

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
Diplomski studij informatike

Diplomski rad

**Sustav za analizu i vizualizaciju sirovih
podataka za praćenje aktivnosti D.
melanogaster**

Rijeka, lipanj 2019.

Milan Petrović
12345678901

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
Diplomski studij informatike

Diplomski rad

**Sustav za analizu i vizualizaciju sirovih
podataka za praćenje aktivnosti D.
melanogaster**

Mentor: izv.prof.dr.sc. Ana Meštrović

Rijeka, lipanj 2019.

Milan Petrović
12345678901

Umjesto ove stranice umetnuti zadatak
za završni ili diplomski rad

Sadržaj

Popis slika	vi
Popis tablica	vii
1 Python	1
1.1 Objektno orijentirano programiranje u Pythonu	2
1.2 Biblioteke i njihovo korištenje	3
1.2.1 Biblioteka TkInter	4
1.2.2 Biblioteka pandas	4
1.2.3 Biblioteka matplotlib	5
2 Podatci	6
2.1 Praćenje aktivnosti D. melanogaster	6
2.2 Sirovi podatci	7
2.3 Obrada podataka	7
3 Izgradnja sustava za obradu podataka	8
3.1 Arhitektura programa	8
3.2 Osnovni dio	8
3.2.1 Frameovi	9
3.2.2 Modul za obradu podataka	9

Sadržaj

4 Implementacija sustava	10
4.1 Obrada	10
4.2 Vizualizacija	11
4.3 Sučelje	11
Bibliografija	13
Sažetak	14

Popis slika

1.1	Vrste modula u programskom jeyiku Python.	4
2.1	D.melanogaster za vrijeme boravka u DAM1 sustavu	7
3.1	Skica rada programa	8

Popis tablica

Poglavlje 1

Python

Python je programski jezik visoke razine opće namjene. Autor programskog jezika Python je Guido van Rossum. Godine 1991 izlazi Python 0.9.0, prva javno dostupna verzija ovoga jezika (A Brief History of Python, 2018). Kod je strukturiran tako da se koristi metoda uvlačenja(tabova) za razlikovanje programskih blokova. Programiranje u python void se jednostavnošću i čitljivosti koda. Ovim stilom pisanja cilj je omogućiti programerima pisanje čistog i jednostavnog koda neovisno o veličini projekta. Stilovi pisanja programskog jezika Python su objektno orijentirano, strukturno i aspektno orijentirano. Zbog ove fleksibilnosti jedan je od najzastupljenijih programskih jezika današnjice. Cijela filozofija pisanja programskog koda u Pythonu najbolje se može vidjeti iz 19 smjernica koje se mogu vidjeti ako u program unesemo naredbu “import this” (The Zen of Python, 2019). Nakon pokretanja programa na ekran se ispisuju smjernice. Neke koje najbolje opisuju stil pisanja programskog koda su:

- Beautiful is better than ugly.
- Explicit is better than implicit.
- Simple is better than complex.
- Flat is better than nested.
- Readability counts.
- If the implementation is hard to explain, it’s a bad idea.

- If the implementation is easy to explain, it may be a good idea.

Inspiraciju za naziv Guido van Rossum dobio je čitajući skripte britanskoga serijala komedije “Monty Python Flying Circus”. Van Rossum je smatrao da sam jezik mora imati naziv koji je kratak, jednostavan I pomalo tajanstven, iz toga razloga nazvao ga je Python. (General Python FAQ, 2019)

