Bevezetés a programozásba

10. Gyakorlat

Ismétlés

- Halmazok – set()
  - Unió
  - Metszet
  - Különbség
  - Szimmetrikus különbség
- Szótárak – dict()
  - kulcs-érték
- Funkcionális programozás
  - Lambda függvények

1. Házi Feladat

Mindennap egy alma az orvost távol tartja

- https://progcont.hu/progcont/100358/?pid=201544

Megoldás

```
import sys

gyerekek = {}
for line in sys.stdin:
    adat = line.split(":")
    nevek = adat[1].strip().split(",")
    for i in range(len(nevek)):
        if nevek[i] in gyerekek:
            gyerekek[nevek[i]] += 1
        else:
            gyerekek[nevek[i]] = 1
for kulcs, ertek in sorted(gyerekek.items()):
    print(f"{kulcs}: {ertek}")
```

2. Házi Feladat

Új mintatanterv

- https://progcont.hu/progcont/100278/?pid=201288

Megoldás

```
n = int(input())
tantargyak = {}

for i in range(n):
    adat = input().split(":")
    tantargyak[adat[0]] = adat[1]

for kulcs, ertek in sorted(tantargyak.items()):
    print(ertek)
```

3. Házi Feladat

Erdei lemmingek

- https://progcont.hu/progcont/100337/?pid=201476

Megoldás

```
N = int(input())
zoos = {}
for i in range(N):
    line = input().split(':')
    zoos[line[0]] = int(line[1])

M = int(input())
for i in range(M):
    line = input().split(':')
    sender_zoo, receiver_zoo = line[0], line[1]
    db = int(line[2])
    if sender_zoo in zoos and zoos[sender_zoo] >= db:
        zoos[sender_zoo] -= db
        zoos[receiver_zoo] += db

for kulcs, ertek in sorted(zoos.items()):
    print(f"{kulcs}:{ertek}")
```

Parancssori argumentumok

- Ha egy programot parancssorból futtatunk, átadhatunk neki paramétereket, úgynevezett **parancssori argumentumokat.**

- A parancssori argumentumok kezelésének három leggyakoribb módja:
  - **sys.argv,**
  - getopt,
  - argparse modulok.

Parancssori argumentumok sys.argv

argc – parancssori argumentumok száma -> int

sys.argv[0] – program neve/helye -> str
sys.argv[1] – első parancssori argumentum -> str
sys.argv[2] – második parancssori argumentum -> str
…
sys.argv[n] – n.ik parancssori argumentum -> str

