

UPORABA IKT V NARAVOSLOVJU IN TEHNIKI

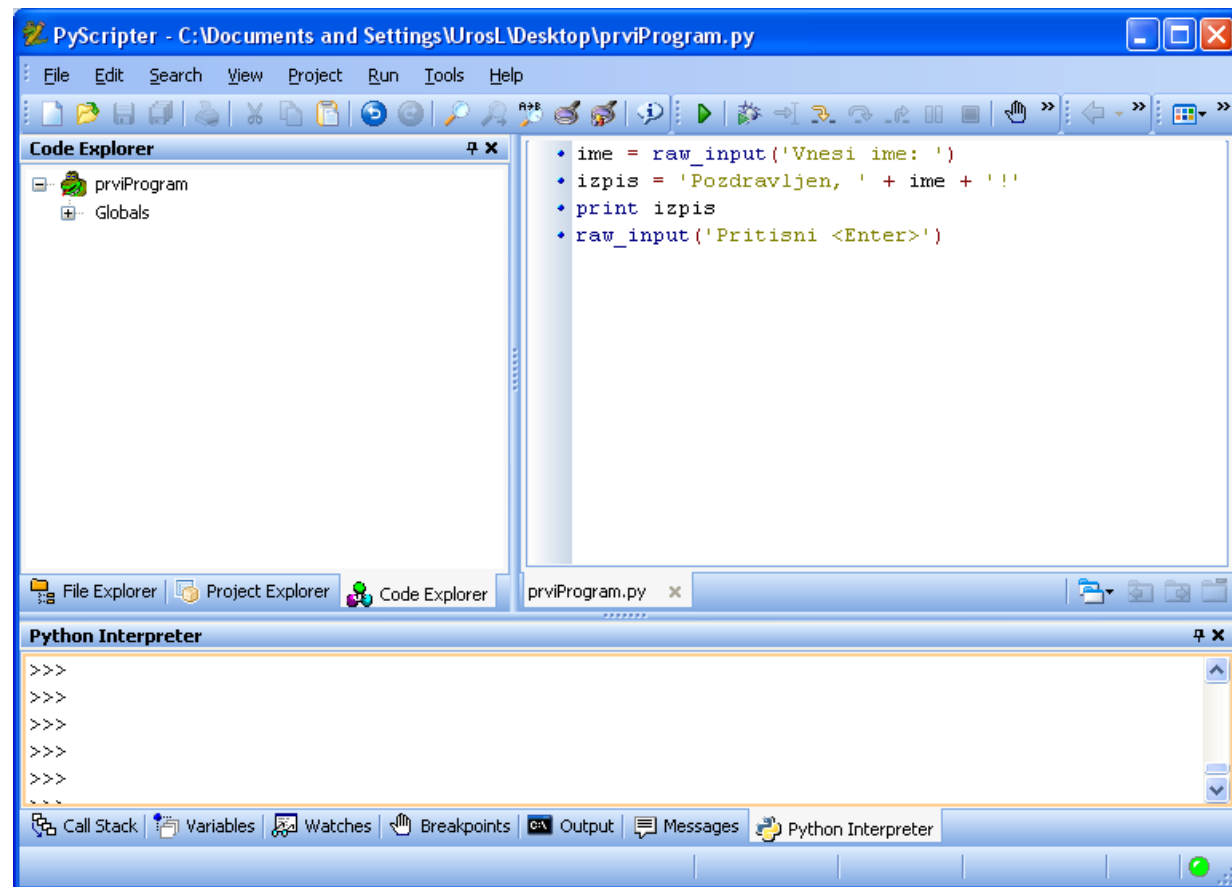
Predavanje 11

Python: zanke

doc.dr. Mira Trebar

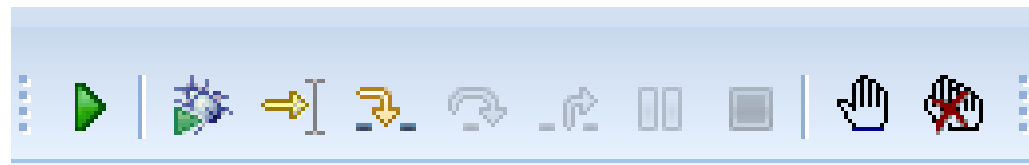
Prvi program

- Zadnjih šest vrstic ukazov prepíšimo v datoteko. Lahko uporabimo PyScripter ali pa kar beležnico.
- Datoteko shranimo kot prviProgram.py
- Poiščimo jo in jo dvakrat kliknimo.
- Bravo!