1.1 Objektno orijentirano programiranje u Pythonu

Prilikom pisanja programa koji su u svojoj strukturi složeni, sadrže više cjelina koje se pokreću u program, javlja se problem organizacije koda. Ovaj problem može se pojednostaviti da se kod strukturira u objekte koji međusobno komuniciraju i izvršavaju naredbe. Ovakav stil strukturiranja koda naziva se paradigma objektno orijentiranog programiranja. Kase i objekti su dva glavna elementa objektno orijentiranog programiranja. Klasa stvara novi tip u kojem su objekti instance te klase. (Object Oriented Programming , 2019) Pohrana podataka u objektu vrši se pomoću običnih varijabli koje pripadaju tom objektu. Objekti također mogu koristiti funkcije koje pripadaju klasi. Takve funkcije nazivamo metodama klase. U klasama razlikujemo dva tipa varijabli, varijable instance i varijable klase. Klasu stvaramo pomoću ključne riječi `class`, u nastavku je prikazan jednostavan primjer stvaranja klase u Pythonu.

```
1 class Student:
2     def __init__(self, name, age):
3         self.name = name
4         self.age = age
5 p1 = Student("Mark", 20)
```

Listing 1.1 yolo

U izrazu `p1 = Student("Mark", 20)` vršimo kreiranje instance klase tako da se prvo upisuje naziv klase, u ovom slučaju riječju `Student`, te zatim klasi prosljeđujemo vrijednosti imena i godine. Unutar same klase vrijednosti se pohranjuju u tu instance

Poglavlje 1. Python

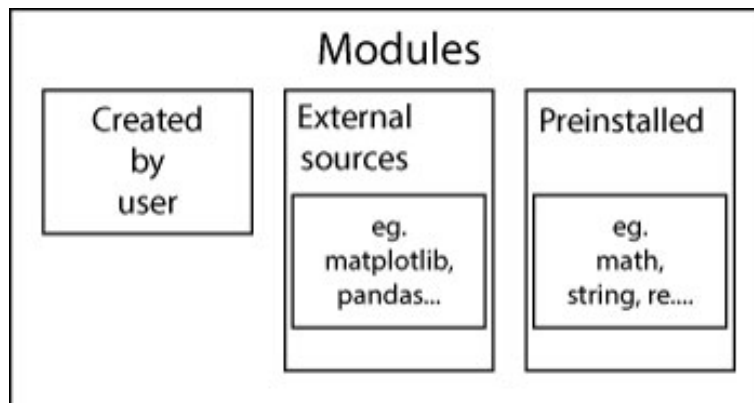
pomoću ključne riječi `self`. Primijetimo u kodu unutar klase nalazi se metoda `__init__`, ova metoda pokreće se čim se object klase instancira(stvori). Ova metoda korisna je za bilo koju inicijalizaciju, tj. Prosljeđivanje početnih vrijednosti klase objektu. Kao što je navedeno klase mogu sadržavati i metode, metode su zapravo funkcije koje sadržavaju ključnu riječ `self`. U nastavku je prikazan primjer koda metode unutar klase.

```
1 class SomeClass:
2     def greetings(self):
3         print('Hello stranger, Good to see you!')
4 p = SomeClass().greetings()
5
6 Output:
7 'Hello stranger, Good to see you!'
```

Listing 1.2 test

1.2 Biblioteke i njihovo korištenje

Biblioteke u Pythonu su skup funkcija i metoda koje omogućuju obavljanje složenih radnji bez pisanja koda. Radnje se vrše pomoću poziva funkcija. Biblioteke sadrže velik broj korisnih modula koji se mogu pozivati u samom programu. Python u svom osnovnom obliku sadržava osnovnu biblioteku koja je sama po sebi opsežna zbirka dobro dokumentiranih modula za svakodnevno programiranje. Osim osnovne biblioteke korisnici su razvili i brojne druge koje se odnose na izvršavanje posebnih zahtjeva u programu. Moduli u Pythonu mogu se klasificirati u tri vrste: stvoreni od strane korisnika, stvoreni od vanjskih izvora I prethodno instalirani moduli koji dolaze s osnovnom verzijom Pythona. Python Package Index ili skraćeno PyPI također poznat i po nadimku Cheese Shop(iz Monty Pythonovg skeča Cheese Shop). Službeni je repozitorij za pakete stvorene od treće strane za Python. Trenutno postoji preko 180 000 Python biblioteka kojima se može pristupiti putem PyPI-a (pypi.org, 2019).



Slika 1.1 Vrste modula u programskom jeziku Python.