Parancssori argumentumok

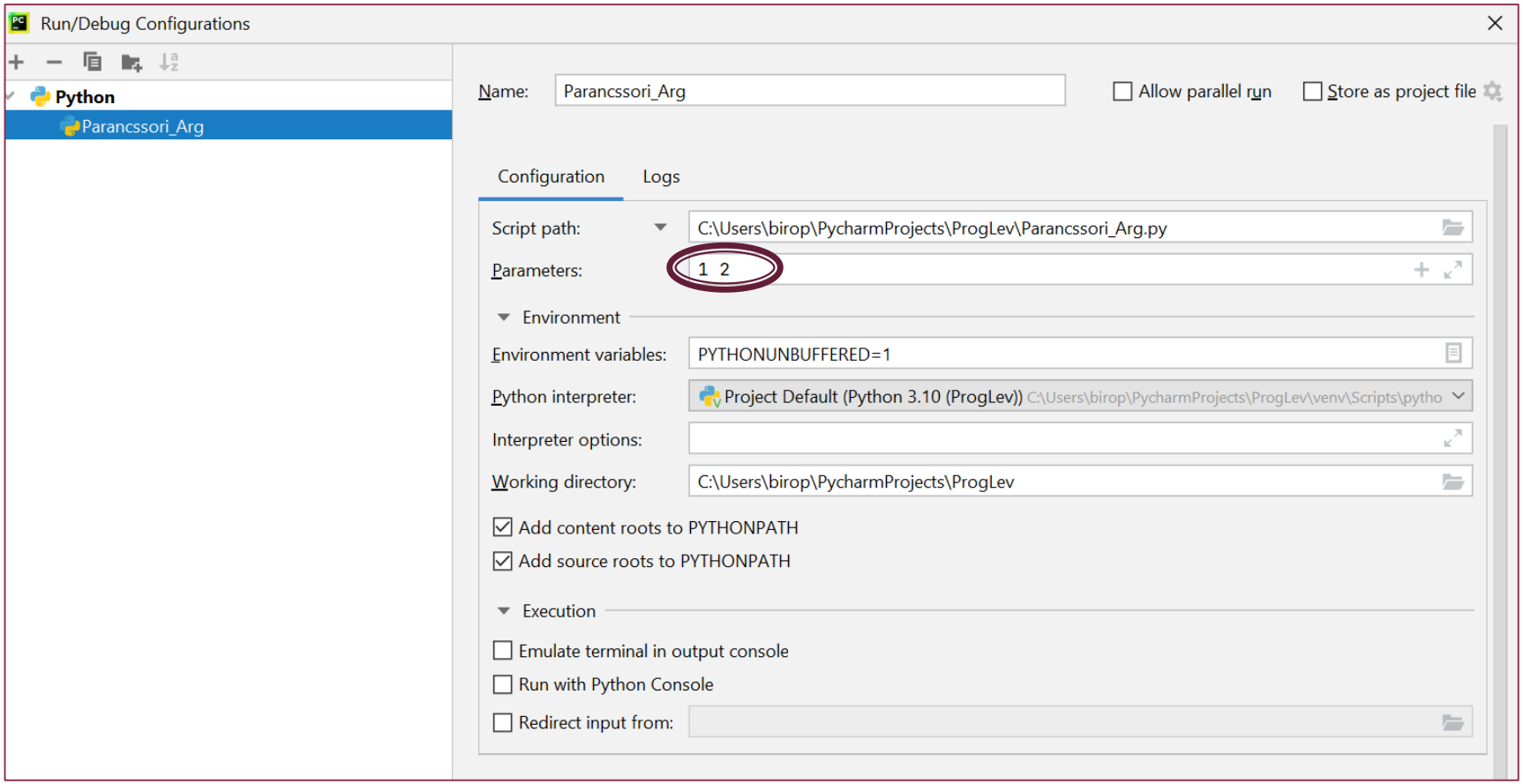
```
import sys

argc = len(sys.argv)
print("A parancssori argumentumok száma:", argc)

print("Fájl helye:", sys.argv[0])

print("Átadott argumetumok:")
for i in range(1, argc):
    print(sys.argv[i])
```

Parancssori argumentumok



Parancssori argumentumok

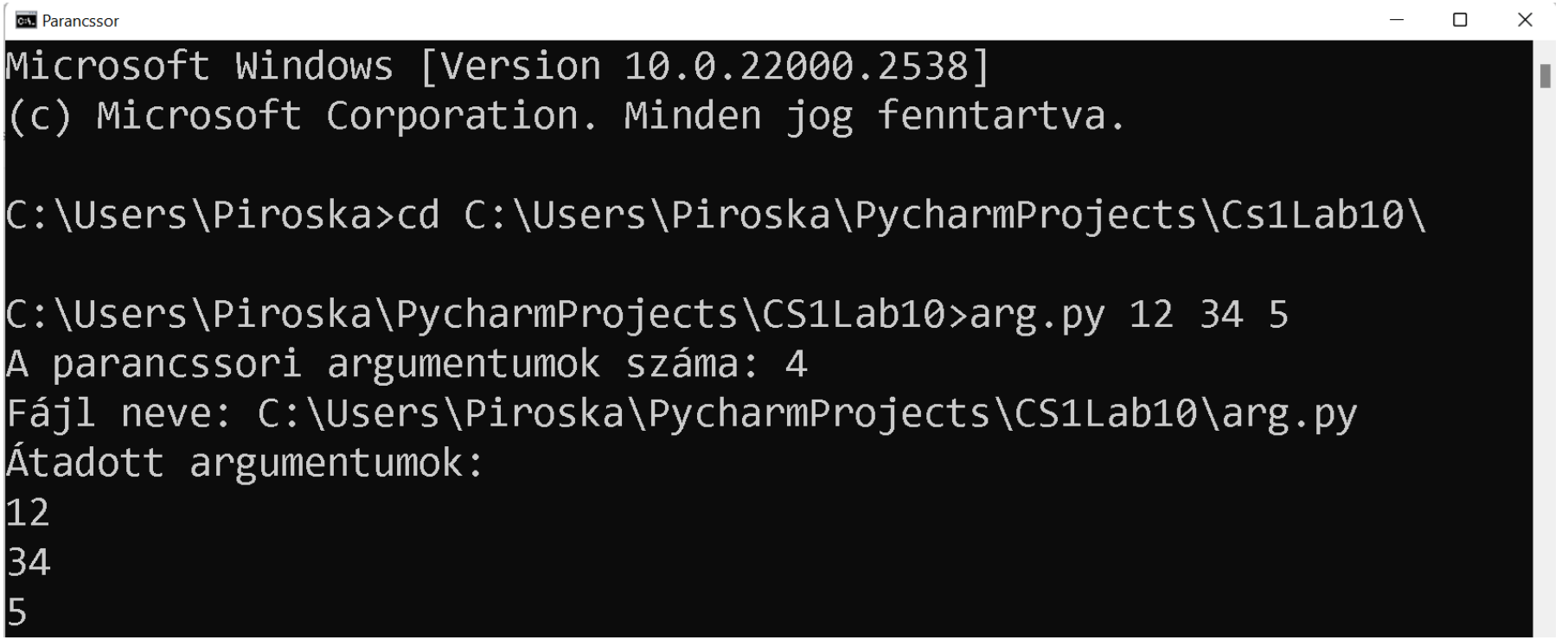
Futtatás – Total Commander -> CMD:

- c:/Users/birop/PycharmProjects/Cs1Lab10/arg.py 1 2

Kimenet:

- A parancssori argumentumok száma: 3
- Fájl helye: c:/Users/birop/PycharmProjects/Cs1Lab10/arg.py
- Átadott argumetumok:
- 1
- 2

CMD megnyitása



Parancssori argumentumok

- Határozzuk meg a parancssori argumentumok szorzatát.

Parancssori argumentumok

```
import sys

argc = len(sys.argv)
print("A parancssori argumentumok száma:", argc)

p = 1
for i in range(1, argc):
    p = p * int(sys.argv[i])
print("Szorzat = ", p)
```

Páratlan elemek száma

A függvény

- Írj egy **count_of_odds** nevű függvényt, mely visszaadja azt, hogy a parancssori argumentumok közül mennyi páratlan szám!

Paraméterlista

- A függvény paraméterlistája üres, azaz egyetlen értéket sem kap meg paraméterként.

Visszaadott érték

- a feltételnek megfelelő elemek száma

https://viskillz.inf.unideb.hu/prog/#/?exercise=P108101c&page=sheet&week=P1081

Páratlan elemek száma

```
import sys

def count_of_odds() -> int:
    count = 0
    for number in sys.argv[1:]:
        if int(number) % 2 == 1:
            count += 1
    return count

def main():
    print(count_of_odds())

if __name__ == '__main__':
    main()
```

Páratlan elemek száma

```
import sys

def count_of_odds(numbers: list[int]) -> int:
    count = 0
    for number in numbers:
        if number % 2 == 1:
            count += 1
    return count

def main():
    numbers = []
    argc = len(sys.argv)
    for i in range(1, argc):
        numbers.append(int(sys.argv[i]))

    print(count_of_odds(numbers))

if __name__ == "__main__":
    main()
```

Prím elemek száma

A függvény

- Írj egy **count_of_primes** nevű függvényt, mely visszaadja azt, hogy a parancssori argumentumok közül mennyi prímszám!

Paraméterlista

- A függvény paraméterlistája üres, azaz egyetlen értéket sem kap meg paraméterként.

Visszaadott érték

- a feltételnek megfelelő elemek száma

https://viskillz.inf.unideb.hu/prog/#/?exercise=P108103c&page=sheet&week=P1081

Prím elemek száma

```
import sys
import math

def is_prime(n: int) -> bool:
    if n == 2:
        return True
    if n < 2 or n % 2 == 0:
        return False
    for i in range(3,
        int(math.sqrt(n)), 2):
        if n % i == 0:
            return False
    return True
```

```
def count_of_primes() -> int:
    count = 0
    for number in sys.argv[1:]:
        if is_prime(int(number)):
            count += 1
    return count

def main():
    print(count_of_primes())

if __name__ == '__main__':
    main()
```

Prím elemek száma

```python
import sys
import math

def is_prime(n: int) -> bool:
    if n == 2:
        return True

    if n < 2 or n % 2 == 0:
        return False

    for i in range(3,
        int(math.sqrt(n)), 2):
        if n % i == 0:
            return False

    return True
```

```python
def count_of_primes(numbers: list[int]) ->
int:
    count = 0
    for number in numbers:
        if is_prime(number):
            count += 1
    return count

def main():
    numbers = []
    argc = len(sys.argv)
    for i in range(1, argc):
        numbers.append(int(sys.argv[i]))

    print(count_of_primes(numbers))

if __name__ == "__main__":
    main()
```

