

Ponavljanje i sve je objekat

Šta pravimo ovog semestra i šta je ostalo iz PZU1, sada viđeno kroz objekte

Danas pravimo: paletu boja i listu pahuljica, pa ih crtamo u prozoru



Rečnik sa tri boje i lista od 60 torki: ništa se ne pomera, ali sve je već objekat

Ovog semestra pravimo igru „Zimska šetnja“



Čikica hvata pahuljice i skuplja poene. Ovako otprilike izgleda igra u temi 10.

Jedan korak igre u svakoj temi:

- 06 zimska scena
- 07 pada sneg
- 08 Čikica hoda strelicama
- 09 sprajtovi i animacija hoda
- 10 poeni, životi, sudari
- 11 zvuk i čestice
- 12 meni, pauza i kraj igre

Teme 01–05: klase i objekti, u konzoli.

Semestar: 13 tema u 7 blokova

Blok 1

01

Ponavljanje i
sve je objekat

02

Klase i objekti

Blok 2

03

Atribut klase,
klasna i statička
metoda

04

Nasleđivanje

Blok 3

05

Enkapsulacija,
polimorfizam i
specijalne
metode

06

Prozor, crtanje,
slike i tekst

Blok 4

07

Petlja igre,
vreme i kretanje

08

Događaji i unos

Blok 5

09

Sprajtovi, grupe
i animacija lika

10

Sudari i pravila
igre

Blok 6

11

Zvuk,
transformacije
slike i čestice

12

Stanja igre i
završni projekat

Blok 7

13

Vežbanje i
priprema za
ispit

**Odbrana
seminarskih
radova**

 klase i objekti (konzola)

 PyGame i igra

Danas: blok 1, tema 01

Ocena ima tri dela, ukupno 100 poena

30

poena

40

poena

30

poena

Seminarski rad

samostalan rad

odbrana u bloku 7

Praktični deo ispita

programi na računaru

klase i mala igra

Teorijski deo ispita

kratka pitanja

pojmovi i „šta ispisuje“

Ispit traje 3 sata: 2 sata praktični deo, 1 sat teorijski deo. Prag je 40% na praktičnom i 40% na teorijskom delu. Ko ne položi jedan deo, u sledećem roku polaže oba dela ponovo.

Ko drži predavanja, a ko vežbe

PREDAVANJA

dr Staša Vujičić Stanković

`stasa.vujicic.stankovic@math.rs`

`www.matf.bg.ac.rs/~stasa.vujicic.stankovic`

pojmovi, primeri i igra korak po korak

VEŽBE

Milan Kocić

`milan.kocic@matf.bg.ac.rs`

`poincare.matf.bg.ac.rs/~milan.kocic`

zadaci za vežbu i seminarski rad

Časovi su najčešće onlajn: delimo ekran, a pitanja i odgovore na glasanjima pišete u čet. Pitanje može i mejlom, između časova.

pygame-ce: dve komande u terminalu

TERMINAL

```
pip uninstall pygame  
pip install pygame-ce
```

- Prva komanda briše stari pygame, ako postoji; dva paketa ne mogu zajedno.
- Uvoz ostaje isti: `import pygame`.
- Na Mac-u i Linux-u komanda je često `pip3`.

```
import pygame as pg  
  
print(pg.version.ver)  
print(bool(getattr(pg, "IS_CE", False)))
```

ISPIS

```
2.5.8  
True
```

Šta pamтите iz PZU1? Tri pitanja bez ocene

1. liste

```
boje = ["beļa", "plava"]  
boje.append("zlatna")  
print(boje[-1], len(boje))
```

zlatna 3

2. rečnici

```
z = {"Ana": 3}  
z["Ana"] += 2  
z["Mia"] = 1  
print(z["Ana"], len(z))
```

5 2

3. funkcije

```
def dupla(x):  
    return 2 * x  
  
print(dupla("ha") + "!")
```

haha!

