

Programmeerimine

5.loeng

Täna loengus

- Funktsioonid ja protseduurid

Protseduurid

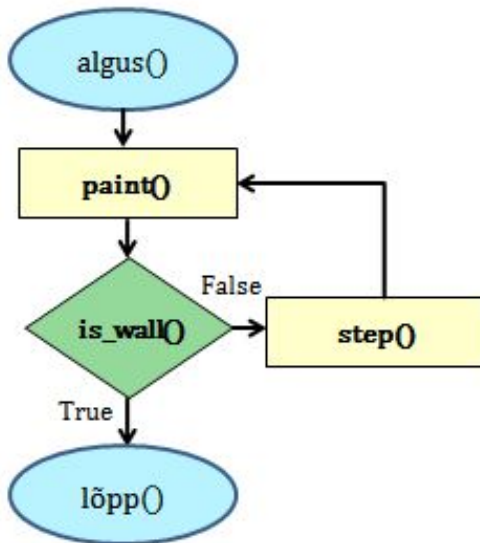
Protseduurid

- **Protseduur** on suhteliselt iseseisev korduvkasutatav programimiosa, mis koosneb käskudest ja mida on võimalik välja kutsuda tema nime kaudu.
- Protseduur võib olla parametrizeeritud argumentidega, mille konkreetsed väärtused määratakse protseduuri väljakutsel.
- Protseduuride defineerimine Pythonis:

```
def procName(arg1, ..., argn):  
    procBody
```

Pykkar värvib triibu

värvi_triip()

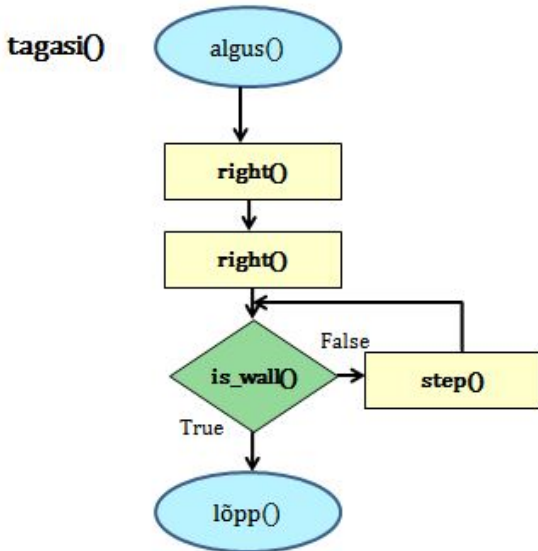


Protseduurid

Algeis: Pykkar on põhjaserval näoga lõunasse. Pykkar värvib triibu

```
from pykkar import *  
def triip ():  
    while True:  
        paint()  
        if is_wall ():  
            break  
    else:  
        step()
```

Pykkar liigub tagasi

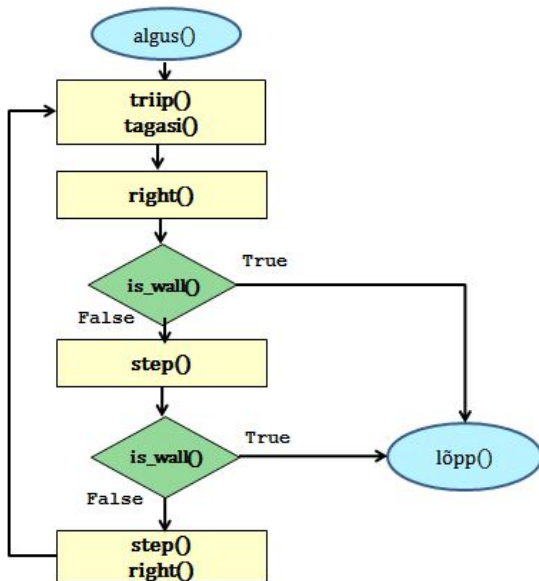


Protseduurid

Algseis: Pykkar on lõunaserval näoga lõunasse. Pykkar pöörab ümber ja liigub põhjaservani.

```
from pykkar import *  
def tagasi():  
    right() # ümberpöörd  
    right()  
    while True:  
        if is_wall():  
            break  
        else:  
            step()
```