Razhroščevalnik

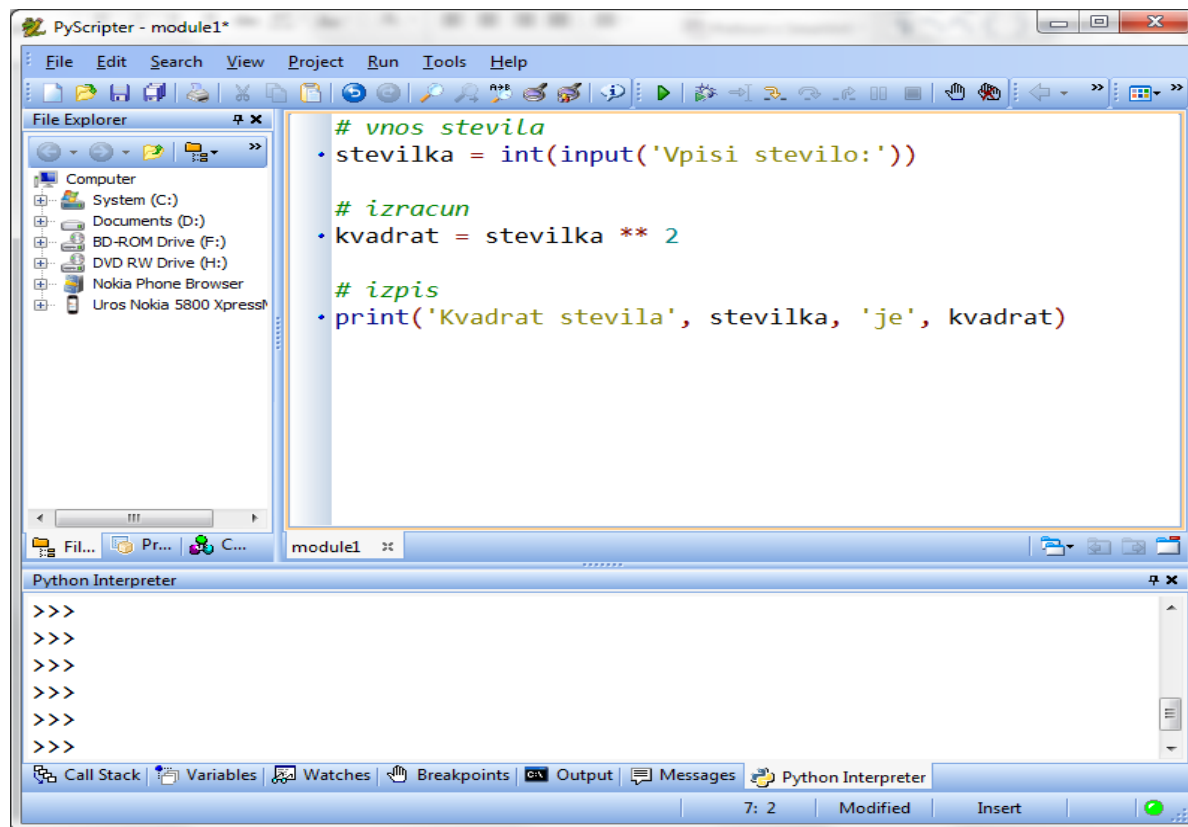
- Zlata vredno orodje pri programiranju
- Omogoča nam
 - izvajanje programa po korakih (F7)
 - izvajanje programa do utripača (F4)
 - izvajanje programa do ustavitvene točke
 - postavljanje in brisanje ustavitvenih točk (F5)



- Opazovanje vrednosti (stanja) spremenljivk

Razhroščevalnik

- z miško se postavimo na spremenljivko in odpremo opazovalno okno (ang. Watches)
 - vanj vpišemo spremenljivke, katerih vrednosti nas zanimajo
 - desni klik v oknu, Add Watch, spremenljivko vpišemo
 - postavimo se na spremenljivko v programu in stisnemo Alt+W



