Seznamy

Jiří Zacpal

KATEDRA INFORMATIKY
UNIVERZITA PALACKÉHO V OLOMOUCI

KMI/ZPP1 Základy programování v Pythonu 1

Řešení úkolu z minulé hodiny

```
import re

cislo=input("Zadej římské číslo: ")
cislice="MDCLXVI"

vzor="^M{0,3}(CM|CD|D?C{0,3})(XC|XL|L?X{0,3})(IX|IV|V?I{0,3})$"
test=re.search(vzor,cislo)
```

Řešení úkolu z minulé hodiny

```
if test!=None:
    i=0
    c=0
    v=0
    delka=len(cislo)
    while i<delka:
        if cislo[i]==cislice[c]:
            match cislo[i]:
                case "M":v+=1000
                case "D":v+=500
                case "C":v+=100
                case "L":v+=50
                case "X":v+=10
                case "V":v+=5
                case "I":v+=1
            i+=1
```

Řešení úkolu z minulé hodiny

```
        elif cislo[i]==cislice[c+1] and cislo[i+1]==cislice[c]:
            match cislo[i+1]:
                case "M":v+=900
                case "D":v+=400
                case "C":v+=90
                case "L":v+=40
                case "X":v+=9
                case "V":v+=4
            i+=2
        else:
            c+=1

    print(v)
else:
    print("Řetězec není římské číslo.")
```

Seznamy

Motivace

- Chceme napsat program, který vylosuje u zkoušky studentovi otázku:

```
import random
pocet=12
los=True
while los:
    volba=input('Další student (zadej "a"):')
    if volba==„a":
        o=random.randint(1,pocet)
        print("Vylosována otázka číslo",o,".")
    else:
        los=False
```

Motivace

- Další student nesmí dostat stejnou otázku:

```
import random
pocet=12
los=True
h=-1
while los:
    volba=input('Další student (zadej "a"):')
    if volba=="a":
        while True:
            o=random.randint(1,pocet)
            if h!=o:break
        h=o
        print("Vylosována otázka číslo",o,".")
    else:
        los=False
```

Motivace

- Chceme, aby se otázky v průběhu zkoušky neopakovaly.
- Jak to udělat?
- Proměnné?
- Potřebujeme datovou struktur, ze které bychom otázky losovali = seznam.

Vytvoření seznamu

- Seznam je uspořádaná kolekce položek.
- Syntaxe:

```
[první_položka_seznamu, druhá_položka_seznamu, ..., poslední_položka_seznamu]
```

- Položky seznamu jsou reference na (libovolné) objekty, které reprezentují tyto položky.
- Seznam je mutabilní.
- Příklady:

  L = [1, 2, 3, 4]

  L = ["s", "p", "a", "m"]

  L = ["spam", "ham"]

  L = ["spam", 2, 3, 4]

Počet prvků v seznamu

- Pro získání počtu položek v seznamu použijeme funkci:

```
len(seznam)
```

  Funkce vrátí počet položek v seznamu.
- Příklad:

```
>>> len([1,2,3,4])
4
>>> len([])
0
```

Seznam jako kolekce

- Seznamy jsou mutabilní kolekce.
- Pro práci s nimi tedy můžeme použít cyklus for:

```
s=[1,2,3,4,5]

for i in s:
    print(i)
)
```

Operátor indexu

`s[i]`

- Operátor vrátí i-tou položku ze seznamu s.
- Příklad:

```
>>> [1,2,3,4][0]
1
>>> s=[1,2,3,4]
>>> s[3]
4
>>> s[-1]
4
>>> s[5]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
IndexError: list index out of range
```

Příklad

- S pomocí cyklu for vypíšeme všechny položky ze seznamu:

```
s=[1,2,3,4,5]

for i in range(len(s)):
    print(s[i])
```

