

Lab4. Wątki w Pythonie

1. Realizacja wątków - przykład pierwszy

```
# import threading

class Hello():          # class Hello(threading.Thread)
    def run(self):
        for i in range(10):
            print('hello')

class Hi():              # class Hi(threading.Thread)
    def run(self):
        for i in range(10):
            print('hi')

t1 = Hello()
t2 = Hi()

t1.run()                 # t1.start()
t2.run()                 # t2.start()

                           # t1.join()
                           # t2.join()

print('stop')
```

2. Realizacja wątków – przykład drugi

```
import time
import threading

def proc():
    print('sleeping ...')
    time.sleep(1)
    print('done')

start = time.perf_counter()

proc()
proc()

# t1 = threading.Thread(target=proc)
# t2 = threading.Thread(target=proc)

# t1.start()
# t2.start()

# t1.join()
# t2.join()

stop = time.perf_counter()
print('time:', stop-start)
```

3. Uruchamianie wielu wątków

```
import time
import threading

def proc(num):
    print(num, 'sleeping ...')
    time.sleep(1)
    print(num, 'done')

start = time.perf_counter()

threads = []
for n in range(10):
    t = threading.Thread(target=proc, args=[n])
    t.start()
    threads.append(t)

for thread in threads:
    thread.join()

stop = time.perf_counter()
print('time:', stop-start)
```

4. Zastosowanie praktyczne – współbieżne pobierania plików

Proszę przekształcić poniższy program aby korzystał z wątków. Proszę zaobserwować różnicę w czasach działania obu programów

```
import requests
import time

urls = [
    'https://images3.alphacoders.com/103/103147.jpg',
    'https://images4.alphacoders.com/975/97548.jpg',
    'https://images4.alphacoders.com/810/81006.jpg',
    'https://images7.alphacoders.com/423/423348.jpg',
    'https://images7.alphacoders.com/381/381455.jpg',
    'https://images.alphacoders.com/868/86853.jpg',
    'https://images5.alphacoders.com/104/1043977.jpg',
    'https://images4.alphacoders.com/788/788878.jpg',
    'https://images8.alphacoders.com/419/419522.jpg',
    'https://images2.alphacoders.com/475/475841.jpg',
    'https://images6.alphacoders.com/595/595234.jpg',
    'https://images.alphacoders.com/872/872716.jpg',
    'https://images2.alphacoders.com/462/462942.jpg',
    'https://images4.alphacoders.com/832/83206.jpg',
    'https://images2.alphacoders.com/861/861016.jpg'
]
```

```

def download(url):
    fname = url.split('/')[-1]
    print(fname,end='')
    buf = requests.get(url).content
    f = open(fname,'wb')
    f.write(buf)
    f.close()
    print(' ok')

start = time.perf_counter()

for url in urls:
    download(url)

stop = time.perf_counter()
print('time:',stop-start)

```

5. Zastosowanie praktyczne – ping

Proszę przekształcić poniższy program aby korzystał z wątków. Proszę zaobserwować różnicę w czasach działania obu programów

```

import os
import time

def ping(ip):
    pingaling = os.popen("ping -n 1 "+ip,"r")
    while True:
        line = pingaling.readline()
        if not line:
            break
        if line.find('Average')>0:
            print(ip)

start = time.perf_counter()
for host in range(1,20):
    ip = "192.168.1."+str(host)
    ping(ip)

stop = time.perf_counter()
print('time:',stop-start)

```

6. Proszę wykonać poniższe zadanie pochodzące z książki D. Van Tassel „Praktyka programowania” z użyciem wątków.

48. Wyspa Wilków to obszar 20 na 20 zamieszkały przez żyjące na wolność króliki, wilki i wilczyce. Na początku nieco o reprezentantach każdej z tych grup rozproszonych po wyspie. Króliki są raczej głupie: po upływie określonego czasu przenoszą się z równym prawdopodobieństwem na jedno z sąsiednich pól (ośmiu, chyba że znajdują się na wybrzeżu). Przez 1/9 czasu siedzą więc po prostu spokojnie. Każdy królik ma 20% szans przekształcenia się w dwa króliki. Każda wilczyca również porusza się losowo, chyba że na jednym z ośmiu sąsiednich pól znajduje się królik. Wówczas wilczyca rozpoczyna pościg. Jeżeli znajdzie się na tym samym polu co królik, to zjada go i zyskuje jedną porcję tłuszczu; jeżeli nie, to traci 0.1 porcji tłuszczu. Wilki pozbawione tłuszczu giną. Każdy wilk zachowuje się tak, jak wilczyca, chyba że w pobliżu nie ma królika, natomiast jest wilczyca; wówczas wilk rzuca się w pogoń. Jeżeli wilk i wilczyca znajdują się na tym samym polu i nie będzie tam królika do zjedzenia, to produkują potomka o losowej płci. Zaprogramuj omówioną symulację ekologiczną i zaobserwuj stan populacji po kilku kolejnych okresach. (Dziękuję Billowi McKeemanowi za to zadanie).
49. Symulacja z poprzedniego zadania jest z natury niestabilna (Wyspa Wilków musi się stać pustynią). Dodaj żywopłot (obszar niedostępny dla wilków) i zaobserwuj wyniki.

Początek programu ...

```
#!/usr/bin/python3
# -*- coding: utf-8 -*-

from threading import Thread

import time
import random
import os

def printxy(x,y,s):
    print("\033["+str(y+1)+";" +str(x+1)+"f"+s)

def clrscr():
    print(chr(27)+"[2J")

class Krolik(Thread):
    def __init__(self,x,y):
        self.x=x
        self.y=y
        Thread.__init__(self)

    def run(self):
        while (True):
            time.sleep(1)
            nx = self.x+random.randint(-1,1)
            ny = self.y+random.randint(-1,1)
            if not (0<=nx<N and 0<=ny<N):
                continue
            printxy(self.x,self.y," ")
```

```

        wyspa[self.x][self.y] = " "
        if wyspa[nx][ny]=='k':
            # break          # królik ginie
            # k=Krolik(x,y)  # powstaje nowy królik
            # k.start()
            pass
        self.x = nx
        self.y = ny
        printxy(self.x,self.y,"k")
        wyspa[self.x][self.y] = "k"
# end class

if __name__ == '__main__':
    clrscr()
    os.system('setterm -cursor off')

    N = 20
    wyspa = [[" " for _ in range(N)] for _ in range(N)]

    for n in range(10):
        x = random.randint(0,N-1)
        y = random.randint(0,N-1)
        wyspa[x][y] = "k"
        k=Krolik(x,y)
        k.start()
        time.sleep(0.1)

    input()
    print("stop")
    os.system('setterm -cursor on')

```