Programmeerimine

6. loeng

Täna loengus

- Loogilised avaldised
- Hargnemisdirektiivid:
  - Üldkujuline if-lause
  - Tingimusavaldis

Loogilised avaldised

Loogilised avaldised

- Avaldisi, mille väärtus on tõeväärtustüüpi, nimetatakse loogilisteks avaldisteks.
- Loogilised avaldised koosnevad muutujatest, tõeväärtuskonstantidest, relatsioonilistest- ja loogilistest operaatoritest.
- Tõeväärtustüüp Pythonis:

Tüübi nimi	*bool*
Tõene väärtus	`True`
Väär väärtus	`False`

- NB! Pythonis on tõeväärtustüüp täisarvutüübi alamtüüp, kus `False` on sama mis `0` ja `True` on sama mis `1`.

Relatsioonilised operaatorid

- NB! Ujukomaarvude võrdlemisel tuleb arvestada, et tegemist ei ole reaalarvudega ja kõik tehted ujukomaarvudel on ligikaudsed.
- Näiteks:

```
0.1+0.1+0.1-0.3        ⟹  5.551115123125783e-17
0.1+0.1+0.1-0.3 == 0   ⟹  False
```

- Ujukomaarvude võrdsuse või mittevõrdsuse kontrolliks tuleks enamik juhtudel operaatorite `==` ja `!=` asemel kasutada arvude vahe nulliga läheduse kontrolli; so.

$$abs(x - y) <= eps$$

kus *eps* on mingi nulli lähedane arv ja näitab millise täsuseni soovime ujukomaarvudega opereerida.

Relatsioonilised operaatorid

- Sõnede võrdlemine toimub leksikograafilises järjekorras.
- Tähemärkide võrdlemisel võrreldakse märkide ASCII või Unicode väärtusi.
- Numbreid tähistavad märgid on loomulikus järjekorras.
  - Märk `'0'` on koodiga `48`.
  - Märk `'1'` on koodiga `49`, jne.
- Ladina tähestiku suurtähed `'A'`, `'B'`, ..., `'Z'`.
  - Täht `'A'` on koodiga `65`, täht `'B'` koodiga `66`, jne.
- Ladina tähestiku väiketähed `'a'`, `'b'`, ..., `'z'`.
  - Täht `'a'` on koodiga `97`, täht `'b'` koodiga `98`, jne.
- NB! Suurtähed on väiketähtedest väiksema koodiga.

Loogilised operaatorid

- Keerukamate loogiliste avaldiste moodustamine toimub loogiliste operaatorite abil.

Operaator	Kirjeldus
`not`	loogiline eitus (unaarne)
`and`	loogiline JA
`or`	loogiline VÕI

- Loogiliste operaatorite formaalseks spetsifitseerimiseks kasutatakse tõeväärtustabeleid.
- Loogiline eitus:

x	`not` x
`True`	`False`
`False`	`True`

Loogilised operaatorid

- Loogiline JA (konjunktsioon):

x	y	x and y
True	True	True
True	False	False
False	True	False
False	False	False

- Loogiline VÕI (disjunktsioon):

x	y	x or y
True	True	True
True	False	True
False	True	True
False	False	False

Operaatorite prioriteedid

Astendamine	`**`
Unaarsed operaatorid	`-, +`
Multiplikatiivsed operaatorid	`*, /, //, %`
Aditiivsed operaatorid	`+, -`
Võrdlusoperaatorid	`<, <=, >, >=`
Võrdusoperaatorid	`==, !=`
Loogiline eitus	`not`
Loogiline JA	`and`
Loogiline VÕI	`or`

Loogiliste avaldiste väärtustamine

- Sarnaselt aritmeetiliste avaldistega, väärtustatakse ka loogilistes avaldistes üldjuhul operaatori argumendid enne operaatori väärtustamist.
- Erandiks on operaatorid `and` ja `or`, millede korral kasutatakse nn. kärbetega väärtustamist (*ingl. shortcut evaluation*).
  - Kõigepealt väärtustatakse esimene argument ja kui selle põhjal on võimalik leida kogu väärtus, siis teist argumenti ei väärtustata.
  - Operaatori `and` korral, kui esimene argument on `False`, siis tulemuseks on `False`.
  - Operaatori `or` korral, kui esimene argument on `True`, siis tulemuseks on `True`.

Loogiliste avaldiste lihtsustamine

- De Morgani seadused:

  $$\textbf{not}\ (x\ \textbf{and}\ y) == \textbf{not}\ x\ \textbf{or}\ \textbf{not}\ y$$

  $$\textbf{not}\ (x\ \textbf{or}\ y) == \textbf{not}\ x\ \textbf{and}\ \textbf{not}\ y$$

Loogiliste avaldiste lihtsustamine

- De Morgani seadused:

  $$\textbf{not}\ (x\ \textbf{and}\ y) == \textbf{not}\ x\ \textbf{or}\ \textbf{not}\ y$$

  $$\textbf{not}\ (x\ \textbf{or}\ y) == \textbf{not}\ x\ \textbf{and}\ \textbf{not}\ y$$