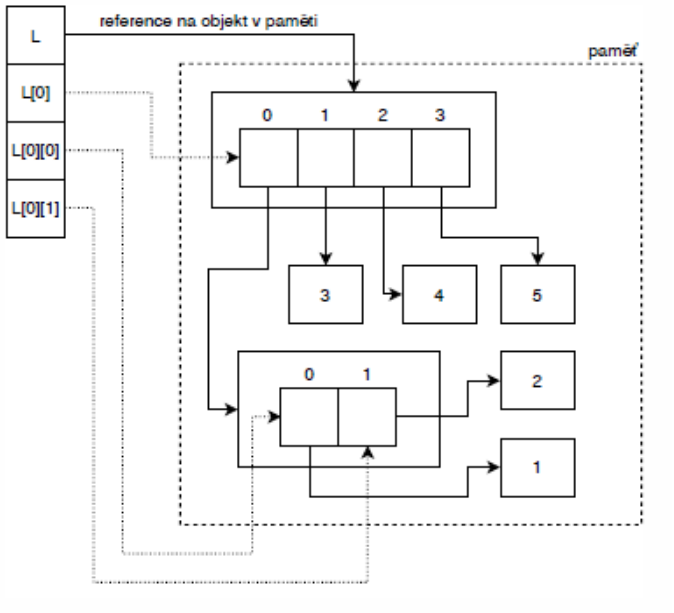
Příklad

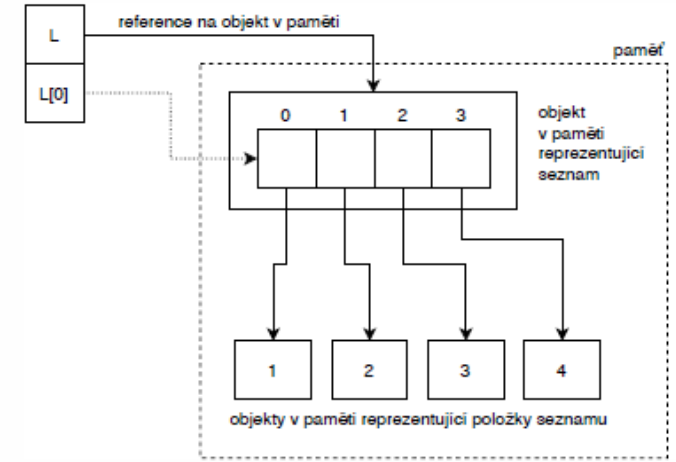
```
s=[]
for i in range(0,10):
    s.append(int(input("Zadej teplotu:")))

soucet=0
for i in s:
    soucet+=i

print(f"Průměrná teplota je {soucet/len(s):.2f}")

max=s[0]
for i in range(1,len(s)):
    if s[i]>max:
        max=s[i]

print(f"Nejvyšší teplota je {max}")
```

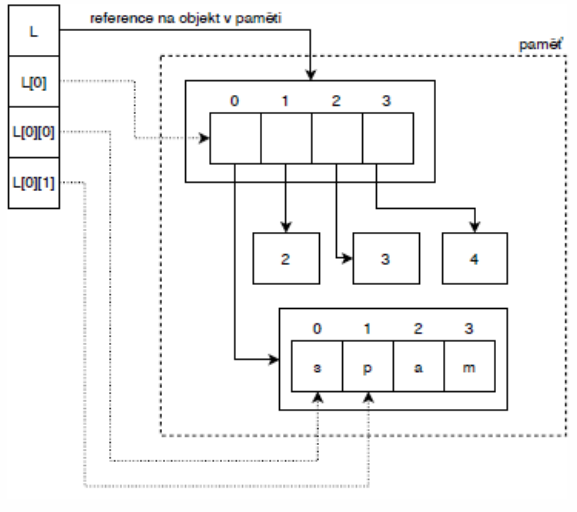
Uložení seznamů v paměti

- Seznam je tvořen ukazateli na hodnoty.
- Příklad:

```
L = [1, 2, 3, 4]
print(L[1]**L[2]) # vypíše: 8
L = ["spam", 2, 3, 4]
L = [[1, 2], 3, 4, 5]
```







