Kopas

Viesturs Vēzis

Problēmjautājums

- Imitēt metamā kauliņa mešanu
- Imitēt numurētu lodīšu izvilkšanu

Metamā kauliņa mešana

```
import random

for i in range(6):
    print (random.randint(1,6))
```

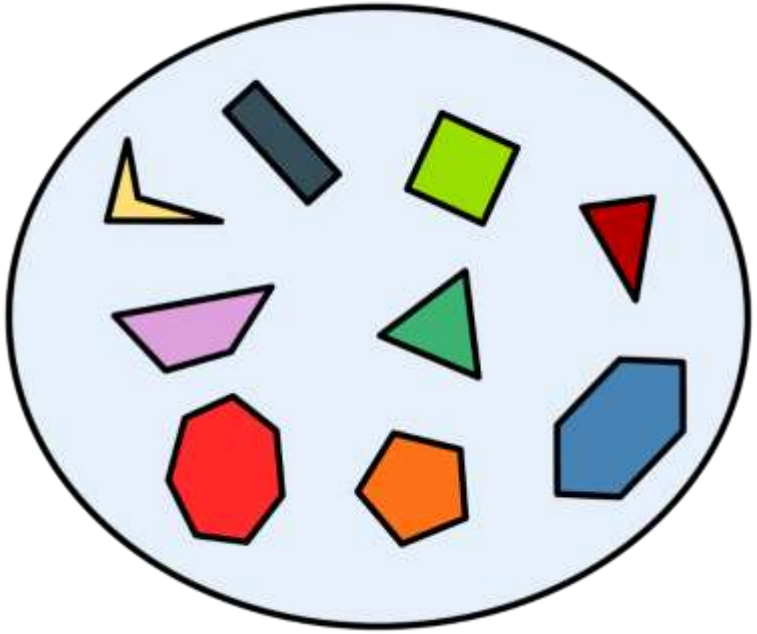
Numurētu lodīšu izvilkšana

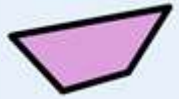
```
import random

a = {}
for i in range(6):
    b = random.randint(1,6)
    while b in a:
        b = random.randint(1,6)
    a.add(b)
    print(b)
```

Kopas

Viesturs Vēzis

Kopa un masīvs



0	1	2	3	4	5	6	7	8
								

Kopas izveide

```
a = {1, 2, 3, 4, 5}

b = {}
c = set()              # set()
d = {6}
# e = set(6)           # Kļūda
f = set((6,))
g = set([6])
h = set((1, 2, 3))
i = set([4, 5, 6])

j = set([1, 2, 2, 3])      # rezultāts {1, 2, 3}
```

NB! *Python* kopās visi elementi ir unikāli!

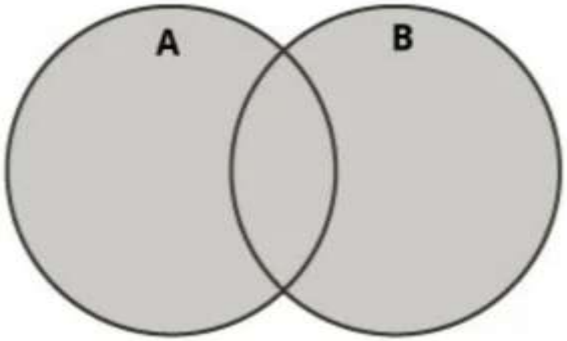
Nemaināmas kopas izveide

```
a = {1, 2, 3, 4, 5}
b = set((1, 2, 3))

c = frozenset ({1, 2, 3, 4, 5})
d = frozenset((1, 2, 3))
```