Põhiprogramm



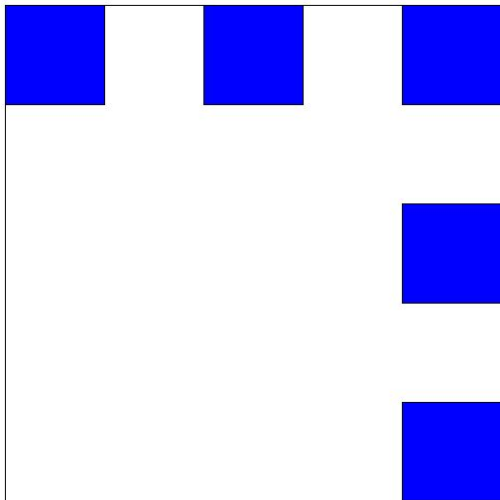
Põhiprogramm

Algseis: Pykkar on loodenurgas näoga lõunasse. Pykkar värvib ruudustiku triibuliseks.

```
from pykkar import *
while True:
    triip ()
    tagasi ()
    right ()
    if is_wall (): # ei lase robotil vastu seina põrgata
        break
    else:
        step() # robot liigub ühe ruudu võrra edasi
        if is_wall ():
            break
        else:
            step ()
            right ()
```

Kontrolltöö ülesanne

Joonistada kilpkonna vahendusel 5×5 ruudu kahe külje ruudud üle ühe alustades nurgast. Bonus joonistatud küljeruutude värvimise eest.



Funktsioonid

Funktsioonid

- **Funktsioon** on protseduur, mis väljastab väärtuse.
- Funktsiooni rakendus on avaldis, mille väärtust võib kasutada sarnaselt teiste (aritmeetilised-, loogilised-, jne) avaldistega.
- Ilma väärtuseta protseduuri saab kasutada lausena ning tema "toimeks" on kõrvalefekt programmi olekule.
- Objektorienteeritud keeltes (näit. Java) on protseduurid seotud mingi klassiga, misjuhul neid nimetatakse **meetodideks**.

Funktsioonid

- Funktsioonide defineerimine Pythonis:

```
def funcName(arg1, ..., argn):  
    funcBody
```

- *funcName* on funktsiooni nimi.
- *arg_i* on funktsiooni formaalsed parameetrid.
- *funcBody* on funktsiooni keha, mis koosneb lausetest.
- Funktsiooni väljakutsel antakse formaalsetele parameetritele tegelike argumentide väärtused.
- Formaalsed parameetrid ja funktsiooni kehas defineeritud muutujad on funktsioonis lokaalsed; so. on nähtavad ainult selle funktsiooni kehas.
- **return** lause lõpetab funktsiooni töö ja kui tal on argument, siis selle väärtus on funktsiooni väljakutse väärtuseks.

Funktsioonid

Näide – keskmise arvutamine

```
def keskmine(arv1, arv2):  
    k = (arv1 + arv2) / 2  
    return k
```

Näide – temperatuuri teisendamise Fahrenheiti skaalasse

```
def celsius2Fahrenheit(tempC):  
    tempF = round(tempC*1.8+32.0)  
    return tempF
```

Funktsioonid

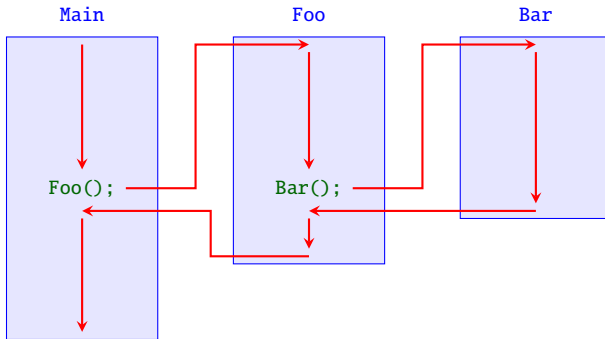
Näide – keskmise arvutamine

```
def keskmine(arv1, arv2):  
    k = (arv1 + arv2) / 2  
    return k
```