Proměnlivost seznamů

Přidávání položek do seznamu

- Můžeme použít tyto způsoby:
  - použít operátor +:

```
>>> s=[1,2,3]
>>> s=s+[4,5,6]
>>> s
[1, 2, 3, 4, 5, 6]
```

  - použít metodu

```
append(položka)
```

    která na konec seznamu přidá jednu položku:

```
>>> s.append(7)
>>> s
[1, 2, 3, 4, 5, 6, 7]
```

Přidávání položek do seznamu

- použít metodu

```
extend(seznam)
```

která na konec seznamu přidá seznam, uvedený jako argument

```
>>> s.extend([8,9,10])
>>> s
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

- použít insert

```
insert(kam, položka)
```

která na místo s indexem kam vloží do seznamu položku a zbytek prvků posune doprava

```
>>> s.insert(1,1.5)
>>> s
[1, 1.5, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

Mutabilita seznamu

- Seznamy jsou na rozdíl od číselných sekvencí a řetězců mutabilní.
- Jejich jednotlivé položky lze měnit.
- Příklad:

```
L = [1, 2, 3, 4, 5]
L[0] = 42
print(L) # vypíše: [42, 2, 3, 4, 5]
```

- Mutabilita přináší drobnou komplikaci. Ke změně objektu může dojít skrze různé reference.
- Příklad:

```
L = [1, 2, 3, 4, 5]
M = L # M nyní obsahuje referenci na L
M[0] = 42 # změní referenci v L
print(L) # vypíše: [42, 2, 3, 4, 5]
```

- V následujícím kódu výše uvedené nenastává:

```
L = [1, 2, 3]
M = [4, 5]
N = [L, M]
M = 42
print(N) # vypíše: [[1, 2, 3], [4, 5]]
```

Odstranění položky ze seznamu

- Můžeme použít metodu

  `.remove(hodnota)`

  odstraní ze seznamu první výskyt objektu s hodnotou hodnota

```
>>> s=[1,2,2,3,4,1,5]
>>> s.remove(1)
>>> s
[2, 2, 3, 4, 1, 5]
```

- Metoda

  `.clear()`

  odstraní všechny položky ze seznamu

```
>>> s=[1,2,2,3,4,1,5]
>>> s.clear()
>>> s
[]
```

Odstranění položky ze seznamu

- Také můžeme použít metodu

  `.pop(index)`

  odstraní položka seznamu na daném indexu (pokud argument index neuvedeme, odstraní se první položka seznamu) a odstraněná položka se vrátí

```
>>> s=[1,2,2,3,4,1,5]
>>> s.pop(4)
4
>>> s
[1, 2, 2, 3, 1, 5]
>>> s.pop()
5
>>> s
[1, 2, 2, 3, 1]
```

Motivace

- Chceme, aby se otázky v průběhu zkoušky neopakovaly:

```
import random
pocet=12
los=True
osudi=[]
for i in range(1,pocet+1):
    osudi+=[i]

while los:
    volba=input('Další student (zadej "a"):')
    if volba=="a":
        if len(osudi)==0:
            print("Už není žádná další otázka.")
            los=False
        else:
            o=osudi.pop(random.randint(1,len(osudi))-1)
            print("Vylosována otázka číslo",o,".")
    else:
        los=False
```

Motivace

- Chceme si uchovat historii vylosovaných otázek:

```
import random
pocet=12
los=True
osudi=[]
for i in range(1,pocet+1):
    osudi+=[i]

h=[]
while los:
    volba=input('Další student (zadej "a"):')
    if volba=="a":
        if len(osudi)==0:
            print("Už není žádná další otázka.")
            los=False
        else:
            o=osudi.pop(random.randint(1,len(osudi))-1)
            print("Vylosována otázka číslo",o,".")
            h+=[o]
    else:
        los=False

print("Dnes byly vylosovány otázky v tomto pořadí:",h)
```