Kopas satura apskate

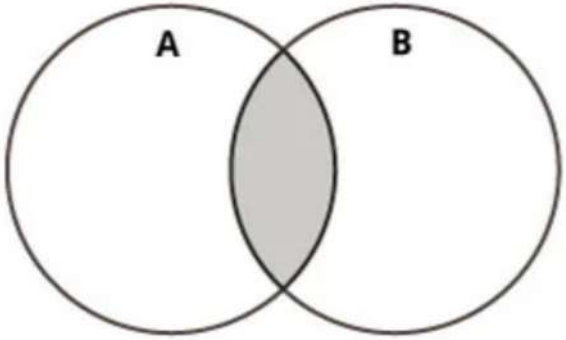
```
a = {1, 2, 3, 4, 5}

for x in a:
    print(x)

y = len(a) # elementu (vienumu) skaits kopā
```

Kopu apvienojums



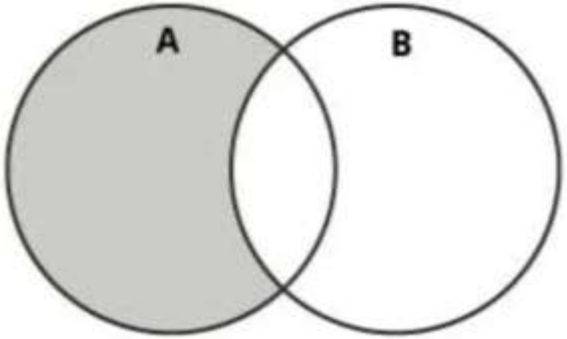

```
a = {1, 2, 3, 4, 5}
b = {3, 4, 5, 6, 7}

x = a | b
print(x)

x = a.union(b)
print(x)
print(a)
print(b)
```

Kopu šķēlums



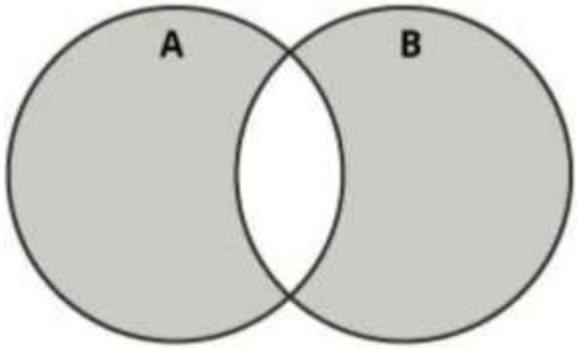

```
a = {1, 2, 3, 4, 5}
b = {3, 4, 5, 6, 7}

x = a & b
print(x)

x = a.intersection(b)
print(x)
print(a)
print(b)
```

Kopu starpība




```
a = {1, 2, 3, 4, 5}
b = {3, 4, 5, 6, 7}

x = a – b
print(x)

x = a.difference(b)
print(x)
print(a)
print(b)
```

Kopu «simetriskā starpība»




```
a = {1, 2, 3, 4, 5}
b = {3, 4, 5, 6, 7}

x = a ^ b
print(x)

x = a.symmetric_difference(b)
print(x)
print(a)
print(b)
```

Stingra apakškopa

```
a = {1, 2, 3, 4, 5}
b = {3, 4, 5, 6, 7}
c = {1, 2, 3, 4, 5}
d = {2, 3, 4}

x = b < a
y = a > b

x = c < a
y = a > c

x = d < a
y = d > a
```

Apakškopa

```
a = {1, 2, 3, 4, 5}
b = {1, 2, 3, 4, 5}
c = {2, 3, 4}

x = a <= b
y = a.issubset(b)

x = a >= b
y = a.issuperset(b)

x = a <= c
y = c <= a
z = a >= c
```

Vienādas kopas

```
a = {1, 2, 3, 4, 5}
b = {3, 4, 5, 6, 7}
c = {1, 2, 3, 4, 5}

x = a == b
print(x)
x = a == c
print(x)

x = a != b
print(x)
x = a != c
print(x)
```

Kopas nepārklājas

```
a = {1, 2, 3, 4, 5}
b = {1, 2, 3, 4, 5}
c = {3, 4, 5, 6, 7}
d = {2, 3, 4}
e = {6, 7, 8}

print(a.isdisjoint(b))
print(a.isdisjoint(c))
print(a.isdisjoint(d))
print(a.isdisjoint(e))
```

