP, self.O)
    s += '\nAdresa: ' + self.A
    s += '\nRazred: { }'.format(self.R.N)
    s += '\nRazredni: { }({ })'.format(self.R.R, self.R.Z)
    return s
```

```
class Trazi(Frame):
```

```
def __init__(self, root):
    self.R = root
    super().__init__(self.R)
    self.R.title('Pretraživanje učenika')
    self.grid(rows = 2, columns = 2)
    self.kreirajGUI()
    return
```

```
def kreirajGUI(self):
```

```
    f = ('Calibri', 14, 'bold')
    self.S = Entry(self, font = f, bg = 'yellow', fg = 'red')
    self.S.grid(row = 0, column = 0)
    self.S.bind('<KeyPress-Return>', self.trazi)
    self.L = Listbox(self, font = f)
    self.L.grid(row = 1, column = 0, rowspan = 3)
    self.L.bind('<<ListboxSelect>>', self.ucitajUcenika)
    self.T = Text(self, width=50, height=12, font=f, fg='blue')
    self.T.grid(row = 0, column = 1, rowspan = 2)
    return
```

```
def trazi(self, e):
```

```
    conn = sqlite3.connect('d:\\skola.db')
    c = conn.cursor()
    upit = 'SELECT u.*, n1.*, n2.*, r.* '
    upit += 'FROM Ucenici u, Nastavnici n1, Nastavnici n2, '
    upit += 'Razredna_odeljenja r '
    upit += 'WHERE u.razredno_odeljenje = r.ID AND '
```

```

upit += 'n1.OIB = r.Razredni AND n2.OIB = r.Zamenik_razrednog '
upit += 'AND (u.Ime LIKE "%{0}%" OR u.Prezime LIKE "%{0}%") '
upit += 'ORDER BY Prezime ASC'.format(self.S.get())
self.U = []
self.L.delete(0, END)
for t in c.execute(upit):
    r = Nastavnik(t[7], t[8], t[9], t[10])
    z = Nastavnik(t[11], t[12], t[13], t[14])
    o = Odeljenje(t[15], t[16], r, z)
    u = Ucenik(t[0], t[1], t[2], t[3], t[4], o)
    self.U.append(u)
    self.L.insert(END, u.IP)
#conn.close()
return

```

```

def ucitajUcenika(self, e):
    conn = sqlite3.connect('d:\\skola.db')
    c = conn.cursor()
    t = self.L.curselection()
    t = int(t[0])
    print(t)
    ""
    if len(t) > 0:
        upit = upit = 'SELECT * FROM Ucenik ORDER BY Prezime ASC'
        try:
            c.execute(upit)
            showinfo('Prikazani učenik', 'Učenik je u bazi')
        except:
            showinfo('Novi učenik', 'Došlo je do greške')
    else:
        showinfo('Novi učenik', 'Niste odabrali razredno odeljenje')
    ""
    u = self.U[t]
    self.T.delete(0.0, END)
    self.T.insert(END, u)

    conn.close()
    return t

```

```

if __name__ == "__main__":
    root = tk.Tk()
    root.title('DMS 2021 GUI 02')
    root.minsize(720, 600)
    my_gui = Trazi(root)
    root.mainloop()

```