Pomembne malenkosti

- program s komentarjem

```
#  
# Program za racunanje kvadrata stevila  
#  
  
# tule bomo uporabnika prosili, da vpise stevilo  
stevilka = int(input('Vpisi stevilo:'))  
  
# izracunamo kvadrat stevila  
kvadrat = stevilka ** 2  
  
# stevilko in njen kvadrat izpisemo  
print('Kvadrat stevila', stevilka, 'je', kvadrat)
```

Program – ITM (1)

- Računanje indeksa telesne mase (ITM)

```
# program izracuna indeks telesne mase

# vnos
teza = float(input('Vnesite vazo tezo v kg:'))
visina = float(input('Vnesite vazo visino v m:'))

# izracun
ITM = teza / visina ** 2

# izpisi
print('Tehtate:', teza, 'kg')
print('Visoki ste', visina, 'm')
print('Vas ITM je', ITM, 'kg/m2')
```

Program – ITM (2)

- Računanje indeksa telesne mase

- Kaj pa, če vpišemo višino 0?

```
Traceback
```

```
      <module> <module1>      8  
exceptions.ZeroDivisionError: float division
```

- Kako to preprečiti?

- Program moramo napisati pametneje.
- Če je višina večja od nič, izračuna ITM, drugače pač ne.

- Kako to naredimo?

- Samo to, kar smo napisali zgoraj moramo napisati v matematično in po Pythonovsko obarvani angleščini.

ITM - pogojni stavki (1)

- Boljši program za računanje indeksa telesne mase

```
# program izracuna indeks telesne mase

# vnos
teza = float(input('Vnesite vazo tezo v kg:'))
visina = float(input('Vnesite vazo visino v m:'))

# preverimo, ce je racun mozen
if visina > 0:
    # izracun
    ITM = teza / visina ** 2

    # izpisi
    print('Tehtate:', teza, 'kg')
    print('Visoki ste', visina, 'm')
    print('Vas ITM je', ITM, 'kg/m2')
```

prg03b.py

ITM - pogojni stavki (2)

- Program bi lahko povedal, zakaj ni izračunal ITM.

```
# program izracuna indeks telesne mase

# vnos
teza = float(input('Vnesite vazo tezo v kg:'))
visina = float(input('Vnesite vazo visino v m:'))

# preverimo, ce je racun mozen
if visina > 0:
    # izracun
    ITM = teza / visina ** 2

    # izpisi
    print('Tehtate:', teza, 'kg')
    print('Visoki ste', visina, 'm')
    print('Vas ITM je', ITM, 'kg/m2')
else:
    print('Z vazo visino je nekaj hudo narobe')
```

prg03c.py

ITM - pogojni stavki (3)

- Na koncu programa dodajmo še pozdrav.
- zakaj nas računalnik ne pozdravi vsakič?

prg03d.py

```
# program izracuna indeks telesne mase

# vnos
teza = float(input('Vnesite vazo tezo v kg:'))
visina = float(input('Vnesite vazo visino v m:'))

# preverimo, ce je racun mozen
if visina > 0:
    # izracun
    ITM = float(teza) / visina ** 2

    # izpisi
    print('Tehtate:', teza, 'kg')
    print('Visoki ste', visina, 'm')
    print('Vas ITM je', ITM, 'kg/m2')
else:
    print('Z vazo visino je nekaj hudo narobe!')
    print('Se kdaj me uporabi. Nasvidenje!')
```

ITM - pogojni stavki (4)

- Pozdrav mora biti pravilno zamaknjen.

```
# program izracuna indeks telesne mase

# vnos
teza = float(input('Vnesite vazo tezo v kg:'))
visina = float(input('Vnesite vazo visino v m:'))

# preverimo, ce je racun mozen
if visina > 0:
    # izracun
    ITM = float(teza) / visina ** 2

    # izpisi
    print('Tehtate:', teza, 'kg')
    print('Visoki ste', visina, 'm')
    print('Vas ITM je', ITM, 'kg/m2')
else:
    print('Z vazo visino je nekaj hudo narobe!')
print('Se kdaj me uporabi. Nasvidenje!')
```

prg03e.py

ITM – gnezdenje pogojnih stavkov (1)