- Näide:

  $$\textbf{not}\ (x > 0\ \textbf{and}\ (y <= 7\ \textbf{or}\ z\ != 3))$$

  $$\Rightarrow \textbf{not}\ (x > 0)\ \textbf{or}\ (\textbf{not}(y <= 7\ \textbf{or}\ z\ != 3))$$

  $$\Rightarrow x <= 0\ \textbf{or}\ (\textbf{not}(y <= 7)\ \textbf{and}\ \textbf{not}(\ z\ != 3))$$

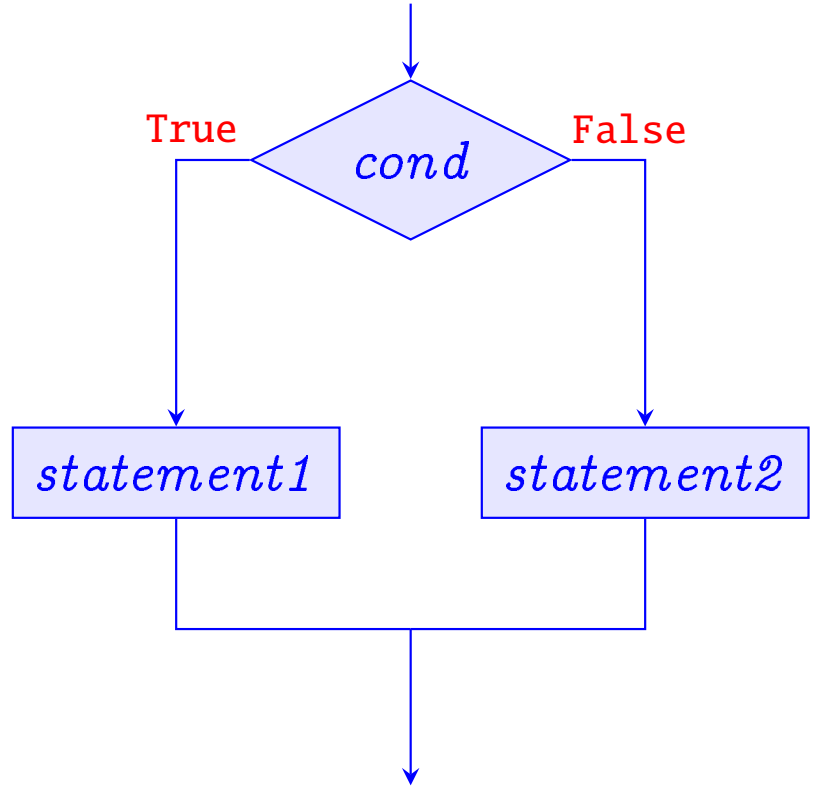
  $$\Rightarrow x <= 0\ \textbf{or}\ (y > 7\ \textbf{and}\ z == 3)$$

Tingimuslaused

- if-else-lause:

```
if cond:
    statement1
else:
    statement2
```

- Kui tingimus *cond* on tõene, siis täidetakse lause *statement1*.
- Kui tingimus *cond* on väär, siis täidetakse lause *statement2*.



Tingimuslaused

- Kolme või enama variandiga hargnemiseks saab kasutada "if-else-if-redelit":

```
if cond1:
    statement1
elif cond2:
    statement2
  ⋮
elif condN:
    statementN
else:
    statementN+1
```

- Võrreldes üksteisesse sisestatud if-else-lausetega, võivad kõik harud olla sama taandega.

Tingimuslaused

Näide – temperatuur Fahrenheiti skaalas

```
t = input("Sisesta temperatuur: ")
f = int(t)* 9/ 5 + 32
if f > 212:
    print (f, "Aur")
elif f > 112:
    print(f, "Väga kuum vesi")
elif f > 32:
    print (f, "Vesi")
else:
    print (f, "Jää")
```

Tingimusavaldised

- Lisaks tingimuslausetele on Pythonis olemas ka tingimusavaldised:
  exp1 **if** *cond* **else** *exp2*
- Kui tingimus *cond* on tõene, on tingimusavaldise väärtuseks avaldise *exp1* väärtus.
- Vastasel korral on kogu tingimusavaldise väärtuseks avaldise *exp2* väärtus.
- NB! Erinevalt tingimuslausest on tingimusavaldisel väärtus; muuhulgas võib tingimusavaldis esineda näiteks omistuslause paremas pooles.

Tingimusavaldised

Näide – kahest arvust maksimaalse leidmine

```
print('Sisestage kaks arvu:')
arv1 = int(input('Esimene: '))
arv2 = int(input('Teine: '))

max = arv2 if arv1 < arv2 else arv1

print('Maksimaalne on arv', max)
```

Tingimuslaused

- Keeruliste tingimusavaldistega if-else- lauseid saab teisendada üksteisesse sisestatud lihtsamate tingimustega if-else- lauseteks.
- Eituse elimineerimine:

```
if not cond:
    stmt1
else:
    stmt2
```

$\Longleftrightarrow$

```
if cond:
    stmt2:
else:
    stmt1
```

Tingimuslaused

- Keeruliste tingimusavaldistega if-else- lauseid saab teisendada üksteisesse sisestatud lihtsamate tingimustega if-else- lauseteks.
- Loogilise JA elimineerimine

```
if cond1 and cond2:
    stmt1
else:
    stmt2
```

