

Sveučilište u Zagrebu
Fakultet strojarstva i brodogradnje

Vježbe iz kolegija Objektno programiranje

Klase: stvaranje klase naredbom class, atributi klase, metode klase,
instance klase - objekti, atributi objekata, nasljeđivanje
(1. dio)

Mario Essert, Tihomir Žilić, Vladimir Milić, Jakov Topić

Zagreb, 2020.

1. ZADATAK Načinite klasu `Krug` koja će računati opseg i površinu kruga izvođenjem sljedećih naredbi:

```
c = Krug()
c.opseg(r)
c.povrsina(r)
```

pri čemu je varijabli `r` pridružena numerička vrijednost koja odgovara polumjeru kruga.

2. ZADATAK Načinite klasu `Gravity` koja će računati privlačnu silu između dva tijela prema Newtonovom zakonu opće gravitacije:

$$F = G \cdot \frac{m_1 \cdot m_2}{r^2} ,$$

gdje su $G = 6.67428 \cdot 10^{-11}$ [N m² kg⁻²] gravitacijska konstanta, m_1 [kg] i m_2 [kg] mase prvog i drugog tijela, r [m] međusobna udaljenost tijela.

Uputa: Klasa `Gravity` mora sadržavati metodu `__init__` za inicijalizaciju objekta. Također, mora imati metodu `force` čijim ćete lokalnim varijablama pridružiti klasne atribute (G , m_1 , m_2), a metoda će vraćati iznos privlačne sile F u ovisnosti od zadanog r .

Zadano:

```
masa = {'Sunce': 1.99E+30, 'Zemlja': 5.97E+24,
        'Mjesec': 7.35E+22, 'raketa': 7.57E+5}

udaljenost = {('Zemlja', 'raketa'): 6.6E+6,
              ('Zemlja', 'Sunce'): 1.5E+11,
              ('Zemlja', 'Mjesec'): 3.85E+8}

lst = ['raketa', 'Mjesec', 'Sunce']

for i in range(len(lst)):
    tpl = 'Zemlja', lst[i]
    r = udaljenost[(tpl[0], tpl[1])]
    gravitacija = Gravity(...) #argumenti za pozivanje konstruktora
    print(gravitacija) # poziv __str__() metode
    F = ... # racunanje sile privlacenja
    print ... # ispisivanje trazenog rjesenja
```

Tražena rješenja:

```
>>>
Gravitacija: ZEMLJA - RAKETA
Sila između Zemlja i raketa: 6.92447e+06 N
Gravitacija: ZEMLJA - MJESEC
Sila između Zemlja i Mjesec: 1.97581e+20 N
Gravitacija: ZEMLJA - SUNCE
Sila između Zemlja i Sunce: 3.52411e+22 N
```

3. ZADATAK Načinite klasu koja će numerički računati prvu derivaciju polinoma

$$f(t; a, b, c) = at^2 + bt + c,$$

primjenom formule za konačne diferencije

$$\frac{df}{dt} \approx \frac{f(t+h) - f(t)}{h}.$$

Uputa:

1. Načinite klasu `Diff` koja će sadržavati metodu `__init__` za inicijalizaciju objekta s parametrima inicijalizacije `f` i `h`. Preddefinirana vrijednost atributa `h` neka bude 10^{-4} .
2. Unutar klase `Diff` načinite metodu `__call__` koja će vraćati vrijednost prve derivacije primjenom formule za konačne diferencije.
3. Načinite klasu `Poly` za stvaranje polinoma koja će sadržavati metode `__init__` i `__call__`, a parametri inicijalizacije će joj biti koeficijenti polinoma `a`, `b`, `c`.

Na primjer izvršavanjem sljedećih naredbi:

```
p = Poly(5, 2, 1)
dpdt = Diff(p)
t = 1
print("Koeficijenti polinoma a*t^2 + b*t + c su a=%g, b=%g, c=%g" \
      % (p.a, p.b, p.c))
print("Vrijednost polinoma u točki t=%g iznosi p(t)=%g" % (t, p(t)))
print("Derivacija polinoma u t=%g iznosi %g" % (t, dpdt(t)))
```

rezultat je:

```
>>>
Koeficijenti polinoma a*t^2 + b*t + c su a=5, b=2, c=1
Vrijednost polinoma u točki t=1 iznosi p(1)=8
Derivacija polinoma u t=1 iznosi 12.0005
>>>
```

4. ZADATAK Neka su zadane klase A i B na sljedeći način:

```
class A:
    varA = 42
    varB = 4.2
    def f1(self):
        print("Class A : f1")

class B:
    varB = 37
    varC = 1.3
    def f1(self):
        print("Class B : f1")
    def f2(self):
        print("Class B : f2")
    def f3(self):
        print("Class B : f3")
```

4.1. Načinite klasu C koja nasljeđuje klase A i B, te ima varijable varC=3.3, varA=88, i metodu f2 koja ispisuje "Class C : f2".

a) Pozovite attribute f1, f2 i f3 direktno iz klase C.

```
>>> Class A : f1
>>> Class C : f2
>>> Class B : f3
```

b) Stvorite objekt iz klase C, te iz njega pozovite metode f1, f2 i f3.

```
>>> Class A : f1
>>> Class C : f2
>>> Class B : f3
```

c) Pozovite varijable varA, varB i varC iz klase C.

```
>>> 88
>>> 4.2
>>> 3.3
```

4.2. Načinite klasu D koja nema niti jednu metodu te klasu E koja nasljeđuje klase C i D. Definirajte objekte d i e iz klasa D i E.

a) Ispitajte da li objekt d pripada klasi A te da li objekt e pripada klasama C i A.

```
>>> False
>>> True
>>> True
```

b) Ispitajte da li je A podklasa od E, E podklasa od D i C podklasa od A.

```
>>> False
>>> True
>>> True
```

5. ZADATAK Načinite klasu `Racunalo` koja ima metode `__init__`, `postavi_os` i `koji_os`. Zatim načinite klasu `ThinkPad` (koja je potomak klase `Racunalo`) te ima svoju metodu `postavi_model`. Tada stvorite glavnu funkciju `main` koja će biti pokrenuta ako se skripta pokreće kao samostalni program (`__name__=="__main__"`). U glavnoj funkciji ćete stvoriti objekt iz klase `ThinkPad` te ga promijeniti kako je prikazano ispod u kôdu.

Nakon izvršavanja sljedeće .py skripte:

```
class Racunalo:
    ...
class ThinkPad(Racunalo):
    ...

def main():
    t = ThinkPad("zlatna", "Lenovo")
    Model = t.postavi_model("L")
    t.postavi_os("Linux")
    print(Model)
    print("Proizvodjac: %s, boja: %s, OS: %s" % \
          (t.proizvodjac, t.boja,t.koji_os()))

if __name__=='__main__':
    main()
```

ispis na ekranu je sljedeći:

```
>>>
Ovo je ThinkPad model: L
Proizvodjac: Lenovo, boja: zlatna, OS: Linux
```