- Indeks ITM običajnemu smrtniku ne pove kaj dosti. Lepo (ali nesramno, kakor hočete) od programa bi bilo, da nam pove diagnozo.
- Zdravniki pravijo takole:
 - $ITM < 18,5$: podhranjenost
 - $18,5 \leq ITM < 25$: normalna prehranjenost
 - $25 \leq ITM < 30$: zmerna debelost
 - $30 < ITM$: huda debelost

ITM – gnezdenje pogojnih stavkov (2)

- Del našega programa, ki nam poda tudi diagnozo.

prg03f.py

```
if visina > 0:
    # izracun
    ITM = teza / visina ** 2
    # izpisi
    print('Tehtate:', teza, 'kg')
    print('Visoki ste', visina, 'm')
    print('Vas ITM je', ITM, 'kg/m2')
    # diagnoza
    if ITM < 18.5:
        print('Bo treba kaj dobrega pojest.')
    else:
        if ITM >= 18.5 and ITM < 25:
            print('Brez pripomb.')
        else:
            if ITM >= 25 and ITM < 30:
                print('Prijetne obline.')
            else:
                print('Tezke kosti.')
else:
    print('Z vaso visino je nekaj hudo narobe')
print('Hvala, se kdaj me uporabi.')
```

ITM – gnezdenje pogojnih stavkov (3)

- Tole je bolj pregledno

```
if visina > 0:
    # izracun
    ITM = teza / visina ** 2
    # izpisi
    print('Tehtate:', teza, 'kg')
    print('Visoki ste', visina, 'm')
    print('Vas ITM je', ITM, 'kg/m2')
    # diagnoza
    if ITM < 18.5:
        print('Bo treba kaj dobrega pojest.')
    elif ITM >= 18.5 and ITM < 25:
        print('Brez pripomb.')
    elif ITM >= 25 and ITM < 30:
        print('Prijetne obline.')
    else:
        print('Tezke kosti.')
else:
    print('Z vaso visino je nekaj hudo narobe')
print('Hvala, se kdaj me uporabi.')
```

prg03g.py

ITM – gnezdenje pogojnih stavkov (4)

- da se še lepše

```
if visina > 0:
    # izracun
    ITM = teza / visina ** 2
    # izpisi
    print('Tehtate:', teza, 'kg')
    print('Visoki ste', visina, 'm')
    print('Vas ITM je', ITM, 'kg/m2')
    # diagnoza
    if ITM < 18.5:
        print('Bo treba kaj dobrega pojest.')
    elif ITM < 25:
        print('Brez pripomb.')
    elif ITM < 30:
        print('Prijetne obline.')
    else:
        print('Tezke kosti.')
else:
    print('Z vaso visino je nekaj hudo narobe')
print('Hvala, se kdaj me uporabi.')
```

prg03h.py

Še en primer gnezdenja

- Napišimo program, ki podano številčno oceno napiše v opisni obliki
 - 1 ...5: nezadostno
 - 6: zadostno
 - 7: dobro
 - 8: prav dobro
 - 9: prav dobro
 - 10: odlično

Še en primer gnezdenja

Rešitev

```
# program za podano številčno oceno
# izpise opisno oceno

# vnos številčne ocene
ocena = int(input('Vnesite številčno oceno:'))

# izpis opisne ocene
if 0 < ocena <= 5:
    print('Nezadostno.')
elif ocena == 6:
    print('Zadostno.')
elif ocena == 7:
    print('Dobro.')
elif 8 <= ocena <= 9:
    print('Prav dobro.')
elif ocena == 10:
    print('Odlično.')
else:
    print('Tale ocena me pa resno skrbi!')
```

prg04.py

Python – zanke (1)

- Kaj pa, če bi želeli, da se naš program večkrat ponovi?

- Izpišimo vsa števila od 1 do 10

```
print(1)
print(2)
print(3)
print(4)
print(5)
print(6)
print(7)
print(8)
print(9)
print(10)
```

prg05a.py in prg05b.py

- Tole je že malo manj potratno

```
print('1\n2\n3\n4\n5\n6\n7\n8\n9\n10')
```

- `\n` pomeni pojdi v novo vrsto

- Kaj pa, če bi želeli izpisati 100 števil?