Elementa piederība kopai

```
a = {1, 2, 3, 4, 5}

x = 3 in a
print(x)

x = 3 not in a
print(x)
```

Elementa pievienošana kopai

```
a = {1, 2, 3, 4, 5}

x = 8
y = {x}
z = a | y
print(z)

a.add(9) # neko neatgreiž
print(a)

a.update({4, 5, 6, 7})
print(a)
```

Elementa izņemšana no kopas I

```
a = {1, 2, 3, 4, 5}

a.remove(3)  # neko neatgreiž
print(a)

a.remove(7)  # kļūda
print(a)

a.discard(3)  # neko neatgreiž
print(a)

a.discard(7)  # neko neizņem
print(a)
```

Elementa izņemšana no kopas II

```
a = {1, 2, 3, 4, 5}

x=a.pop()  # izņem vienu elementu no netukšas kopas
print(a)
print(x)

a.clear()  # kopas visa satura dzēšana
```

Unikālie simboli simbolu virknē

```
# x = { }
# y = { }
x = set()
y = set()
sv = "KOKOSIUNBANANI"
n = len(sv)
for i in range(n):
    s = sv[i]
    if s in x:
        y.add(s)
    else:
        x.add(s)
z = x - y
print("Unikālie simboli ir: ", z)
print("Atkārtojas: ", y)
```

```
x = set()
y = set()
sv = "KOKOSIUNBANANI"
n = len(sv)
for i in range(n):
    s = sv[i]
    if s in x:
        y = y | {s}
    else:
        x = x | {s}
z = x - y
print("Unikālie simboli ir: ", z)
print("Atkārtojas: ", y)
```

```
x = set()
y = set()
sv = "KOKOSIUNBANANI"
n = len(sv)
for i in range(n):
    s = sv[i]
    if s in x:
        y.update({s})
    else:
        x.update({s})
z = x - y
print("Unikālie simboli ir: ", z)
print("Atkārtojas: ", y)
```

Lotereja

```
import random

a = set()
for i in range(5):
    b = random.randint(1,35)
    while b in a:
        b = random.randint(1,35)
    a.add(b)
```

```
x = set()
for i in range(1,6):
    y = int(input("ievadi "
            + str(i)
            + ".skaitli ===>"))
    while y in x:
        y = int(input("ievadi "
                + str(i)
                + ".skaitli ===>"))
    x.add(y)

z = a & x
sk = len(z)

print("Izlozētie skaitļi: ", a)
print("Jūsu skaitļi: ", x)
print("Jūs atminējāt ", sk, " skaitļus un tie ir ", z)
```

Deju pāri

```
import random

zk = set()
mk = set()

for i in range(10):

    z = chr(random.randint(0,25) + 65)
    while z in zk:
        z = chr(random.randint(0, 25) + 65)
    zk.add(z)

    m = chr(random.randint(0, 25) + 97)
    while m in mk:
        m = chr(random.randint(0, 25) + 97)
    mk.add(m)

    print(z, m)
```

Kopu izveide ar citām datu struktūrām

Metodes darbam ar sarakstiem I

```
saraksts.append(x)   - pievieno elementu x sarakstam
                       saraksts
saraksts.clear()     - izņem visus elementus no
                       saraksta saraksts
                       (saraksts kļūst tukšs)
saraksts.copy()      - atgriež saraksta saraksts
                       kopiju
saraksts.count(x)    - saskaita, cik elementu x ir
                       sarakstā saraksts
saraksts.extend(s1) - pievieno visus saraksta s1
                       elementus sarakstam saraksts
saraksts.index(x)    - atgriež vietu (indeksu),
                       kur vērtība x pirmo reizi
                       ir sarakstā saraksts
```