Näide – temperatuuri teisendamise Fahrenheiti skaalasse

```
def celsius2Fahrenheit(tempC):  
    tempF = round(tempC*1.8+32.0)  
    return tempF
```


Kontrollvoog funktsioonide väljakutsetel



Funktsioonid



```
arv = 7
```

```
arv = keskmine(arv + 1, 2)
```

```
arv = keskmine(arv, arv - 2)
```

```
def keskmine(arv1, arv2) :
```

```
    k = (arv1 + arv2)/2
```

```
    return k
```

Funktsioonid

 $arv = 7$
 $arv = keskmine(arv + 1, 2)$
 $arv = keskmine(arv, arv - 2)$

arv 7

```
def keskmine(arv1, arv2) :  
    k = (arv1 + arv2)/2  
    return k
```

Funktsioonid

arv = 7

arv = *keskmine*(*arv* + 1, 2)

arv = *keskmine*(*arv*, *arv* - 2)

arv

7



def *keskmine*(*arv1*, *arv2*) :

k = (*arv1* + *arv2*)/2

return *k*

arv1

8

arv2

2

Funktsioonid

arv = 7

arv = *keskmine*(*arv* + 1, 2)

arv = *keskmine*(*arv*, *arv* - 2)

arv

7

def *keskmine*(*arv1*, *arv2*) :

k = (*arv1* + *arv2*)/2

return *k*

arv1

8

arv2

2

k


5



Funktsioonid

arv = 7

arv = *keskmine*(*arv* + 1, 2)

 *arv* = *keskmine*(*arv*, *arv* - 2)

arv

5

def *keskmine*(*arv1*, *arv2*) :

k = (*arv1* + *arv2*)/2

return *k*

Funksioonid

arv = 7

arv = *keskmine*(*arv* + 1, 2)

arv = *keskmine*(*arv*, *arv* - 2)

arv

5



def *keskmine*(*arv1*, *arv2*) :

k = (*arv1* + *arv2*)/2

return *k*

arv1

5

arv2

3

Funktsioonid

arv = 7

arv = *keskmine*(*arv* + 1, 2)

arv = *keskmine*(*arv*, *arv* - 2)

arv

5

def *keskmine*(*arv1*, *arv2*) :

k = (*arv1* + *arv2*)/2

return *k*

arv1

5

arv2

3

k

4



Funktsioonid

arv = 7

arv = *keskmine*(*arv* + 1, 2)

arv = *keskmine*(*arv*, *arv* - 2)



arv

4

def *keskmine*(*arv1*, *arv2*) :

k = (*arv1* + *arv2*)/2

return *k*

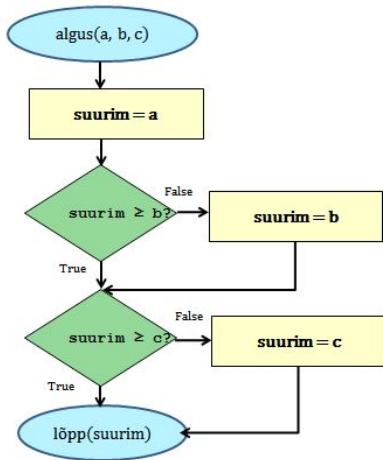
Funktsioonid

- Funktsiooni keha võib alata sõnega, mis annab funktsiooni lühikirjelduse (*ingl. docstring*).
 - Vastav sõne seotakse funktsiooniobjekti atribuudiga `--doc--`.

Näide – Fahrenheiti skaalast temperatuuri teisendamine

```
def fahrenheit2Celsius(t):  
    """Teisendab temperatuuri Fahrenheiti skaalast  
       Celsiuse skaalasse."""  
    return round(5*(t-32)/9)
```

Kolmest arvust suurim



Funktsioonid

Näide – kolmest arvust suurim

```
def kolmestSuurim(a, b, c):  
    suurim = a  
    if suurim < b:  
        suurim = b  
    if suurim < c:  
        suurim = c  
    return suurim
```

Järgmiseks korraks

- Lugada läbi õpiku peatükid:
 - Ptk. 5 "*Alamprogrammid ehk funktsioonid*"
 - Ptk. 6 "*Tingimuslaused*"

Suur tänu osalemast

ja

kohtumiseni!