Python – zanke (2)

- V slovenščini bi napisali nekaj takega
 1. postavi x na 1
 2. dokler je x manjši ali enak 10:
 - a. izpiši vrednost x
 - b. povečaj x za 1
 3. izpisi, da je izpis končan
- V pythonovski angleščini pa nekaj takega

```
x = 1
while x <= 10:
    print(x)
    x = x + 1
print('Tako. Kocano je.')
```

prg05c.py

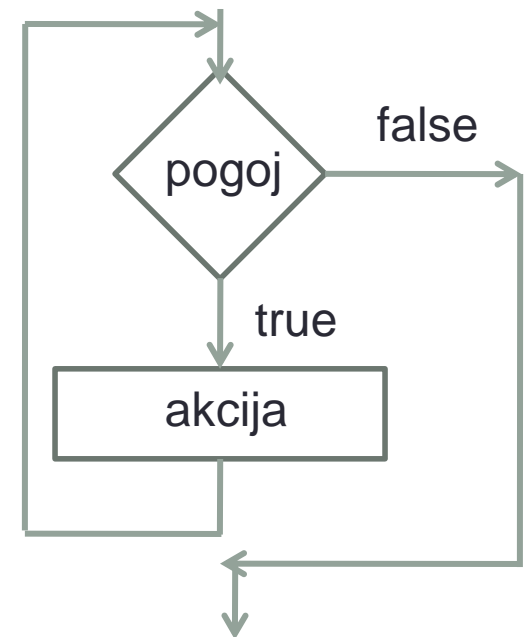
Python – zanke (3)

- Zakaj
 - izognili smo se vsakršnemu ponavljanju
- Če 10 zamenjamo s spremenljivko, je program popolnoma splošen

```
meja = int(input('Določite mejo za izpis števil:'))  
  
x = 1  
while x <= meja:  
    print(x)  
    x = x + 1  
print('Tako. Končano je.')
```

Python – zanke (4)

- Spoznali smo zanko while
 - Sestavljena je iz
 - pogoja (tega že poznamo iz stavka if) in
 - Telesa (akcija)
 - Telo zanke se izvaja dokler je izpolnjen pogoj
 - Tako kot pri stavku if moramo tudi tu pravilno zamakniti telo zanke



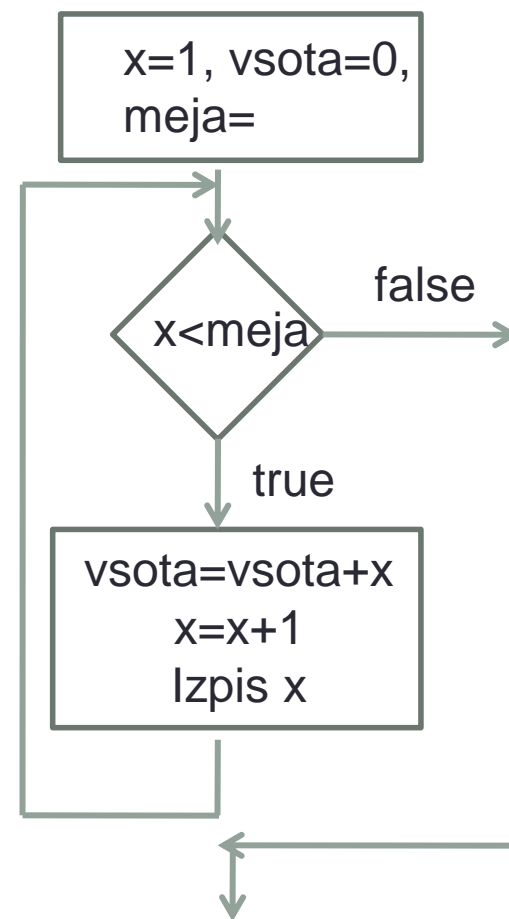
Python – zanke (5)

- Poskusimo števila od 1 do meja zdaj še sešteti!
 - potrebujemo spremenljivko za vsoto
 - ta je na začetku postavljena na nič
 - v vsakem koraku vsoto povečamo

