



IPAQ PETA

V. GIMNAZIJA ZAGREB

GIMNAZIJA VUKOVAR

SREDNJA ŠKOLA LOVRE MONTIJA KNIN

SREDNJA ŠKOLA PAKRAC

GIMNAZIJA METKOVIĆ

PMF ZAGREB

Višedretveno programiranje

ULAGANJE U BUDUĆNOST



PROJEKT
SUFINANCIRA
EUROPSKA UNIJA

Ova publikacija je izrađena uz pomoć Europske unije. Sadržaj publikacije je u isključivoj nadležnosti V. gimnazije te se ni na koji način ne može smatrati da odražava stajališta Europske unije.

Europsku uniju čini 28 država članica koje su odlučile postupno povezivati svoja znanja, resurse i sudbine. Tijekom 50-godišnjeg razdoblja proširivanja, zajedno su izgradile područje stabilnosti, demokracije i održivog razvoja, zadržavajući pritom kulturnu raznolikost, toleranciju i slobode pojedinaca. Europska unija je posvećena dijeljenju svojih postignuća i vrijednosti s državama i narodima izvan svojih granica.

Višedretvene aplikacije

- kako omogućiti da istovremeno više korisnika može zahtijevati i primati datoteke?
- kako paralelno s primanjem i slanjem zahtijeva primati nove klijente?
- Rješenje:
 - dretve (thread) – omogućavaju istovremeno izvodenje dvaju ili više dijelova programa

ULAGANJE U BUDUĆNOST



PROJEKT
SUFINANCIRA
EUROPSKA UNIJA



Dretve

- modul **threading**
- klasa **Thread**
- parametri konstruktora:
 - **target** – metoda koja će se izvoditi unutar dretve
 - **args** – parametri metode (navode se unutar okruglih zagrada)
- metode:
 - **start ()** – pokreće dretvu
 - **join ()** – osigurava da će glavni program čekati dok se odgovarajuća dretva ne izvede do kraja

ULAGANJE U BUDUĆNOST



PROJEKT
SUFINANCIRA
EUROPSKA UNIJA



Primjer 3:

Kreirajmo program koji će imati dvije dretve:

- prva dretva ispisuje redom parne prirodne brojeve do 20 i nakon svakog ispisanog parnog broja čeka jednu sekundu
- druga petlja ispisuje redom neparne prirodne brojeve do 20 i nakon ispisa čeka dvije sekunde

ULAGANJE U BUDUĆNOST



PROJEKT
SUFINANCIRA
EUROPSKA UNIJA



Primjer 3 - rješenje:

```
from threading import *
from time import *

def parni():
    for i in range(2, 21, 2):
        print(i)
        sleep(1)
    return

def neparni():
    for i in range(1, 20, 2):
        print(i)
        sleep(2)
    return

if __name__ == '__main__':
    t1 = Thread(target = parni)
    t2 = Thread(target = neparni)
    t1.start()
    t2.start()
    t1.join()
    t2.join()
```

```
Command Prompt
D:\mreze>c:\python34\Python primjer3.py
2
1
4
3
6
5
8
7
10
12
7
14
16
9
18
20
11
13
15
17
19
D:\mreze>_
```

ULAGANJE U BUDUĆNOST



PROJEKT
SUFINANCIRA
EUROPSKA UNIJA



Primjer 3 – rješenje 2:

```
from threading import *
from time import *

def f(a):
    for i in range(a, 21, 2):
        print(i)
        sleep(a % 2 + 1)
    return

if __name__ == '__main__':
    t1 = Thread(target = f, args = (1,))
    t2 = Thread(target = f, args = (2,))
    t1.start()
    t2.start()
    t1.join()
    t2.join()
```

ULAGANJE U BUDUĆNOST



PROJEKT
SUFINANCIRA
EUROPSKA UNIJA



Primjer 3 – rješenje 3:

```
from threading import *
from time import *

def f(a):
    for i in range(a, 21, 2):
        print(i)
        sleep(a % 2 + 1)
    return

if __name__ == '__main__':
    t = [Thread (target = f, args = (i, )) for i in range(1, 3)]
    for k in t:
        k.start()
    for k in t:
        k.join()
```

ULAGANJE U BUDUĆNOST



PROJEKT
SUFINANCIRA
EUROPSKA UNIJA



Primjer 4:

Kreirajmo chat aplikaciju koja će omogućiti razgovor više korisnika istovremeno u istoj sobi te prihvaćanje novih korisnika.