Další metody

- Metoda

  `.reverse()`

  obrátí pořadí položek v seznamu

```
>>> s=[1,2,3,4,5]
>>> s.reverse()
>>> s
[5, 4, 3, 2, 1]
```

- Metoda

  `.count(hodnota)`

  vrátí počet položek s hodnotou hodnota

```
>>> s=[1,2,2,3,4,1,5]
>>> s.count(2)
2
```

Další metody

- Metoda

  `.copy()`

  vytvoří kopii seznamu

```
>>> s=[1,2,3,4,5]
>>> r=s.copy()
>>> r
[1, 2, 3, 4, 5]
>>> s[0]='x'
>>> s
['x', 2, 3, 4, 5]
>>> r
[1, 2, 3, 4, 5]
>>>
```

Logické operace se seznamy

Porovnání totožnosti

- Pokud v1 a v2 jsou výrazy, pak

```
v1 is v2
```

- je výraz porovnání totožnosti.
- Totožnost nás zajímá pouze u seznamu – ty jediné umíme měnit, proto budeme předpokládat, že hodnoty v1 a v2 jsou seznamy.
- Hodnota výrazu porovnání totožnosti je pravda, pokud seznamy v1 a v2 jsou totožné, jinak je hodnota nepravda.

Příklad

- Podívejme se na příklad:

```
>>> l1 = [1, 2, 3]
>>> l2 = l1
>>> l3 = [1, 2, 3]
>>> l1 is l2
True
>>> l1 is l3
False
```

- Vidíme, že seznamy l1 a l3 nejsou totožné a to i přesto, že jsou si všechny seznamy rovny:

```
>>> l1 == l2
True
>>> l2 == l3
True
```

Příklad

- Ve skutečnosti l1 a l2 jsou různé názvy pro tu samou hodnotu, l3 je jiná hodnota, která je pouze rovna (ekvivalentní) seznamům l1 a l2.
- Proto změna l1 změní i hodnotu l2 ale nezmení hodnotu l3:

```
>>> l1 [1] = 5
>>> l1
[1, 5, 3]
>>> l2
[1, 5, 3]
>>> l3
[1, 2, 3]
```

Příklad

- Pokud bychom chtěli hodnotu l1 změnit a neovlivnit hodnotu l2, musíme udělat kopii hodnoty l1.
- Toho lze docílit například řezem:

```
>>> l1 = [1, 2, 3]
>>> l2 = l1.copy()
>>> l1 [1] = 5
>>> l1
[1, 5, 3]
>>> l2
[1, 2, 3]
```

Test přítomnosti položky v seznamu

- Pokud chceme zjistit, zda se daná položka nachází respektive nenachází v seznamu, můžeme použít operátory

```
s in h
s not in h
```

který vrací True (False), jestli položka h je (není) v seznamu s.

```
>>> s=[1,2,3,4,5]
>>> 4 in s
True
>>> 10 not in s
True
>>>
```

Vnořené seznamy

Vnořené seznamy

- Prvkem seznamu může být cokoliv, tedy i seznam.
- Vytvoříme tak seznam seznamů:

```
>>> ss=[[1,2,3],[4,5,6],[7,8,9]]
>>> ss[1]
[4, 5, 6]
>>> ss[1][2]
6
```

Příklad

- Vytvoříme seznam, pro uložení pozice na šachovnici:

```
sachovnice=[[0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,
0],[0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0],[0,0,0,0,0,0,0,0]]

for radek in sachovnice:
    for sloupec in radek:
        print(sloupec,end=" ")
    print()
```

- Vytvoření šachovnice lze udělat v cyklu:

```
sachovnice=[]
for radek in range(8):
    sachovnice+=[[]]
    for sloupec in range(8):
        sachovnice[radek]+=[0]
```