Metodes darbam ar sarakstiem II

saraksts.**insert**(i, x) - ievieto elementu x saraksta saraksts vietā (indeksā) i

saraksts.**pop**() - izņem un atgriež saraksta pēdējo elementu saraksts

saraksts.**remove**(x) - izņem pirmo elementu ar vērtību x, kas atrodams sarakstā saraksts

saraksts.**reverse**() - apgriež elementus pretējā secībā sarakstā saraksts

saraksts.**sort**() - sakārto saraksta saraksts elementus augošā secībā

Kopu izveide ar sarakstiem

```
def jauns():
    return list()
```

```
def vaiPieder(elements, kopa):
    # return elements in kopa
    for x in kopa:
        if x == elements:
            return True
    return False
```

```
def pievienot(elements, kopa):
    paz = True
    for x in kopa:
        if x == elements:
            paz = false
    if paz:
        kopa.append(elements)
```

```
def izmest(elements, kopa):
    for x in kopa:
        if x == elements:
            kopa.remove(elements)
```

```
def apvienojums(kopa1, kopa2):
    kopa3 = list()
    for x in kopa1:
        kopa3.append(x)
    for x in kopa2:
        if x not in kopa3:
            kopa3.append(x)
    return kopa3
```

```
def skelums(kopa1, kopa2):
    kopa3 = list()
    for x in kopa1:
        if x in kopa2:
            kopa3.append(x)
    return kopa3
```

```
def starpiba(kopa1, kopa2):
    kopa3 = list()
    for x in kopa1:
        if x not in kopa2:
            kopa3.append(x)
    return kopa3
```

```
def vaiApakskopa(kopa1, kopa2):
    for x in kopa1:
        if x not in kopa2:
            return False
    return True
```

```
def vaiVienadas(kopa1, kopa2):
    for x in kopa1:
        if x not in kopa2:
            return False
    for x in kopa2:
        if x not in kopa1:
            return False
    return True
```

```
def saturs(kopa):
    kopa.sort()
    sv = ""
    for x in kopa:
        sv = sv + str(x) + " "
    if sv == "":
        sv = "Tukša kopa"
    return sv
```

```
def kopasIzveide():
    kopa = list()
    n = int(input("Ievadi kopas apjomu ===>"))
    for i in range(1, n+1):
        x = input("Ievadi kopas "
                  + str(i)
                  + ". elementu ===>")
        if x not in kopa:
            kopa.append(x)
    return kopa
```

Unikālie simboli simbolu virknē

```
x = jauns()
y = jauns()
sv = "KOKOSIUNBANANI"
n = len(sv)
for i in range(n):
    s = sv[i]
    if vaiPieder(s, x):
        pievienot(s, y)
    else:
        pievienot(s, x)
z = starpiba(x, y)
print("Unikālie simboli ir: ", saturs(z))
print("Atkārtojas: ", saturs(y))
```

Lotereja

```
a = jauns()
for i in range(5):
    b = random.randint(1,35)
    while b in a:
        b = random.randint(1,35)
    pievienot(b, a)
```

```
x = jauns()
for i in range(1,6):
    y = int(input("ievadi "
            + str(i)
            + ".skaitli ===>"))
    while y in x:
        y = int(input("ievadi "
                + str(i)
                + ".skaitli ===>"))
    pievienot(y, x)

z = skelums(a, x)
sk = len(z)
print("Izlozētie skaitļi: ", saturs(a))
print("Jūsu skaitļi: ", saturs(x))
print("Jūs atminējāt ", sk, " skaitļus un tie ir ",
      saturs(z))
```

Deju pāri

```
import random

zk = jauns()
mk = jauns()

for i in range(10):

    z = chr(random.randint(0,25) + 65)
    while z in zk:
        z = chr(random.randint(0, 25) + 65)
    pievienot(z, zk)

    m = chr(random.randint(0, 25) + 97)
    while m in mk:
        m = chr(random.randint(0, 25) + 97)
    pievienot(m, mk)

    print(z, m)
```