Programmeerimise süvendatud algkursus ITI0140

2014

Teemad

Andmehulkade hoidmine:

- ennikutes (*tuple*)
- (mitmemõõtmelistes) järjendites (*list*)
- hulkades (*set*)
- sõnastikes (*dictionary*)

Ennikud *(tuple)*

- Sisuliselt muutumatu järjend, st peale defineerimist elemente lisada, muuta ega eemaldada ei saa

```
t = 12345, 54321,
'hello!'
print(t)
print(t[1])
u = t, 1, 2, 3
print(u)
print(u[0][2])
t[1] = 8
```

```
(12345, 54321, 'hello!')
54321
((12345, 54321, 'hello!'), 1, 2, 3)
hello!
Traceback (most recent call last):
  File "p.py", line 7, in <module>
    t[1] = 8
TypeError: 'tuple' object does not
support item assignment
```

Operatsioonid järjenditega

```
a = [2, 5, 1, 0, 5]
a.append(3)
a.extend([1, 7])
a.insert(3, 2)
a.remove(1)
a.pop()
a.pop(5)
a.index(5)
a.count(2)
a.sort()
a.reverse()
del a[4]
del a[2:4]
```

```
[2, 5, 1, 0, 5]
[2, 5, 1, 0, 5, 3]
[2, 5, 1, 0, 5, 3, 1, 7]
[2, 5, 1, 2, 0, 5, 3, 1, 7]
[2, 5, 2, 0, 5, 3, 1, 7]
[2, 5, 2, 0, 5, 3, 1]
[2, 5, 2, 0, 5, 1]
1
2
[0, 1, 2, 2, 5, 5]
[5, 5, 2, 2, 1, 0]
[5, 5, 2, 2, 0]
[5, 5, 0]
```

Hulgad (*set*)

- Kasulikud unikaalsete elementide hoidmiseks

```
a = set([1, 2, 5, 2, 3, 3, 0])
a.add(8)
a.add(8)
a.remove(5)
```

```
{0, 1, 2, 3, 5}
{0, 1, 2, 3, 5, 8}
{0, 1, 2, 3, 5, 8}
{0, 1, 2, 3, 8}
```

Operatsioonid hulkadega

```
a = set("kala")
b = set("lammas")
print(a)
print(b)
# vahe
print(a - b)
# ühend
print(a | b)
# ühisosa
print(a & b)
# ühend ühisosata
print(a ^ b)
print((a | b) - (a & b))
```

```
{'k', 'a', 'l'}
{'s', 'a', 'm', 'l'}
{'k'}
{'s', 'a', 'k', 'm', 'l'}
{'a', 'l'}
{'s', 'k', 'm'}
{'s', 'k', 'm'}
```

Sõnastikud (*dictionary*)

- Kasulik vastavustabelite loomiseks

```
telefonid = {
   'politsei' : '110',
   'päästeamet' : '112'
}
print(telefonid)
print(telefonid['politsei'])
telefonid['ttü'] = '620 2002'
print(telefonid)
del telefonid['politsei']
print(telefonid)
print(telefonid.items())
print(telefonid.values())
print(telefonid.keys())
```

```
{'politsei': '110', 'päästeamet':
'112'}


110


{'politsei': '110', 'päästeamet':
'112', 'ttü': '620 2002'}


{'päästeamet': '112', 'ttü': '620
2002'}


dict_items([('päästeamet', '112'),
('ttü', '620 2002')])


dict_values(['112', '620 2002'])


dict_keys(['päästeamet', 'ttü'])
```

Mitmemõõtmelised järjendid

```
a = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
print(a)
print(a[1])
print(a[1][2])
a[2][1] = 10
print(a)
a[1] = [2, 3]
print(a)
```

```
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
[4, 5, 6]
6
[[1, 2, 3], [4, 5, 6], [7, 10, 9]]
[[1, 2, 3], [2, 3], [7, 10, 9]]
```

Ülesanne: varandusejaht

Kesine õppetoetus motiveerib teid lisaraha teenima ning läbi õnneliku juhuse saate kursavenna käest laenata metallidetektori. Paraku on laenutingimuseks, et te jagate leitud aarded pooleks. Teie sõbra arvates on aus, kui te kõigepealt kaardistate metallidetektori abil kõik aarded ja siis hiljem lähete koos kaardi alusel väljakaevamisi tegema. Nii ei pidavat te saama tal nahka üle kõrvade tõmmata. Kursavend teab rääkida, et metallidetektor piniseb lineaarselt kõvemini mida enam aardeid lähedal on.

Teha funktsioon ***treasure_map***, mis võtab vastu kaks argumenti:

- kaheelemendilise enniku, mis määrab tabeli suuruse (***width***, ***height***)
- järjendi kaheelemendilistest ennikutest, mis defineerivad tabeli asukohad, kus paiknevad aarded ja tagastab kahemõõtmelise järjendi, kus on igal positsioonil:
  - 'X', kui seal paikneb aare
  - number, mis näitab mitu aaret selle ruudu vahetus läheduses on (st kokku lugeda kaheksas suunas)

Teha funktsioon ***print_treasure***, mis trükib ***treasure_map*** funktsiooni tagastatava tabeli välja.

print_treasure(treasure_map((4, 5), [(2, 0), (1, 1), (2, 2), (1, 3)]))

```
12X1
1X32
23X1
1X21
1110
```