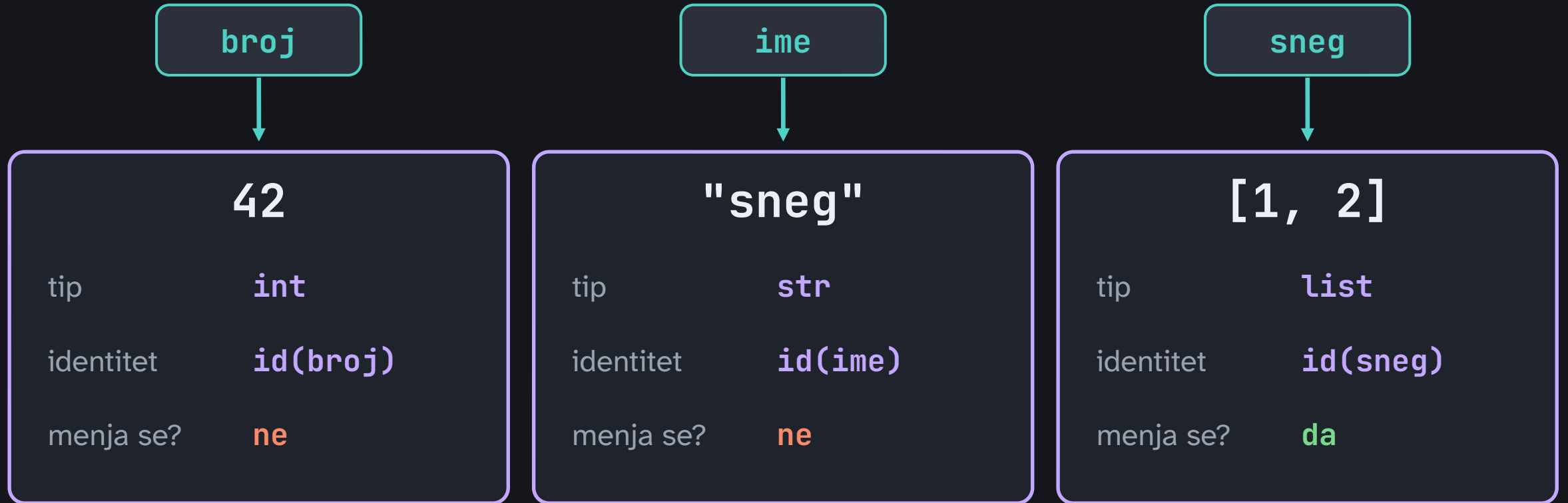
Šest novih reči za danas

objekat	svaka vrednost u Pythonu: broj, string, lista, pa i funkcija
tip (klasa)	vrsta objekta; kaže šta objekat ume: <code>type(x)</code>
identitet	broj po kome Python razlikuje objekte: <code>id(x)</code>
referenca	ime koje pokazuje na objekat; dva imena, isti objekat
metoda	funkcija koja pripada objektu: <code>tekst.upper()</code>
nepromenljiv	objekat koji se ne može izmeniti: string, torka

Sve je objekat

Broj, string, lista, funkcija: sve ima tip, identitet i vrednost

Svaki objekat ima tip, identitet i vrednost



Ime nije objekat: ime samo pokazuje na objekat, kao etiketa na kutiji.

type kaže kojoj klasi pripada vrednost

```
print(type(42))
print(type(3.14))
print(type(1 + 2j))
print(type(True))
print(type("sneg"))
print(type([1, 2]))
print(type((1, 2)))
print(type({1, 2}))
print(type({"x": 1}))
```

ISPIS

```
<class 'int'>
<class 'float'>
<class 'complex'>
<class 'bool'>
<class 'str'>
<class 'list'>
<class 'tuple'>
<class 'set'>
<class 'dict'>
```

„Tip“ i „klasa“ znače isto: 42 je objekat klase int.