- V matematiki lahko vsoto izračunamo

$$\text{vsota} = (\text{meja}+1) * \text{meja} / 2$$

- vendar v tem primeru ne spoznamo zank



Python – zanke (6)

```
meja = int(input('Dolocate mejo za izpis stevilk:'))

vsota = 0
x = 1
while x <= meja:
    vsota = vsota + x
    print('x = ', x, ', delna vsota =', vsota)
    x = x + 1
print('Končna vsota je', vsota)

print('Končna vsota je tudi', (meja+1)*meja/2)
```

Python – zanke (7)

- Funkcija `range()`

- `range(10)` [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
- `range(5, 10)` [5, 6, 7, 8, 9]
- `range(0, 10, 3)` [0, 3, 6, 9]
- `range(-10, -100, -30)` [-10, -40, -70]

- Poskusimo števila od 1 do meja zdaj izpisati z zanko:

`for x in range (1, n)`, izvede se n-1 ponovitev

```
meja = int(input('Koliko števil naj izpišem?'))
for x in range(1, meja + 1):
    print(x)
print('Tako. Končano je.')
```

prg13b.py

Python – zanke (8)

- Uporabnik ne sme uporabljati programa, dokler ne vtipka gesla

```
geslo = input('Kdo gre?')

while geslo != 'uiktnt':
    print('Kar zunaj ostani!')
    geslo = input('Kdo gre?')
print('Izvoli naprej!')
```

- Da se tudi krajše, brez ponavljanja vprašanja

```
while input('Kdo gre?') != 'uiktnt':
    print('Kar zunaj ostani!')
print('Izvoli naprej!')
```

- Program sprašuje, dokler uporabnik ne ugame gesla. Bolje bi bilo, da bi se po treh poskusih zaprl.

prg07a.py in prg07b.py

Python – zanke (9)

- Vstopno geslo z omejitvijo števila poskusov

```
poskusov = 3
while (input('Kdo gre?') != 'uiktnt') and (poskusov > 0):
    poskusov = poskusov - 1
    print('Imas se', poskusov, 'poskusov.')
if poskusov == 0:
    print('Kar zunaj ostani!')
else:
    print('Izvoli naprej!')
```

prg07c.py

- Vpisovali smo lahko 4-krat.
 - Če smo v četrtem poskusu vpisali pravo geslo, nam računalnik še vedno ne zaupa.
- V čem je težava?
 - Zamenjajmo pogoja

Python – zanke (10)

- Vstopno geslo z omejitvijo števila poskusov - drugič

```
poskusov = 3
while (poskusov > 0) and (input('Kdo gre?') != 'uiktnt'):
    poskusov = poskusov - 1
    print('Imas se', poskusov, 'poskusov.')
if poskusov == 0:
    print('Kar zunaj ostani!')
else:
    print('Izvoli naprej!')
```

prg07d.py

- Zakaj zdaj dela tako kot je treba?
 - Python pogoje računa od leve proti desni.
 - Ko ugotovi, kakšen bo rezultat, se s pogojem ne obremenjuje več.
 - Torej, ko pogoj `(poskusov > 0)` ni izpolnjen, pogoja `(input('Kdo gre?') != 'uiktnt')` ne preverja več.

Python - prekinitev zanke

- Stavek **break** - izvajanje zanke prekinemo preden pridemo do pogoja

```
# vstopna kontrola še enkrat
poskusov = 3
while poskusov > 0:
    geslo = input('Kdo gre?')
    if geslo == 'uiktnt':
        break
    poskusov = poskusov - 1
    print('Imaš še', poskusov, 'poskusov.')

if poskusov > 0:
    print('Vstopi, prosim!')
else:
    print('Kar zunaj ostani!')
```

Program – ugibanje števil

- Napišimo svojo prvo igrico – ugibanje števil
 - računalnik si izmisli naključno številko med 1 in 100
 - mi jo moramo v čim manj poskusih uganiti
 - zato, da si računalnik izmisli številko moramo naložiti modul random in poklicati funkcijo, ki vrne naključno število
 - primer

```
>>> from random import *
>>> random()
0.22250607529732191
>>> random()
0.88965905354138841
>>> randint(1, 100)
1
>>> randint(1, 100)
6
```

