Knihovny

January 10, 2024

1 Knihovny

Příklad: knihovna math

Různé možnosti volání

[1]:
```
import math
math.sin(0)
```

[1]: 0.0

[2]:
```
import math as m
m.sin(0)
```

[2]: 0.0

[3]:
```
from math import sin, cos    # naimportujeme metodu sin
sin(0)                       # POZOR na nejednoznacnost jmen
```

[3]: 0.0

[ ]:
```
from math import *       # naimportujeme vse
sin(0)                   # POZOR na nejednoznacnost jmen
```

[ ]:
```

```

Importovat můžeme také vlastní funkce

Např. obsah souboru podil.py:

[ ]:
```
def deleni(x, y):
    try:
        print(x/y)
    except ZeroDivisionError:
        print('POZOR : Deleni nulou!')
```

Chceme využít funkci v makru volani.py:

[ ]:
```
from podil import *
deleni(10, 3)
```

1.1 Knihovna os

Interakce s operačním systémem

[4]:
```
import os
```

[5]:
```
os.getcwd()
```

[5]:
```
'/home/jana/Programovani/Python/MFF/NMIN111/2023_24'
```

[ ]:
```
os.listdir()
```

[ ]:
```
os.chdir('/home/jana')
```

[ ]:
```
os.system('mkdir TestDirectory')
```

1.2 Knihovna random

Generátor pseudonáhodných čísel

[6]:
```
import random
```

[8]:
```
random.random()
```

[8]:
```
0.5942892193015453
```

[9]:
```
random.uniform(0, 100)
```

[9]:
```
32.17389432417488
```

[10]:
```
random.randint(0, 100)
```

[10]:
```
76
```

[ ]:
```
random.seed(12345)
```

[ ]:
```
random.random()
```

Pseudonáhodný výběr

[11]:
```
random.choice(['banan', 'jablko', 'hruska'])
```

[11]:
```
'jablko'
```

[12]:
```
random.choices('012345', k = 5)    # vyber s vracenim zpet
```

[12]:
```
['5', '0', '3', '3', '3']
```

[13]:
```
random.sample('012345', k = 5)    # vyber bez vraceni zpet
```

[13]:
```
['5', '0', '3', '4', '1']
```

[14]:
```
random.sample(range(100), 10)
```

[14]:
```
[37, 24, 89, 97, 4, 44, 19, 45, 75, 62]
```

[ ]:
```

```

1.3 Příklady dalších modulů

- argparse - zpracování argumentů pro příkazovou řádku
- datetime, calendar - data, čas
- array - homogenní pole
- cmath - počítání s komplexními čísly
- statistics - statistické funkce
- re - regulární výrazy pro práci s textem
- turtle - želví grafika

1.4 Úkoly

- Simulujte 1000 hodů kostkou. Spočítejte výskyty každého čísla.
- **Aproximujte číslo pi**: Generujte náhodně body ve čtverci [-1, 1] x [-1,1]. Spočítejte,kolik z nich padlo do jednotkového kruhu a s touto znalostí určete hodnotu pi.
- Vygenerujte náhodně permutaci na množině 1 až N a spočítejte, kolik má pevných bodů. (Jako pevný bod označujeme bod, který se v daném zobrazení zobrazí sám na sebe.) Opakováním experimentu odhadněte, jaká je pravděpodobnost, že náhodná permutace nemá pevný bod.
- **Soutěž o nejhezčí obrázek s využitím želví grafiky.**

[15]:
```
# Ukazka: zelvi grafika

"""
Zelvi grafika - modul turtle
Priklad: ctvercova spirala
Prikazy pro zelvu:
forward(10): pohyb o 10 pixelu dopredu
backward(10): pohyb o 10 pixelu dozadu
right(35): otoceni po smeru hodin. rucicek, 35 - uhel ve stupnich
left(35): otoceni proti smeru hodin. rucicek
penup(): zvednuti pera
pendown(): pero dolu
goto(x,y): posun na pozici (x,y)
home(): presun do stredu
"""
# Nacteme moduly turtle a random
import turtle
# Modul random pro nahodny vyber barvy (neni nutne)
```

```
import random

# t je objekt typu Turtle (zelva)
t = turtle.Turtle()
# Kurzor se bude vykreslovat jako zelva
t.shape('turtle')

# Barvy tuzky zelvy
barvy = ['red','green','blue','cyan','black','brown','yellow']

# Ukazka: vykresleni ctverce
n = 4
for i in range(0,n):
    # Vybereme nahodne barvu, kterou nakreslime hranu
    t.pencolor(random.choice(barvy))
    # pohyb dopredu o zvetsujici se cislo
    t.forward(200)
    # otoceni proti smeru hodinovych rucicek o 90 stupnu
    t.left(90)

# Ukonceni zelvy
turtle.done()
```

[19]:
```
# Aproximace pi
import random

def pi(n_pokus):
    n_all = 0
    n_in = 0

    for i in range(n_pokus):
        x = random.uniform(-1, 1)
        y = random.uniform(-1, 1)
        n_all += 1
        if x**2 + y**2 < 1.0:
            n_in += 1

    pi_cislo = n_in/n_all*4

    return pi_cislo

pi(100000)
```

[19]: 3.14112

[ ]: