МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

ІНСТИТУТ КОМП'ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра ІСМ

******

Звіт

до лабораторної роботи №1

На тему “Введення в Python”

З дисципліни “Спеціалізовані мови програмування”

*Виконав:*

*ст. гр. ІТ-31*

*Даніїл Верхутін*

*Прийняв:*

*Щербак С. С.*

*Львів - 2023*

**Мета роботи:** створення консольної програми-калькулятора за допомогою основних синтаксичних конструкцій Python, з іншим завданням на заміну тестуванню та валідації.

**Хід роботи**

**Елементи програмного коду:**

lab1/calculation.py

import math

class CalculateException(Exception):

pass

def add(num1, num2):

return num1 + num2

def subtract(num1, num2):

return num1 - num2

def multiply(num1, num2):

return num1 \* num2

def divide(num1, num2):

try:

if num2 == 0:

raise CalculateException("EXCEPTION: DIVISION BY ZERO")

else:

return num1 / num2

except CalculateException as error:

return error

def sqrt(num1, num2):

return math.sqrt(num1)

def power(num1, num2):

return num1 \*\* num2

def remDiv(num1, num2):

return num1 % num2

operators = {

'+': add,

'-': subtract,

'\*': multiply,

'/': divide,

'sqrt': sqrt,

'^': power,

'%': remDiv

}

def calculate(num1, operator, num2=None):

num1 = float(num1)

if num2 is not None:

num2 = float(num2)

return operators.get(operator)(num1, num2)

lab1/history.py

from .calculation import \*

from .input import InputException

from collections import OrderedDict

historyDict = OrderedDict()

def inputMaxHistory():

historyMax = ""

while not historyMax.isdigit():

try:

historyMax = input("Enter the maximum amount of calculate records that you want to save: ")

if not historyMax.isdigit():

raise InputException("EXCEPTION: NOT A DIGIT")

except InputException as error:

print(error)

return historyMax

def addResult(num1, operator, num2):

global historyMax

if len(historyDict) == 0:

historyMax = inputMaxHistory()

if operator != "sqrt":

problem = num1 + operator + num2

result = calculate(num1, operator, num2)

else:

problem = num1 + operator

result = calculate(num1, operator)

historyDict[problem] = result

if len(historyDict) > int(historyMax):

historyDict.popitem(last=False)

def printHistory():

for index, problem in enumerate(historyDict):

index += 1

print(f"{index}. {problem} = {historyDict[problem]}")

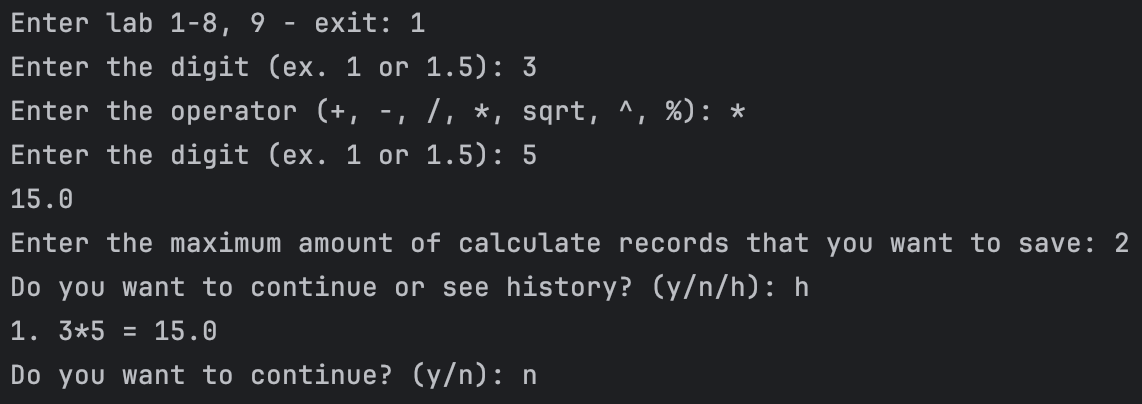
def getResultByIndex(historyIndex):

for index, problem in enumerate(historyDict):

if index == historyIndex:

print("num1 =", historyDict[problem])

return historyDict[problem]

**Результат виконання програми:**

**Висновок:** Під час виконання даної лабораторної роботи було створено консольну програму-калькулятор за допомогою основних синтаксичних конструкцій Python.