1.2.1 Biblioteka TkInter

TkInter je Pythonog standardni GUI(Graphical User Interface) paket(TkInter, 2019). TkInter je uključen u standardni instalacijski paket Pythona. Ime TkInter dolazi od izvedenice iz riječi “Tk” I “Interface”. Autori TkInter biblioteke I osobe zaslužne za razvoj su Fredrik Lundh, Steen Lumholt i Guido van Rossum (Shipman, 2010). Terminologija koja se najčešće spominju prilikom rada s TkInterom su:

- Window(prozor) – Iako sam pojam ima više značenja ovisno o kontekstu, (pypi.org, 2019) najčešće se odnosi na kvadratni prikaz na korisnikovom zaslonu
- Widget – pojam za bilo koji od blokova koji čine aplikaciju u korisničkom sučelju. Widgete možemo podijeliti ovisno o njihovoj ulozi u programu(tekstualni, gumb, oznaka, poruka...).
- Frame(okvir) – Osnovna jedinica prilikom organizacije složenijih sučelja. Okvir je pravokutno područje koje sadržava ostale widgete.

1.2.2 Biblioteka pandas

Pandas je biblioteka za manipulaciju i analizu podataka. Pruža korisniku mogućnosti strukturiranja podataka u vremenskim serijama ukoliko vremenska varijabala postoji u podacima. Naziv ove biblioteke izveden je iz riječi “panel” i “data”. Autor

Poglavlje 1. Python

ove biblioteke je Wes McKinney. Za vrijeme rada u AQR Capital Managamentu javila se potreba za fleksibilnim alatom visokih performansi za kvantitativnu alnalizu finansijskih podataka. Iz toga razloga Wes McKinney 2008. počinje raditi na razvoju biblioteke pandas koja bi to omogućila. Prije samog napuštanja AQR-a McKinney uvjerava upravu da se ova pandas objavi kao biblioteka otvorenog koda. Pandas nudi razne funkcije koje omogućavaju jednostavan rad sa velikim skupoveima podataka. Neke od tih funkcija su:

- Pohrana podataka u DataFrame objekte koji omogućavaju daljnju manipulaciju nad istima.
- Rukovanje sa zapisima koji nedostaju u podatcima.
- Promjena indeksa u podatcima, dodjeljivanje indeksa nad određenim stupcem jedinstvenih podataka.
- Umetanje I brisanje stupaca I redaka.
- Vizualizacija podataka pomoću vremena.
- Filtriranje podataka
- Pozivanje funkcija za izračun nad podatcima te pohrana vrijednosti u nove stupce.

1.2.3 Biblioteka matplotlib

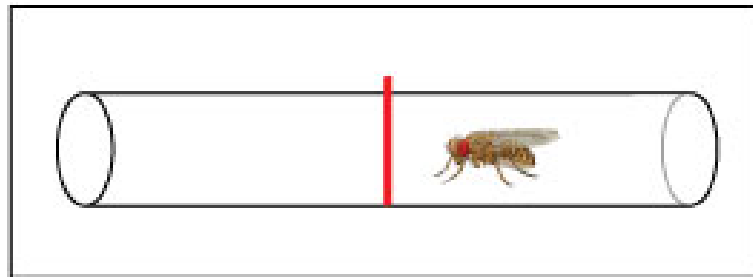
Matplotlib je biblioteka za vizualizaciju podataka. Pomoću API-a omogućava ugrađivanje vizualizacija u aplikacije koje koriste korisničko sučelje(GUI) kao što su TkInter, Qt ili GTK+. Matplotlib izvorno je napisao John D. Hunter, matplotlib ima aktivnu zajednicu koja održava I unapriješuje ovu biblioteku. Prije smrti Johna D. Huntera Michael Droettboom nominiran je za vodećeg developera matplotliba. Matplotlib 1.2 je prva verzija koja podržava Python 3.x te će u skorije vrijeme prebaciti se potpuno na Python 3.