I funkcije i moduli su objekti

```
import math

print(type(print))
print(type(math))
ispisi = print # novo ime, ista funkcija
ispisi("Zdravo!")
print(ispisi is print)
```

ISPIS

```
<class 'builtin_function_or_method'>
<class 'module'>
Zdravo!
True
```

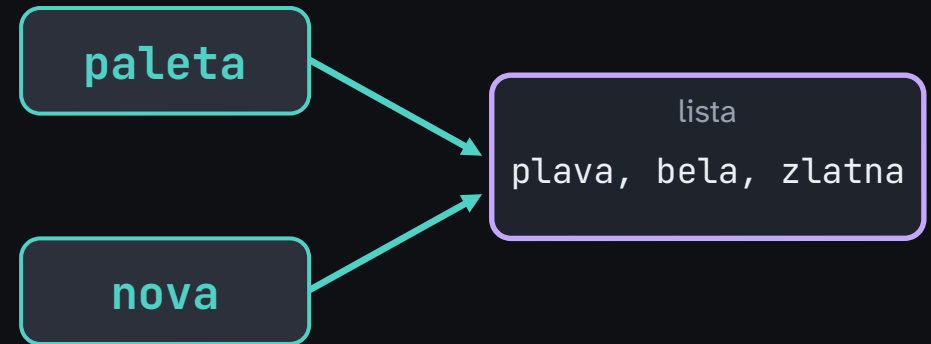
- print je objekat, kao i broj 42.
- Ime ispisi pokazuje na istu funkciju kao print.
- U temi 02 pravimo svoju klasu, pa i svoje objekte.

Dodela ne pravi kopiju: dve etikete, jedan objekat

```
paleta = ["plava", "bela"]  
nova = paleta  
print(id(paleta) == id(nova))  
nova.append("zlatna")  
print(paleta)
```

ISPIS

```
True  
['plava', 'bela', 'zlatna']
```



nova.append(...) menja objekat, pa se promena vidi i preko imena paleta.

Šta ispisuje ovaj program?

```
sneg = [1, 2, 3]
kopija = sneg
kopija = kopija + [4]
print(len(sneg))
```

Pažljivo: u trećem redu nema append.

A

3

tačno

B

4

C

TypeError

D

1

== pita „jednaki?“, a is pita „isti objekat?“

```
a = [10, 20]
b = [10, 20]
c = a
print(a == b, a is b)
print(a is c)
print(a is not b)
```

ISPIS

```
True False
True
True
```

Primer iz života: dve iste šolje su jednake (==), ali nisu ista šolja (is). Dva imena za jednu šolju su ista šolja.

Šta sme da se menja

Stringovi i torke ostaju isti; menja se samo ime koje na njih pokazuje

Metode stringa vraćaju nov string

```
tekst = "zimska šetnja"  
print(tekst.upper())  
print(tekst.title())  
print(tekst.replace("zimska", "letnja"))  
print(tekst.find("š"))  
print(tekst.split())  
print(tekst.center(21, "*"))  
print(tekst)
```

ISPIS

```
ZIMSKA ŠETNJA  
Zimska Šetnja  
letnja šetnja  
7  
['zimska', 'šetnja']  
****zimska šetnja****  
zimska šetnja
```

Poslednji red: tekst je ostao isti.

Znak u stringu ne možemo da zamenimo

```
rec = "mačka"  
rec[0] = "t"
```

PORUKA

```
TypeError:  
'str' object does not support  
item assignment
```

Rešenje: napravimo nov string

```
rec = "mačka"  
nova = "t" + rec[1:]  
print(nova, rec)
```

ISPIS

```
tačka mačka
```

Čudovište: torka koja se ne da menjati

```
polozaj = (100, 50)
polozaj[0] = 120
```

TypeError:

'tuple' object does not support item assignment

Ispravka:

```
polozaj = (100, 50)
polozaj = (120, polozaj[1])
print(polozaj)
```

Kako izgleda

TypeError: 'tuple' object does not support item assignment, u redu gde dodeljujemo vrednost elementu torke.

Zašto se pojavljuje

Torka je nepromenljiva: kad je napravljena, njeni elementi se više ne menjaju.

Kako se pobeđuje

Napravimo novu torku i damo je istom imenu, ili od početka koristimo listu.

Šta ispisuje ovaj program?

```
boja = (30, 40, 90)
boja = boja + (255,)
print(len(boja))
```

Torka se ne menja. Da li to znači grešku?