Program – ugibanje števil – V1

```
# racunalnik si izmisli stevilko
from random import *
skrita = randint(1, 100)

# ugibanje uporabnika
poskusov = 0
stevilka = int(input('Vpisi stevilko:'))
while stevilka != skrita:
    poskusov = poskusov + 1
    if skrita > stevilka:
        print('Iskana stevilka je vecja od', stevilka)
    elif skrita < stevilka:
        print('Iskana stevilka je manjsa od', stevilka)
    stevilka = int(input('Vpisi stevilko:'))
print('Bravo, uganil si v', poskusov, ' poskusih!')
print('Skrita stevilka je bila', skrita)
```

Program – ugibanje števil - V2

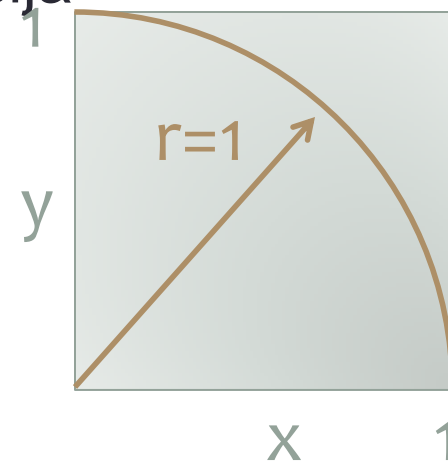
```
# racunalnik si izmisli številko
from random import *
skrita = randint(1, 100)

# ugibanje uporabnika
poskusov = 0
pogoj = True
while pogoj:
    stevilka = int(input('Vpisi številko:'))
    poskusov = poskusov + 1
    if skrita > stevilka:
        print('Iskana številka je večja od', stevilka)
    elif skrita < stevilka:
        print('Iskana številka je manjša od', stevilka)
    pogoj = (stevilka != skrita)
print('Bravo, uganil si v', poskusov, 'poskusih!')
print('Skrita številka je bila', skrita)
```

prg08b.py

Program – izračun PI (1)

- Tudi s kockanjem se da izračunati število π
- Postopek znanstveno imenujemo integracija Monte Carlo
- Ideja:
 - imamo kvadrat s stranico 1
 - v njem je vrisana četrtna kroga
 - naključno določimo x in y
 - če je $x^2 + y^2 < 1$, potem smo zadeli krog
 - $PI = 4 \times (\text{število zadetkov v krog}) / (\text{število vseh poskusov})$
- Poskusimo



Program – izračun PI (2)

```
# knjiznica s funkcijo za nakljucna stevila
from random import *

# stevce postavimo na nic
strelov = 0
zadetekov = 0

# streljanje
while strelov < 1000000:
    x = random()
    y = random()
    r2 = x ** 2 + y ** 2
    if r2 < 1:
        zadetekov = zadetekov + 1
    strelov = strelov + 1

# izpis
print('Stevilo pi je', 4.0*zadetekov/strelov)
```

prg09.py

Program – izračun PI (3)

```
# knjižnica s funkcijo za naključna števila
from random import *

# števce postavimo na nič
strellov = 1000000
zadetekov = 0

# streljanje
for i in range(strellov):
    x = random()
    y = random()
    r2 = x ** 2 + y ** 2
    if r2 < 1:
        zadetekov = zadetekov + 1

# izpis
print('Število pi je', 4*zadetekov/strellov)
```

prg15.py

Naloga 1

- Še malo za oči - pravokotni trikotnik obrnjen na glavo

```
*****
*****
****
***
**
*
```

```
velikost = int(input('Vnesi višino trikotnika:'))
for i in range(velikost, 0, -1):
    print('*' * i)
```

- risanje kepe

prg19.py in prg20.py

radij=5

```
from math import *
razmerjexy = 2
radij = int(input('Vnesi velikost kepe:'))
```

```

**
*****
*****
*****
*****
*****
*****
*****
*****
*****
**
```

```
for x in range(-radij, radij + 1):
    y = sqrt(radij ** 2 - x ** 2)
    zvezdice = int(y * razmerjexy) + 1
    presledki = radij * razmerjexy - zvezdice + 1
    print(' ' * presledki, end='')
    print('*' * zvezdice * razmerjexy)
```