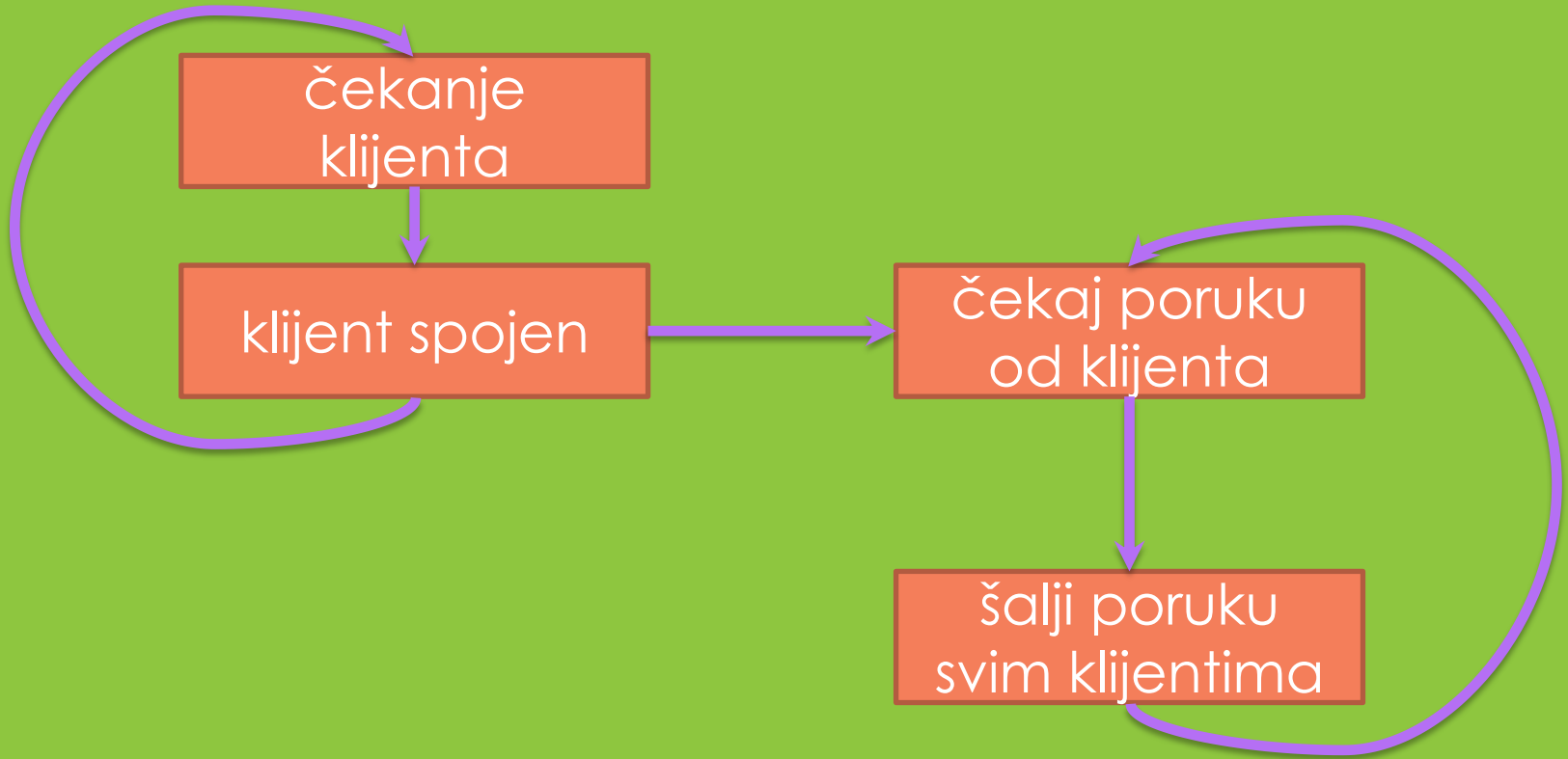
ULAGANJE U BUDUĆNOST



PROJEKT
SUFINANCIRA
EUROPSKA UNIJA



Primjer 4 – skica rješenja - poslužitelj:



ULAGANJE U BUDUĆNOST



Primjer 4 - poslužitelj:

```
if __name__ == '__main__':
    klijenti = dict()
    s = socket()
    s.bind(('localhost', 10000))
    s.listen(5)
    def citaj(c, ime):
        while True:
            poruka = c.recv(1024).decode('utf-8')
            if poruka == '@kraj@':
                del(klijenti[ime])
                saljiSvima('{0} je napustio chat'.format(ime))
                return
            else:
                poruka = '{}:: {}'.format(ime, poruka)
                saljiSvima(poruka)
    while True:
        c, t = s.accept()
        ime = c.recv(1024).decode('utf-8')
        citaj(c, ime)
    return
```

```
def saljiSvima(poruka):
    print(poruka)
    for t in klijenti:
        klijenti[t].send(poruka.encode('utf-8'))
    return
```

ULAGANJE U BUDUĆNOST



PROJEKT
SUFINANCIRA
EUROPSKA UNIJA



Primjer 4 - klijent:

```
class Klijent(Frame):
    def kreirajSucelje(self):
        self.grid(rows = 2, column
        self.T = Text(self)
        self.T.grid(row = 1, colu
        self.UV = StringVar()
        self.V = Entry(self)
        self.V.grid(row =
        self.V.bind('<Key
        return

    def salji(self, e):
        t = self.UV.get()
        self.UV.set('')

    def close(self):
        self.S.send('@kraj@'.encode('UTF-8'))
        print('Kraj')
        self.R.destroy()
        return

    def citaj(self):
        while True:
            try:
                t = self.S.recv(1024).decode('U
                self.T.insert(END, t+ '\n')
            except:
                return
        return

if __name__ == '__main__':
    k = Klijent(Tk())
    t = Thread(target = k.citaj)
    t.start()
    mainloop()
```

ULAGANJE U BUDUĆNOST



PROJEKT
SUFINANCIRA
EUROPSKA UNIJA