A
3

B tačno
4

C
TypeError

D
255

Kolekcije i ispis

Lista, torka, skup i rečnik, pa print sa zvezdicom, join i map

Četiri kolekcije: šta se menja i kako se pristupa

Kolekcija	Zapis	Menja se?	Redosled i pristup	Duplikati?
lista	[40, 130, 480]	da	po položaju: l[0]	da
torka	(100, 50)	ne	po položaju: t[0]	da
skup	{"krug", "kvadrat"}	da	nema redosleda ni indeksa	ne
rečnik	{"nebo": (30, 40, 90)}	da	po ključu: d["nebo"]	ključ ne

Rečnik pamti redosled dodavanja, ali vrednost tražimo po ključu, ne po broju.

Metode liste menjaju istu listu

```
jelke = [40, 130, 480]
jelke.append(560)
jelke.insert(0, 10)
jelke.remove(130)
poslednja = jelke.pop()
del jelke[0]
print(jelke, poslednja)
print(jelke[-1], jelke[1:])
```

ISPIS

```
[40, 480] 560
480 [480]
```

- append: na kraj
- insert(i, x): na mesto i
- remove(x): prvo x
- pop(): uzima poslednji
- del l[i]: briše po indeksu

Rečnik kao paleta: ime boje je ključ

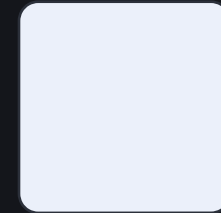
```
paleta = {"nebo": (30, 40, 90), "sneg": (235, 240, 250)}  
paleta["jelka"] = (46, 150, 67)  
print(paleta["sneg"])  
print(len(paleta), "jelka" in paleta)  
for ime, boja in paleta.items():  
    print(ime, boja)
```

ISPIS

```
(235, 240, 250)  
3 True  
nebo (30, 40, 90)  
sneg (235, 240, 250)  
jelka (46, 150, 67)
```



nebo



sneg



jelka

Ovu paletu koristimo u igri do kraja semestra.

print(*l, sep=...): zvezdica raspakuje kolekciju

```
l = [1, 2, 3]
print(l)
print(*l)
print(*l, sep="*")
print(*range(10, 20, 3), sep=", ")
d = {"ana": 10, "aca": 7}
print(*d)
```

ISPIS

```
[1, 2, 3]
1 2 3
1*2*3
10, 13, 16, 19
ana aca
```

print(*l) je isto što i print(1, 2, 3): tri posebna argumenta. sep i end su imenovani argumenti funkcije print.

Šta ispisuje ovaj program?

```
slova = "sneg"  
print(*slova, sep="-")
```

I string je kolekcija: kolekcija znakova.

A

sneg

B

tačno

s-n-e-g

C

s n e g

D

-s-n-e-g-

join spaja stringove, a map priprema elemente

```
reci = ["pada", "sneg"]  
print(" ".join(reci))  
print(", ".join(map(str, range(3))))  
brojevi = map(int, ["7", "8"])  
print(sum(brojevi))
```

ISPIS

```
pada  sneg  
0, 1, 2  
15
```

Bez map:

```
print(", ".join([1, 2, 3]))
```

PORUKA

```
TypeError:  
sequence item 0: expected  
str instance, int found
```

Imenovani argumenti: bilo kojim redom, po imenu

```
def pahuljica(x, y, velicina=3, boja="bela"):
    return f"{boja} pahuljica {velicina} px u ({x}, {y})"

print(pahuljica(10, 20))
print(pahuljica(10, 20, boja="plava"))
print(pahuljica(y=5, x=1, velicina=8))
```

ISPIS

```
bela pahuljica 3 px u (10, 20)
plava pahuljica 3 px u (10, 20)
bela pahuljica 8 px u (1, 5)
```

- velicina=3 je podrazumevana vrednost.
- boja="plava" u pozivu je imenovani argument.
- Pozicioni argumenti idu pre imenovanih.