$\Longleftrightarrow$

```
if cond1:
    if cond2:
        stmt1
    else:
        stmt2
else:
    stmt2
```

Tingimuslaused

- Keeruliste tingimusavaldistega if-else- lauseid saab teisendada üksteisesse sisestatud lihtsamate tingimustega if-else- lauseteks.
- Loogilise VÕI elimineerimine:

```
if cond1 or cond2:
    stmt1
else:
    stmt2
```

$\Longleftrightarrow$

```
if cond1:
    stmt1
elif cond2:
    stmt1
else:
    stmt2
```

Tingimuslaused - näide

```
if not (a > 3 or b != 2 and a == 0) :
    print(a)
else:
    print(b)
```

Tingimuslaused - näide

```
if not (a > 3 or b != 2 and a == 0) :
    print(a)
else:
    print(b)
```

```
if a >3 or b != 2 and a == 0 :
    print(b)
else:
    print(a)
```

Tingimuslaused - näide

```
if not (a > 3 or b != 2 and a == 0) :
    print(a)
else:
    print(b)
```

```
if a >3 or b != 2 and a == 0 :
    print(b)
else:
    print(a)
```

```
if a > 3:
    print(b)
elif b != 2 and a == 0:
    print(b)
else:
    print(a)
```

Tingimuslaused - näide jätkub

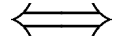
```
if a > 3:
    print(b)
elif b != 2 and a == 0:
    print(b)
else:
    print(a)
```

$\Longleftrightarrow$

```
if a > 3:
    print(b)
elif b != 2:
    if a == 0:
        print(b)
    else:
        print(a)
else:
    print(a)
```

Tingimuslaused - näide 2

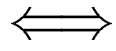
```
if x == 0:
    if y < 0:
        print(x)
    else:
        print(y)
elif x == 1:
    if y >= 0:
        print(y)
    else:
        print(x)
else:
    print(x)
```



```
if x == 0:
    if not (y < 0):
        print(y)
    else:
        print(x)
elif x == 1:
    if y >= 0:
        print(y)
    else:
        print(x)
else:
    print(x)
```

Tingimuslaused - näide 2

```
if x == 0:
    if y < 0:
        print(x)
    else:
        print(y)
elif x == 1:
    if y >= 0:
        print(y)
    else:
        print(x)
else:
    print(x)
```



```
if x == 0:
    if y >= 0:
        print(y)
    else:
        print(x)
elif x == 1:
    if y >= 0:
        print(y)
    else:
        print(x)
else:
    print(x)
```

Tingimuslaused - näide 2

```
if x == 0:
    if y >= 0:
        print(y)
    else:
        print(x)
elif x == 1:
    if y >= 0:
        print(y)
    else:
        print(x)
else:
    print(x)
```

Tingimuslaused - näide 2

```
if x == 0:
    if y >= 0:
        print(y)
    else:
        print(x)
elif x == 1:
    if y >= 0:
        print(y)
    else:
        print(x)
else:
    print(x)
```

$\Longleftrightarrow$

```
if x == 0 or x == 1:
    if y >= 0:
        print(y)
    else:
        print(x)
else:
    print(x)
```

Tingimuslaused - näide 2

```
if x == 0 or x == 1:
    if y >= 0:
        print(y)
    else:
        print(x)
else:
    print(x)
```

$\Longleftrightarrow$

```
if (x == 0 or x == 1) and y >= 0:
    print(y)
else:
    print(x)
```

Funktsioonid, mis tagastavad tõeväärtuse

Näide – kas kolmnurk on täisnurkne

Eeldame, et a, b, c moodustavad kolmnurga

```
#eps – väike positiivne arv
def kasTaisnurkneKolmnurk(a, b, c, eps):
    if abs(a**2-b**2-c**2)<eps or abs(b**2-a**2-c**2)<eps or\
       abs(c**2 - a**2 - b**2) < eps:
        return True
    else:
        return False
```

```
a = 3; b = 4; c = 5
if kasTaisnurkneKolmnurk(a, b, c):
    print('Kolmnurk külgedega', a, b, c, 'on täisnurkne')
else:
    print('Kolmnurk külgedega', a, b, c, 'ei ole täisnurkne')
```

Funktsioonid, mis tagastavad tõeväärtuse

Näide – kas arv on algarv

```
from math import sqrt
def kasAlgarv(arv):
    s = int(sqrt(arv))
    while s > 1:
        if arv % s == 0:
            return False
            break
        else:
            s -= 1
    return True
```

```
a = int(input('Sisesta naturaalarv: '))
if kasAlgarv(a):
    print('Arv', a, 'on algarv')
else:
    print('Arv', a, 'ei ole algarv')
```

Järgmiseks korraks

- Lugeda läbi õpiku peatükid:
  - Ptk. 6 *"bool ja tingimuslaused"*
  - Ptk. 7 *"Järjendid ja for-tsükkel"*

Suur tänu osalemast

ja

kohtumiseni!