

Vežbe 6 - Zadaci

Zadatak 1. Realizovati klasu trougao sa atributima stranica a, b i c. Klasa Trougao sadrži metode površina.trougla (određuje površinu trougla Heronovim formulom) i obim.trougla. Reimplementirati destruktor, kao i metode za proveru jednakosti i `|` operator (provera se vrši po površini). Naslediti klasu trougao u JKKTrougao (jednakokraki) i JKSTrougao(jednakostranici trougao).

```
1 import math
2
3 class Trougao():
4     def __init__(self, a, b, c):
5         self.a = a
6         self.b = b
7         self.c = c
8
9     def površina_trougla(self):
10        s = self.obim_trougla()
11        P = math.sqrt(s * (s-self.a) * (s-self.b) * (s-self.c))
12        return P
13
14    def obim_trougla(self):
15        return self.a + self.b + self.c
16
17    def __lt__(self, other):
18        return self.površina_trougla() < other.površina_trougla()
19
20    def __eq__(self, other):
21        return self.obim_trougla() == other.obim_trougla()
22
23    def __del__(self):
24        print("Objekat je unisten.")
25
26
27 class JKKTrougao(Trougao):
28     def __init__(self, a, b):
29         super().__init__(a,b,b)
30
31 class JKSTrougao(JKKTrougao):
32     def __init__(self, a):
33         super().__init__(a,a)
34
35
36 trougao = Trougao(3,4,5)
37 print("Površina trougla:")
38 print(trougao.površina_trougla())
39 print("Obim trougla:")
40 print(trougao.obim_trougla())
41
```

```

42 jkktrougao = JKKTrougao(2,5)
43 print("Povrsina jednakokrakog trougla:")
44 print(jkktrougao.povrsina_trougla())
45 print("Obim trougla:")
46 print(jkktrougao.obim_trougla())
47
48 jkstrougao = JKSTrougao(5)
49 print("Povrsina jednakostranicnog trougla:")
50 print(jkstrougao.povrsina_trougla())
51 print("Obim jednostranicnog trougla:")
52 print(jkstrougao.obim_trougla())
53
54 if jkktrougao < trougao:
55     print("Jednakokraki trougao ima manju povrsinu od opsteg trougla.")
56 else:
57     print("Opsti trougao ima manju povrsinu od jednakokrakog trougla.")
58
59 if jkstrougao == trougao:
60     print("Jednakostranicni trougao ima isti obim kao opsti trougao.")
61 else:
62     print("Jednakostranicni trougao nema isti obim kao opsti trougao.")
63
64 del jkktrougao
65 del trougao
66 del jkstrougao

```

Zadatak 2. Realizovati klasu Krug sa atributom r metodama površina_kruga i povrsina_kruga, kao i klasu Pravougaonik sa atributima a i b stranicom sa metodama obim_kruga i povrsina_kruga. Korišćenjem višestrukog nasleđivanja, realizovati klasu Valjak sa metodama površina_valjka i dijagonala_valjka. Atributi u klasama su privatni.

```

1     import math
2
3     class Krug:
4         def __init__(self, r):
5             self.__r = r
6
7         def obim_krug(self):
8             return 2*math.pi*self.__r
9
10        def povrsina_krug(self):
11            return self.__r**2*math.pi
12
13        def poluprecnik(self):
14            return self.__r
15
16    krug = Krug(5)
17    print("Obim kruga:")

```

```

18     print(krug.obim_krug())
19     print("Povrsina kruga:")
20     print(krug.povrsina_krug())
21
22     class Pravougaonik:
23         def __init__(self, a, b):
24             self.__a = a
25             self.__b = b
26
27         def obim_pravougaonik(self):
28             return 2*self.__a + 2*self.__b
29
30         def povrsina_pravougaonik(self):
31             return self.__a * self.__b
32
33         def stranica_a(self):
34             return self.__a
35
36         def stranica_b(self):
37             return self.__b
38
39
40     class Valjak(Krug, Pravougaonik):
41         def __init__(self, r, a, b):
42             Krug.__init__(self, r)
43             Pravougaonik.__init__(self, a, b)
44
45         def povrsina_valjak(self):
46             # Omotac + 2* Baza
47             if self.stranica_a() != self.obim_krug():
48                 return None
49             else:
50                 return 2*self.povrsina_krug() + self.povrsina_pravougaonik()
51
52         def dijagonala_valjka(self):
53             a = self.poluprecnik()
54             h = self.stranica_b()
55             d = math.sqrt(a**2 + h**2)
56             return d
57
58         def zapremina_valjak(self):
59             return self.povrsina_pravougaonik() * self.stranica_b()
60
61         def visina_valjka(self):
62             return self.stranica_b()
63
64
65     valjak = Valjak(5, krug.obim_krug(), 7)

```

```

66
67     print("Povrsina valjka:")
68     print(valjak.povrsina_valjak())
69     print("Dijagonala valjka:")
70     print(valjak.dijagonala_valjka())
71     print("Zapremina valjka:")
72     print(valjak.zapremina_valjak())
73     print("Visina valjka:")
74     print(valjak.visina_valjka())

```

Zadatak 3. Realizovati klasu Kamera sa atributima mega_piksel (broj megapiksela kamere tj. prostor koji slika zauzima), free_space (količina slobodnog prostora) i photo_counter (broj slika) i metodama slikaj i briši. Realizovati klasu Telefon sa atributom broj i metodom pozovi. Realizovati klasu Telefon_Gen_2 koja nasleđuje klase Telefon i Kamera sa njihovim funkcionalnostima.

```

1  class Kamera:
2      def __init__(self, mega_piksel, free_space):
3          self.mega_piksel = mega_piksel
4          self.free_space = free_space
5          self.photo_counter = 0
6
7      def slikaj(self):
8          if self.free_space < self.mega_piksel:
9              return False
10         else:
11             self.free_space -= self.mega_piksel
12             self.photo_counter += 1
13             return True
14
15     def brisi(self):
16         if self.photo_counter <= 0:
17             return False
18         else:
19             self.free_space += self.mega_piksel
20             self.photo_counter -= 1
21             return True
22
23     def broj_slika(self):
24         return self.photo_counter
25
26     def preostale_slike(self):
27         return self.free_space//self.mega_piksel
28
29 class Telefon:
30     def __init__(self, broj):
31         self.broj = broj
32
33     def pozovi(self, poziv):

```

```

34         if poziv == self.broj:
35             return False
36         else:
37             return True
38
39 class Telefon_Gen_2(Kamera, Telefon):
40     def __init__(self, mega_piksel, free_space, broj):
41         Kamera.__init__(self, mega_piksel, free_space)
42         Telefon.__init__(self, broj)
43
44 kamera = Kamera(4, 12)
45 print(kamera.slikaj())
46 print(kamera.slikaj())
47 print(kamera.broj_slika())
48 print(kamera.preostale_slike())
49
50 telefon = Telefon(123456)
51 print(telefon.pozovi(654321))
52 print(telefon.pozovi(123456))
53
54 telefon2 = Telefon_Gen_2(10, 55, 123987)
55 print(telefon2.pozovi(123987))
56 print(telefon2.pozovi(123456))
57 print(telefon2.slikaj())
58 print(telefon2.broj_slika())
59 print(telefon2.preostale_slike())

```