Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут ім. Ігоря Сікорського»

**Протокол лабораторної роботи № 6**

Принципи програмування на мові Python

**Виконав:**

Студент 2 курсу, ФТІ

Група ФЕ-71

Василенко К.Р.

**Київ 2019**

**Мета роботи: :** Оволодіти навичками роботи з файлами даних та засвоїти модульний принцип програмування.

**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.

**Завдання**

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.

**Реалізація алгоритму**

Перший код:

from firstModule import func  
import checkModule  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
  
 try:  
 file = open('input.txt', 'r') # Открыть файл на чтение.  
 except FileNotFoundError:  
 print("Файл не найден. Проверьте, находится ли файл в одной директории с программой.")  
 exit()  
 inp = [line.strip() for line in file]  
 file.close()  
  
 if len(inp) > 2:  
 print("Недостаточно данных в файле. Должно быть 2 числа. Каждое с новой строки.")  
 exit()  
 else:  
 rightInp = checkModule.checkAndInp(inp[0], inp[1])  
 f\_S, f\_T = rightInp[0], rightInp[1]  
 try:  
 res = func(1.2, f\_S) + func(f\_T, f\_S) - func(2 \* f\_S - 1, f\_S \* f\_T)  
 except ZeroDivisionError:  
 print("Введённые числа дают деление на 0. Невозможно продолжить.")  
 exit()  
 out = open('output.txt', 'w')  
 out.write("Результат: " + str(res))  
 print("Результат успешно записан в файл 'output.txt'")  
 out.close()

Модулі до коду:

№1

def func(a, b):  
 return (a\*\*2 + b\*\*2)/(a\*\*2 + 2\*a\*b + 3\*(b\*\*2) + 4)

№2 from math import isnan, isinf  
  
  
def checkAndInp(strS, strT):  
 try:  
 f\_S = float(strS)  
 except ValueError:  
 print("Скорее всего вы ввели не число.")  
 return []  
 if isnan(f\_S):  
 print("Невозможно выполнить действия с не числом (NAN). ")  
 return []  
 elif isinf(f\_S):  
 print("Невозможно выполнить действие с бесконечностью.")  
 return []  
  
  
 try:  
 f\_T = float(strT)  
 except ValueError:  
 print("Скорее всего вы ввели не число.")  
 return []  
 if isnan(f\_T):  
 print("Невозможно выполнить действия с не числом (NAN). ")  
 return []  
 elif isinf(f\_T):  
 print("Невозможно выполнить действие с бесконечностью.")  
 return []  
  
 return [f\_S, f\_T]

Другий код:

import math  
from module import getNSum  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 try:  
 file = open('input.txt', 'r')  
 except FileNotFoundError:  
 print("Такого файла не существует.")  
 exit()  
  
 data = [line.strip() for line in file]  
 file.close()  
  
 if len(data) < 3:  
 print("Недостаточно данных в файле.")  
 exit()  
  
 try:  
 f\_Elem = float(data[0])  
 except ValueError:  
 print("Скорее всего вы ввели не число.")  
 exit()  
 if math.isnan(f\_Elem):  
 print("Начальный член не может быть не числом (NAN).")  
 exit()  
 elif math.isinf(f\_Elem):  
 print("Начальный член не может быть бесконечностью.")  
 exit()  
  
 try:  
 f\_Step = float(data[1])  
 except ValueError:  
 print("Скорее всего вы ввели не число.")  
 exit()  
 if math.isnan(f\_Step):  
 print("Шаг не может быть не числом (NAN).")  
 exit()  
 elif math.isinf(f\_Step):  
 print("Шаг не может быть бесконечностью.")  
 exit()  
 elif f\_Step == 0:  
 print("При шаге равному 0 прогрессия не будет изменяться, а состоять будет из одного числа:\n {0}".format(  
 f\_Elem))  
 exit()  
  
 try:  
 i\_N = int(data[2])  
 except ValueError:  
 print("Скорее всего вы ввели не число.")  
 exit()  
 if i\_N <= 0:  
 print("Число членов прогрессии должно быть положительным.")  
 exit()  
  
 summ = getNSum(i\_N, f\_Elem, f\_Step)  
  
 out = open('output.txt', 'w')  
  
 out.write("Результат: " + str(summ))  
  
 print("Результаты записаны в файл.")  
  
 out.close()

Модуль до коду:

def getNSum(n, elem, step):  
 if n == 1:  
 return elem  
 else:  
 return elem+(n-1)\*step+getNSum(n-1, elem, step)