Poglavlje 2

Podatci

2.1 Praćenje aktivnosti *D. melanogaster*

Vinska mušica (lat. *Drosophila melanogaster*) je kukac dvokrilac, postoji puno različitih vrsta mušica a među njima vinska je najpoznatija (Vinegar Flies , 2019). Glavni razlog popularnosti vinske mušice je u tome da je ona dobar laboratorijski model za istraživanje. Ovaj organizam naširoko se koristi za biološka istraživanja u genetici, fiziologiji, evoluciji i sl. *Drosophila* se koristi u istraživanja zbog vrlo lakog uzgoja, brzog razmnožavanja i stvaranja nekoliko stotina potomaka. Vrijeme stvaranja jedne generacija je 14 dana što je čini prednost prilikom istraživanja gena i nasljednih svojstava. Praćenje aktivnosti *D. melanogaster* vrši se pomoću *Drosophila Activity Monitor* (DAM) sustava. DAM sustav bilježi prekide infracrvenog snopa u staklenim cjevčicama gdje se mušica nalazi. DAM sustav ima 32 kanala za praćenje lokomotornih aktivnosti 32 jedinki mušica. Na Slici 2.1



Slika 2.1 D.melanogaster za vrijeme boravka u DAM1 sustavu

2.2 Sirovi podatci

2.3 Obrada podataka

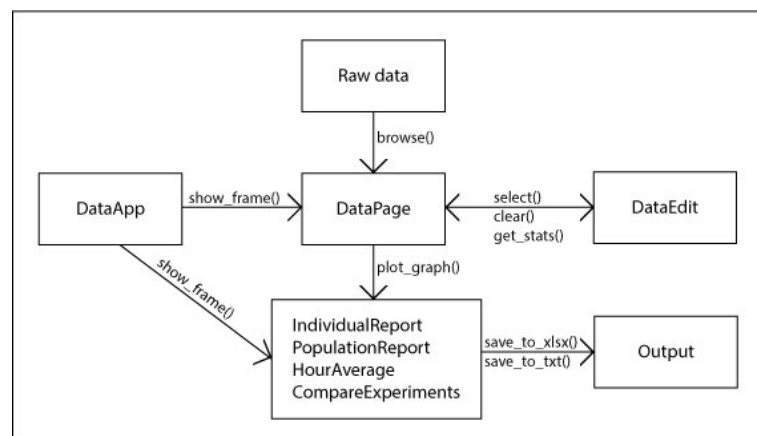
```
1 def browse(self):  
2     filepath = filedialog.askopenfilename(  
3         initialdir = r"C:\\Users\\Desktop\\gui-master",  
4         title = "Select file",  
5         filetypes = (('txt', '*.txt'), ("all files", "*.*"))  
6  
7     self.controller.df = pd.read_csv(filepath, sep='\\t', header=  
        None)
```

Listing 2.1 test

Poglavlje 3

Izgradnja sustava za obradu podataka

3.1 Arhitektura programa



Slika 3.1 Skica rada programa

3.2 Osnovni dio

Poglavlje 3. Izgradnja sustava za obradu podataka

```
1 from DataApp import DataApp_  
2  
3 def main():  
4     app = DataApp_  
5     app.mainloop()  
6  
7 if __name__ == '__main__':  
8     main()
```

Listing 3.1 Funkcija main

```
1 self.frames = {}  
2 for F in (DataPage_, GraphPage_, IndividualReport_,  
3           CompareExperiments_, HourAve_):  
4     frame = F(container, self)  
5     self.frames[F] = frame  
6     frame.grid(row=0, column=0, sticky='nsew')  
7 self.show_frame(DataPage_)
```

Listing 3.2 Funkcija main

```
1 def show_frame(self, cont):  
2     frame = self.frames[cont]  
3     frame.tkraise()
```

Listing 3.3 Funkcija main

3.2.1 Frameovi

```
1 class StartPage_(tk.Frame):  
2     def __init__(self, parent, controller):  
3         tk.Frame.__init__(self, parent)
```

