

IB111 - skupina 10 - cvičení 06 - 09 - shrnutí

- příkaz input, do proměnné uloží vstup zadaný uživatelem a potvrzený klávesou ENTER

```
promenna = input("Zadej vstup")
```

- práce se seznamy (indexování apod.)

```
# vytvoreni seznamu
seznam = [1, 3, 2]

# pridani prvku na konec seznamu
seznam.append(10)
# seznam: [1, 3, 2, 10]

# pridani cisla 10 do seznamu na index 1
seznam.insert(1, 10)
# seznam: [1, 10, 3, 2, 10]

# odebrani prvnioho vyskytu cisla 10 ze seznamu
seznam.remove(10)
# seznam: [1, 3, 2, 10]

# odebrani prvku na indexu 1
del seznam[1]
# seznam: [1, 2, 10]

# otoceni seznamu
seznam.reverse()
# seznam: [10, 2, 1]
```

- procházení seznamů, hledání největšího prvku, průměrné hodnoty

- řadící algoritmy - bubble sort, insert sort, quick sort (v Pythonu i předdefinovaná funkce)

```
# serazeni seznamu
seznam.sort()
# seznam: [1, 2, 10]
```

- rekurze = funkce volá sebe samu, akorát se zmenšeným parametrem, pro nejmenší parametr (často 0 nebo 1) už vrátí přímo hodnotu = tj. program se vždy zastaví

- faktoriál rekurzivně:

```
def fact(n):
    if n == 1:
        return 1
    else return n * fact(n-1)
```

- rekurzivní Fibonacci, NSD, ochutnávka Hanojské věže, fraktály

- datové struktury: Zásobník

```
# vytvoreni zasobniku
zasobnik = [1, 2, 3]

# pridani do zasobniku
zasobnik.append(4)
# zasobnik: [1, 2, 3, 4]

# odebrani ze zasobniku
top = zasobnik.pop()
# zasobnik: [1, 2, 3], top: 4

# dalsi odebrani ze zasobniku
top = zasobnik.pop()
# zasobnik: [1, 2], top: 3
```

- datové struktury: Fronta

```
from collections import deque

# 3 studenti cekaji ve fronte na obed
fronta = deque(["Petr", "Zdenek", "Filip"])

# Kuba prisel na konec fronty
fronta.append("Kuba")
# fronta: deque(["Petr", "Zdenek", "Filip", "Kuba"])

# Roman prisel na konec fronty
fronta.append("Roman")
# fronta: deque(["Petr", "Zdenek", "Filip", "Kuba", "Roman"])

# prvni student ve fronte (Petr) dostal obed
student = fronta.popleft()
# fronta: deque(["Zdenek", "Filip", "Kuba", "Roman"]), student: "Petr"

# dalsi student ve fronte (Zdenek) dostal obed
student = fronta.popleft()
# fronta: deque(["Filip", "Kuba", "Roman"]), student: "Zdenek"
```

- datové struktury: Množina

```
# vytvoreni prazdne mnoziny
mnozina = set()

# pridani prvku do mnoziny
mnozina.add(5)
# mnozina: {5}

# pridani stejneho prvku do mnoziny vickrat
mnozina.add(6)
mnozina.add(6)
# mnozina: {5, 6}

# pridani kolekce prvku do mnoziny
```

```
mnozina.update([1, 3, 10, 15])
# mnozina: {1, 3, 5, 6, 10, 15}

# odebrani prvku z mnoziny
mnozina.remove(5)
# mnozina: {1, 3, 6, 10, 15}

# zjisteni velikosti mnoziny
pocet_prvku = len(mnozina)
# pocet_prvku: 5

# test vyskytu
if 10 in mnozina:
    print '10 je prvkem mnoziny'

# iterovani pres prvky mnoziny
for x in mnozina:
    print x, x * x
```

- Ukázka množinových operací:

```
# vytvoreni 2 mnozin
>>> a = {1, 3, 5, 7}
>>> b = {2, 3, 4, 5}

# sjednoceni mnozin
>>> a | b
set([1, 2, 3, 4, 5, 7])

# prunik mnozin
>>> a & b
set([3, 5])

# mnozinovy rozdíl
>>> a - b
set([1, 7])
```