

Računalništvo (fizika): 3. izpit

9. september 2020

Naloge rešujte na strežniku Tomo, <https://www.projekt-tomo.si>.

Brskanje po spletni učilnici, nalogah na strežniku Tomo, video zapiskih predavanj in *uradni* dokumentaciji za Python *je dovoljeno*. Brskanje po spletnih straneh, ki vsebujejo rešitve nalog in nasvete o tem, kako se rešuje naloge, *ni dovoljeno*. Komunikacija s komerkoli, razen z asistentom, *ni dovoljena*.

Čas reševanja je 180 minut. Doseženih 100 točk šteje za maksimalno oceno. Veliko uspeha!

1. naloga (25 točk)

Aljaž se je odločil obogateti s prodajo programske opreme za trgovine. V ta namen cenik izdelkov predstavi s slovarjem, katerega ključi so izdelki in vrednosti pripadajoče cene, na primer

```
mercator = {  
    'mleko' : 1.20,  
    'kruh'  : 2.20,  
    'jajce' : 0.50  
}
```

pove, da v Mercatorju steklenica mleka stane 1,20, kruh 2,20 in eno jajce 0,50 evra. Nakup predstavi s seznamom parov $[(a_1, n_1), \dots, (a_k, n_k)]$, ki pove, da smo kupili n_i kosov izdelka a_i . Na primer, če kupimo 2 steklenici mleka in 6 jajc, to predstavimo s seznamom $[('mleko', 2), ('jajce', 6)]$.

Sestavite funkcijo `vrednost(cenik, nakup)`, ki sprejme cenik in nakup ter vrne skupno vrednost nakupa. Primer uporabe:

```
>>> vrednost(mercator, [('mleko', 2), ('jajce', 6)])  
5.4
```

2. naloga (25 točk)

V računalniški igri je stanje igralca predstavljeno z močjo, spretnostjo in domiselnostjo. Igralec lahko izbere eno od treh vlog:

- *bojevnik* začne igro z močjo 15, spretnostjo 10 in domiselnostjo 5;
- *zmikavt* začne igro z močjo 5, spretnostjo 15 in domiselnostjo 10;
- *vila* začne igro z močjo 10, spretnostjo 5 in domiselnostjo 15.

a) (10 točk) Sestavite razred `Igralec`, ki predstavlja igralca. Konstruktor sprejme vlogo (enega izmed nizov "bojevnik", "zmikavt" ali "vila") in primerno nastavi začetne lastnosti igralca. Razredu dodajte tudi ustrezni metodi `__str__` in `__repr__`, ki prikažeta vlogo in stanje igralca. Primer:

```
>>> Igralec('bojevnik')
Igralec('bojevnik')
>>> str(Igralec('bojevnik'))
'bojevnik[moc=15, spretnost=10, domiselnost=5]'
```

b) (15 točk) Igralec lahko med igro uporabi razne predmete, ki vplivajo na njegove lastnosti. Predmet predstavimo z objektom, ki ima metodo `vpliv(self)`. Le-ta vrne trojico (m, s, d) , ki določa, kako predmet spremeni lastnosti igralca. Primer:

```
class Kruh():
    def vpliv(self): return (10, 0, 1)

class CarobniNapoj():
    def vpliv(self): return (-5, 1, 10)
```

Razredu `Igralec` dodajte metodo `uporabi(self, p)`, ki spremeni stanje igralca, ko ta uporabi predmet `p`. Primer:

```
>>> aljaz = Igralec('zmikavt')
>>> print(aljaz)
zmikavt[moc=5, spretnost=15, domiselnost=10]
>>> aljaz.uporabi(Kruh())
>>> print(aljaz)
zmikavt[moc=15, spretnost=15, domiselnost=11]
```

3. naloga (30 točk)

Neskončno potenčno vrsto $\sum_{n=0}^{\infty} a_n x^n$ lahko v Pythonu predstavimo z iteratorjem, ki generira koeficiente $a_0, a_1, a_2, \dots$. Na primer, $\exp x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$ predstavimo z

```
def exp():
    (n, a) = (1, 1)
    while True:
        yield a
        (n, a) = (n+1, a*n)
```

a) (10 točk) Definirajte generator `sin`, s katerim predstavite potenčno vrsto

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

Primer uporabe:

```
>>> s = sin()
>>> [next(s) for i in range(7)]
[0, 1.0, 0, -0.16666666666666666, 0, 0.008333333333333333, 0]
```

b) (10 točk) Definirajte generator `odvod(a)`, ki sprejme generator `a` za vrsto $\sum_{n=0}^{\infty} a_n x^n$ ter vrne generator za njen odvod $\sum_{n=1}^{\infty} n a_n x^{n-1}$. Primer uporabe:

```
>>> a = odvod(odvod(sin()))
>>> [next(a) for i in range(7)]
[0, -1.0, 0, 0.16666666666666666, 0, -0.008333333333333333, 0]
```

c) (10 točk) Definirajte generator `vsota(a, b)`, ki sprejme generatorja `a` in `b` za vrsti $\sum_{n=0}^{\infty} a_n x^n$ in $\sum_{n=0}^{\infty} b_n x^n$ ter vrne generator za njuno vsoto $\sum_{n=0}^{\infty} (a_n + b_n) x^n$. Primer uporabe:

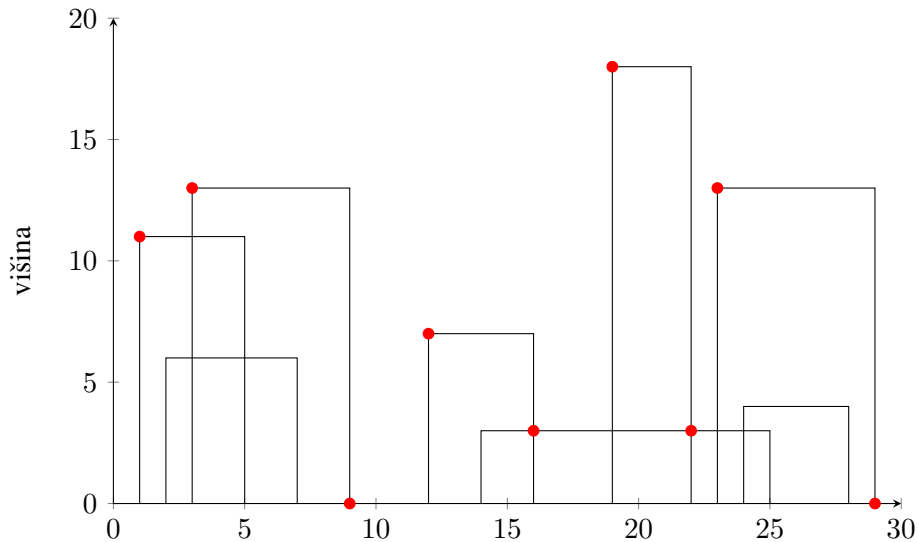
```
>>> v = vsota(sin(), odvod(odvod(sin())))
>>> [next(v) for i in range(10)]
[0, 0.0, 0, 0.0, 0, 0.0, 0, 0.0, 0, 0.0]
```

4. naloga (30 točk)

Profesor Bauer se v prostem času ukvarja z računalniško umetnostjo. V želji, da bi prodal vsaj kakšno sliko, želi sestaviti program, ki nariše silhueto mesta. Podatke o stavbah vnese kot seznam trojic (ℓ, v, d) , kjer sta ℓ in d leva in desna koordinata stavbe ter v njena višina. Na primer, seznam

$[(1, 11, 5), (2, 6, 7), (3, 13, 9), (12, 7, 16), (14, 3, 25), (19, 18, 22), (23, 13, 29), (24, 4, 28)]$

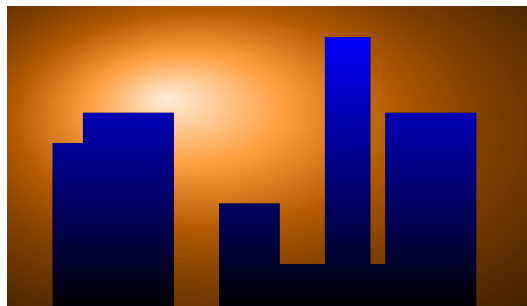
predstavlja stavbe, kot je prikazano na skici:



Profesor Bauer želi izračunati koordinate točk (na sliki obarvane rdeče), v katerih se višina silhuete spremeni:

$[(1, 11), (3, 13), (9, 0), (12, 7), (16, 3), (19, 18), (22, 3), (23, 13), (29, 0)]$

Iz danega seznama točk nariše sliko:



a) (15 točk) Sestavite funkcijo `silhueta`, ki sprejme seznam stavb in vrne seznam točk, ki določajo silhueto. Primer:

```
>>> silhueta([(1, 11, 5), (2, 6, 7), (3, 13, 9), (12, 7, 16), (14, 3, 25),  
             (19, 18, 22), (23, 13, 29), (24, 4, 28)])  
[(1, 11), (3, 13), (9, 0), (12, 7), (16, 3), (19, 18), (22, 3),  
 (23, 13), (29, 0)]
```

b) (15 točk) Poskrbite, da funkcija `silhueta` deluje učinkovito, to je v času $\mathcal{O}(n \log n)$, kjer je n število stavb v mestu.

Namig: kako zlijemo dve silhueti? Sestavite pomožno funkcijo, ki to naredi.