Listing 3.4 Funkcija main

3.2.2 Modul za obradu podataka

Poglavlje 4

Implementacija sustava

4.1 Obrada

```
1 def select_cols_rows(self, start_col, end_col, start_row, end_row
   ):
2
3 def get_stats(self):
4     df['mean'] = df.iloc[:, 7:].mean(axis=1)
5     df['std'] = df.iloc[:, 7:].std(axis=1)
6     df['sum'] = df.iloc[:, 7:].sum(axis=1)
7
8 def split_for_graph(self, start_bsl_morning, start_1st_expo,
   start_bsl_noon, start_2nd_expo):
9     df_ave = self.controller.df.copy()
10
11     df_ave['datetime'] = pd.to_datetime(df_ave['datetime'])
12     df_ave.index = df_ave['datetime']
13     df_p = df_ave.resample('H').mean()
14
15     df_p['Hour'] = df_ave['datetime'].dt.hour
16
17     df_hour_ave = pd.DataFrame(df_p['mean'])
18
```

```
19 self.controller.df_hour_ave = df_hour_ave
```

Listing 4.1 Funkcija main

4.2 Vizualizacija

```
1 def plot_ave_hour_graph(self, event):
2     try:
3         self.canvas.get_tk_widget().pack_forget()
4     except AttributeError:
5         pass
6
7     df = self.controller.df_hour_ave
8
9
10    f = Figure(figsize=(5, 5), dpi=100)
11    a = f.add_subplot(111)
12
13    a.plot(df['mean'])
14
15    a.legend(['hour average'])
16    self.canvas = FigureCanvasTkAgg(f, self)
17    self.canvas.get_tk_widget().pack(side=tk.TOP, fill=tk.BOTH,
18                                     expand=True)
19    self.canvas.draw()
```

Listing 4.2 Funkcija main

4.3 Sučelje

```
1 tk.Label(self, text='Data edit').grid(pady=10, padx=10)
2
3 ttk.Button(self, text='Population',
4            command=lambda: controller.show_frame(GraphPage_)).grid(row=1,
5                               column=4)
```

Poglavlje 4. Implementacija sustava

```
6 start_col = tk.StringVar()
7 self.start_col = ttk.Entry(self, textvariable=start_col).grid(row
    =3, column=1)
8
9 T = tk.Text(self, height=30, width=200, wrap=None)
10 T.grid(row=2, column=4, columnspan=14, rowspan=14, padx=10, pady
    =10)
11 T.insert(tk.END, self.controller.df.to_string())
12
13 self.canvas = FigureCanvasTkAgg(f, self)
14 self.canvas.get_tk_widget().pack(side=tk.TOP, fill=tk.BOTH,
    expand=True)
15 self.canvas.draw()
```

Listing 4.3 Funkcija main

Bibliografija

- [1] LaTeX - a document preparation system. , s Interneta, <http://www.latex-project.org/> , srpanj 2016.
- [2] TeXstudio. , s Interneta, <http://www.texstudio.org/> , srpanj 2016.
- [3] JabRef. , s Interneta, <http://www.jabref.org/> , srpanj 2016.
- [4] Tehnički fakultet u Rijeci, *Upute za pisanje završnog/diplomskog rada*, srpanj 2016.
- [5] WikiBooks: LaTeX. , s Interneta, <http://en.wikibooks.org/wiki/LaTeX/> , srpanj 2016.
- [6] Tex—LaTeX Stack Exchange. , s Interneta, <http://tex.stackexchange.com/> , srpanj 2016.

Sažetak

Ovo je tekst u kojem se opiše sažetak vašega rada. Tekst treba imati duh rekapitulacije što je prikazano u radu, nakon čega slijedi 3-5 ključnih riječi (zamijenite dolje postavljene općenite predloške riječi nekim suvislim vlastitim ključnim riječima).

Ključne riječi — ključna riječ 1, ključna riječ 2, ključna riječ 3

Abstract

This is a text where a brief summary of your work is outlined. The text should have a sense of recap of what was presented in the thesis, followed by 3-5 keywords (replace the general keyword templates below with some meaningful keywords of your own) .

Keywords — keyword 1, keyword 2, keyword 3