Šta ispisuje ovaj program?

```
def tacka(x, y=0, z=0):  
    return (x, y, z)  
  
print(tacka(1, z=5))
```

A

(1, 5, 0)

B

tačno

(1, 0, 5)

C

TypeError

D

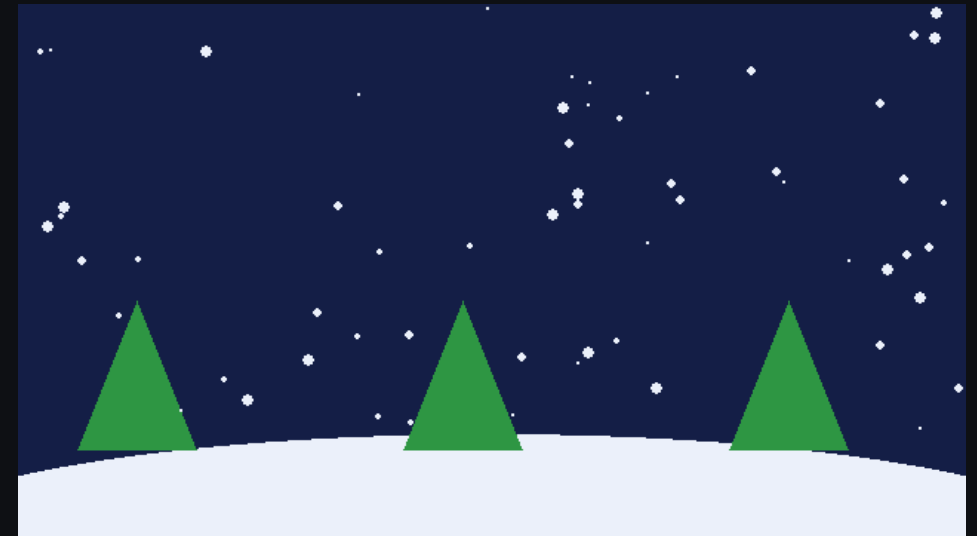
(1, 5, 5)

Paleta i lista pahuljica u prvom prozoru

Cilj: za 15 minuta podaci iz rečnika i liste postaju zimska scena.

- 1 Rečnik paleta: tri boje kao torke
- 2 Lista pahuljica: torke (x, y, r)
- 3 Petlja dodaje 60 slučajnih pahuljica
- 4 Prozor: nebo, sneg i tri jelke
- 5 Svaku pahuljicu crtamo iz torke

KAKO IZGLEDA NA KRAJU



Isto nebo, sneg i jelke vraćaju se u temi 06, kao pozadina igre.

Podaci: rečnik boja i lista torki

```
paleta = {  
    "nebo": (20, 30, 70),  
    "sneg": (235, 240, 250),  
    "jelka": (46, 150, 67),  
}  
pahuljice = [(60, 40, 3), (150, 90, 5), (260, 30, 2)]  
pahuljice.append((330, 120, 4))  
print(len(pahuljice), *pahuljice[0])  
print(*paleta, sep=", ")
```

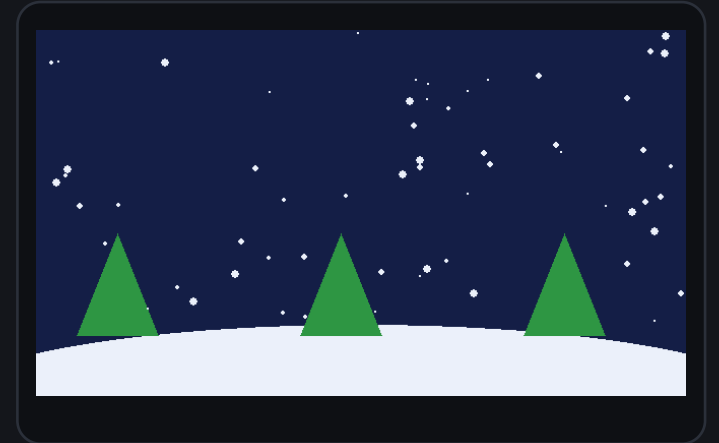
ISPIS

```
4 60 40 3  
nebo, sneg, jelka
```

- Pahuljica je torka: ne menja se, kao tačka na slici.
- Lista se menja: append dodaje novu pahuljicu.

Crtanje: svaka torka postaje jedna pahuljica

```
ekran.fill(paleta["nebo"])
pg.draw.ellipse(ekran, paleta["sneg"], (-100, 290, 840, 160))
for x in (80, 300, 520):
    vrh = (x, 200)
    temena = [vrh, (x + 40, 300), (x - 40, 300)]
    pg.draw.polygon(ekran, paleta["jelka"], temena)
for (x, y, r) in pahuljice:
    pg.draw.circle(ekran, paleta["sneg"], (x, y), r)
pg.display.flip()
```



- for (x, y, r) in pahuljice raspakuje svaku torku.
- paleta["sneg"] je torka boje.