Állomány kezelés

- **Fájl megnyitása**
- Fájlok megnyitására az **`open()`** függvényt használhatjuk, amely egy fájl objektumot ad vissza.
  - Első paraméterként a megnyitandó fájl nevét veszi át.
  - Második paraméterként megadhatjuk a megnyitás módját:
    - olvasás (read): **`'r'`**,
    - írás (write): **`'w'`**,
    - hozzáfűzés (append): **`'a'`**.
    - (Az alapértelmezett érték az olvasás, azaz 'r’.)

```
file = open("teszt.txt")
file = open("c:/BevProg/teszt.txt")
file = open("teszt.txt", encoding="utf-8")
```

Állomány adatainak beolvasása

Az állomány teljes tartalmának beolvasása:

```
with open('teszt.txt', encoding="utf-8") as file:
    lines = file.read()
    print(lines)
```

Állomány adatainak beolvasása

Az állomány első sorának beolvasása:

```
with open("teszt.txt", encoding="utf-8") as file:
    line = file.readline().strip("\n")
    print(line, len(line))
```

Állomány adatainak beolvasása

Az állomány összes sorának beolvasása, eredmény egy sztringeket (állomány sorait) tartalmazó lista:

```
with open("teszt.txt", encoding="utf-8") as file:
    lines = file.readlines()
    print(lines)
```

Állomány adatainak beolvasása

Az állomány soronként való beolvasása:

```
with open("teszt.txt", encoding="utf-8") as file:
      for line in file:
         print(line.strip("\n"))
```

Állomány kezelés

- **Fájl létrehozása**

- Fájlba írni a fájl objektum `write()` függvényével lehet, ez hasonlóan működik a már ismert print() függvényhez.

-

- Arra azonban figyelni kell, hogy a `write()` függvénynek csak egy sztring paramétere lehet, tehát ha egy sorban több változó értékét akarjuk kiírni, konkatenálást vagy sztring formázást kell alkalmaznunk.

Állományba való írás

Állományba való írása – egy sor:

```
with open("ki.txt", "w") as file:
    file.write("alma\n")
```

Állományba való írás

Állományba való írása – több sor:

```
with open("ki.txt", "w") as file:
    file.write("alma\n")
    file.writelines(["banan\n", "citrom\n"])
```

Kivételkezelés

```
try:
    with open("ki3.txt", encoding="utf-8") as file:
        for line in file:
            print(line.strip("\n"))
except FileNotFoundError:
    print("Nem létezik a fájl!")
```

Legnagyobb közös osztó

- https://progcont.hu/progcont/100132/?pid=200801

Legnagyobb közös osztó – Megoldás

```python
import sys

def lnko(a: int, b: int) -> int:
    while a != b:
        if a > b:
            a = a - b
        else:
            b = b - a
    return a

def main():
    with open(sys.argv[1]) as file:
        for line in file:
            numbers = line.strip("\n").split(" ")
            lnk = int(numbers[0])
            for i in range(1, len(numbers)):
                lnk = lnko(lnk, int(numbers[i]))
            print(lnk)

if __name__ == "__main__":
    main()
```

Utcahosszal jobb

- https://progcont.hu/progcont/100133/?pid=200802

Utcahosszal jobb – Megoldás

```
import sys
utcak={}

with open(sys.argv[1]) as file:
    for line in file:
        adat = line.strip("\n").split(";")
        #print(adat)
        if adat[0] in utcak:
            utcak[adat[0]]+=float(adat[2])
        else:
            utcak[adat[0]]=float(adat[2])
    #print(utcak)
for kulcs, ertek in sorted(utcak.items(),
        key=lambda rend: (-rend[1], rend[0])):
    print(kulcs)
```

1. Házi Feladat

Lokális maximumhelyek száma

- https://viskillz.inf.unideb.hu/prog/#/?week=P1081&exercise=P108109c&page=sheet

2. Házi Feladat

Körözgetés

- https://progcont.hu/progcont/100132/?pid=200799

3. Házi Feladat

Prímválogatás

- https://progcont.hu/progcont/100317/?pid=201426