

Міністерство освіти і науки, молоді та спорту України
Національний технічний університет України
«Київський політехнічний інститут»
Фізико-технічний інститут

Лабораторна робота з програмування № 6

Виконав:

Студент 2 курсу
групи ФЕ-81
Адамантіс М. В.

Перевірив:

Прогонов Д. О.

Київ 2019

Формулювання завдання

1. Порядок виконання роботи

1. Проаналізувати умову задачі.
2. Розробити алгоритм та створити програму розв'язання задачі згідно з номером варіанту.
При цьому розмістити функції програми у двох (або більше) різних файлах, та об'єднати ці файли у спільну програму двома різними способами:
 - 1) реалізувати модульний принцип програмування, створивши проект, у якому всі функції, крім головної, містяться в окремому файлі;
3. Результати роботи оформити протоколом.

2. Варіанти завдань

Модифікувати створену під час виконання у лабораторній роботі № 5 програму згідно з номером варіанту, використовуючи потоки файлового вводу/виводу для зчитування вхідної інформації з одного файла та запису отриманого результату в інший.

Програмний код

```
# main.py
#!/usr/bin/env python3
import sys, os, re
sys.path.append(os.path.abspath(os.path.join(os.path.dirname(__file__), "..")))
from inc.errorhandling import handle, error_message
from logic import f, my_pow

description = {
    'A': 'Enter two numbers (s and t)\n',
    'B': 'Enter two numbers (a and n)\n',
    'Usage': 'main.py [input file] [output file]\n\
[input file] format:\n\
\tmode: ...args\n\
Example: input.txt\n\
\tA: 1 2\n\
\tB: 3 4\n\
\tA: 0 0\n\
\t...\n'
}
```

```

# Main
def interpret(cmd, part):
    try:
        if cmd == "": return

        if cmd in ['A', 'B']:
            part = cmd[1]
            return description[part], part

        if part == 'A':
            s, t = [float(x) for x in cmd.split(' ')]
            return f(t, -2*s) - f(2.2, t, s - t), part

        elif part == 'B':
            a, n = [float(x) for x in cmd.split(' ')]
            n = int(n)
            return my_pow(a, n), part

        else:
            raise ValueError("Part must be either A or B")
    except Exception as e:
        handle(e)

def main():
    if len(sys.argv) != 3:
        print(description['Usage'])
        return

    del sys.argv[0]
    input_file, output_file = sys.argv

    if not os.path.exists(input_file):
        handle(ValueError('Input file doesn\'t exist'))
        return

    try:
        results = []
        with open(input_file, 'r') as data_file:

```

```

for line in data_file:
    options = re.search(r'^(A|B): (\d \d)$', line.strip(), flags=0)
    if not options:
        results.append(error_message())
        continue

    part = options.group(1)
    cmd = options.group(2)
    result = interpret(cmd, part)

    if result:
        message, part = result
        results.append(message)

with open(output_file, 'w') as f:
    f.writelines([str(str(s) + '\n') for s in results])

except KeyboardInterrupt:
    print('Bye!')
    exit(0)
except Exception as e:
    if e.__class__ == EOFError:
        print('Bye!')
        exit(0)
    handle(e)

if __name__ == "__main__":
    main()

```

logic.py

```
import math
```

Part A

```
def f(a, b, c = 1.17):
    return (2*a - b - math.sin(c))/(5 + abs(c))
```

Part B

```
def my_pow(a, n):
    if a == 0:
```

```
    raise ValueError("base number must be non-zero")
if n < 0:
    raise ValueError("power must be non-negative")
return 1 if n == 0 else a*my_pow(a, n - 1)
```

Приклад виконання

IPT-2020-FE81 on  **work/m.adamantis**

→ **python3** results/lab6/main.py

main.py [input file] [output file]

[input file] format:

mode: ...args

Example: input.txt

A: 1 2



B: 3 4

A: 0 0

...

IPT-2020-FE81 on  **work/m.adamantis** [**X!?**] via  **v4.8.2** took **10s 679ms**

→ **python3** results/lab6/main.py results/lab6/input-example.txt results/lab6/out.txt

IPT-2020-FE81 on  **work/m.adamantis** [**X!?**] via  **v4.8.2**

→ **cat** results/lab6/out.txt

0.28297191889028794

81.0

-1.0292302427449167