Pronađi grešku: poeni su se sami sortirali

```
poeni = [30, 10, 20]
sortirani = poeni
sortirani.sort()
print("pre:", poeni)
print("posle:", sortirani)
```

Program treba da ispiše poene pre sortiranja i posle njega. Program se ne ruši. Šta nije u redu?

ISPIS

```
pre: [10, 20, 30]
posle: [10, 20, 30]
```

sortirani = poeni

ne pravi kopiju, nego drugo ime za istu listu. Ispravka:

sortirani = sorted(poeni)

Izaberite nivo i počnite

LEVEL 1

Moja paleta

Dodajte paleti još tri boje i ispišite svaku kao „ime: r, g, b“.

Pomoć: `items()`, `join` i `map(str, ...)`.

LEVEL 2

Pahuljice niže

Napravite novu listu u kojoj je svaka pahuljica 10 piksela niže.

Pomoć: torka se ne menja; pravite nove torke.

LEVEL 3

Paleta u prozoru

Svaku boju iz rečnika nacrtajte kao pravougaonik u prozoru.

Pomoć: `pg.draw.rect` i `x` koji raste za 105.

Sledeći put pravimo svoju klasu

```
class Pahuljica:  
    def __init__(self, x, y):  
        self.x = x  
        self.y = y  
  
p = Pahuljica(60, 40)  
print(p.x, p.y, type(p))
```

U temi 02 pahuljica više nije torka, nego objekat naše klase Pahuljica: zna gde je, koliko je velika i ume da se opiše i nacрта. `type(p)` tada kaže `<class '__main__.Pahuljica'>`.

Rezime

sve je objekat

```
type(x)    id(x)  
a == b     a is b
```

Tip, identitet i vrednost.

reference

```
b = a      # isto  
b = a.copy() # kopija
```

Dodela ne pravi kopiju.

nepromenljivo

```
t = t + (4,)
```

String i torka se ne menjaju.

ispis

```
print(*l, sep=", ")  
", ".join(map(str, l))
```

Zvezdica, join i map.

Pitanje pre odlaska

```
a = (1, 2)
b = a
a = a + (3,)
print(b)
```

Šta ispisuje program: (1, 2) ili (1, 2, 3)?

Pošaljite odgovor i jednu rečenicu zašto.



Literatura

- Lutz, M. Learning Python, 5. izdanje, O'Reilly Media, 2013 (poglavljja o tipovima i referencama)
- Wentworth, P., Elkner, J., Downey, A. B., Meyers, C. How to Think Like a Computer Scientist: Learning with Python 3 (besplatna knjiga na internetu)
- Zvanični Python tutorijal, poglavlje 5 „Data Structures“: docs.python.org/3/tutorial
- Materijali iz PZU1: teme 06, 09, 10 i 11 (stringovi, liste, kolekcije, funkcije)

Dodatna literatura i zanimljivi sajtovi

Python Tutor

pythontutor.com

Crta imena i objekte strelicama: vidi se kad dva imena pokazuju na isto.

Python: strukture podataka

docs.python.org/3/tutorial/datastructures.html

Metode liste, del, torke, skupovi i rečnici, sa primerima.

Python: objekti, vrednosti i tipovi

docs.python.org/3/reference/datamodel.html

Zvanično objašnjenje identiteta, tipa i vrednosti objekta.

pygame-ce na PyPI

pypi.org/project/pygame-ce

Zvanična stranica paketa: komanda za instalaciju i poslednja verzija.

Hvala!

Pitanja su uvek dobrodošla: na času, u četvrtu ili međim.

Slajdovi, primeri i kartica za podsećanje su na sajtu predmeta.

PREDAVANJA

dr Staša Vujičić Stanković

stasa.vujicic.stankovic@math.rs

www.matf.bg.ac.rs/~stasa.vujicic.stankovic

VEŽBE

Milan Kocić

milan.kocic@matf.bg.ac.rs

poincare.matf.bg.ac.rs/~milan.kocic