МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра інформаційних систем і мереж

Лабораторна робота №2

на тему

“Основи роботи з модулем tkinter”

з дисципліни

“Екстремальне програмування”

Виконала:

студентка групи КН-311

Рибалкіна Марина

Перевірив:

к.т.н.

Щербак С.С.

Львів 2020

**Мета роботи**: ознайомитись з відповідними модулями, створити програмний продукт “Калькулятор” та протестувати його.

**Теоретичний матеріал**:

Пакет tkinter ("Tk interface") - це стандартний інтерфейс Python до інструментарію Tk GUI. Як Tk, так і tkinter доступні на більшості платформ Unix, а також на системах Windows. (Сам Tk не є частиною Python; він підтримується в ActiveState.) Запуск python -m tkinter з командного рядка повинен відкрити вікно, що демонструє простий інтерфейс Tk, повідомляючи про те, що tkinter правильно встановлений у вашій системі, а також показує яка версія Tcl / Tk встановлена, щоб ви могли прочитати документацію Tcl / Tk, специфічну для цієї версії.

**Хід роботи**:

Було написано тести до калькулятора:

import unittest

from tkinter import \*

from calculator import Application

class MyTestCase(unittest.TestCase):

def \_\_init\_\_(self, \*args, \*\*kwargs):

super(MyTestCase, self).\_\_init\_\_(\*args, \*\*kwargs)

self.calculator = Tk()

self.calculator.title("Calculator")

self.calculator.resizable(0, 1)

self.app = Application(self.calculator)

self.app.grid()

def test\_sum(self):

self.app.display.delete(0)

self.app.display.insert(0, "2+2")

self.app.calculateExpression()

self.assertEqual(self.app.display.get(), "4", "2+2 should have been 4")

def test\_sum\_float(self):

self.app.display.delete(0)

self.app.display.insert(0, "2.5+2.1")

self.app.calculateExpression()

self.assertEqual(self.app.display.get(), "4.6", "2.5+2.1 should have been 4.6")

def test\_sub(self):

self.app.display.delete(0)

self.app.display.insert(0, "4-2")

self.app.calculateExpression()

self.assertEqual(self.app.display.get(), "2", "4-2 should have been 2")

def test\_sub\_float(self):

self.app.display.delete(0)

self.app.display.insert(0, "4-2.1")

self.app.calculateExpression()

self.assertEqual(self.app.display.get(), "1.9", "4-2.1 should have been 1.9")

def test\_div(self):

self.app.display.delete(0)

self.app.display.insert(0, "4/4")

self.app.calculateExpression()

self.assertEqual(int(float(self.app.display.get())), 1, "4/4 should have been 1")

def test\_mul(self):

self.app.display.delete(0)

self.app.display.insert(0, "4\*4")

self.app.calculateExpression()

self.assertEqual(self.app.display.get(), "16", "4 \* 4 should have been 16")

def test\_mul\_float(self):

self.app.display.delete(0)

self.app.display.insert(0, "4\*4.5")

self.app.calculateExpression()

self.assertEqual(self.app.display.get(), "18.0", "4 \* 4.5 should have been 18.0")

def test\_percentage(self):

self.app.display.delete(0)

self.app.display.insert(0, "20%")

self.app.calculateExpression()

self.assertEqual(self.app.display.get(), "0.2", "20% should have been 0.2")

def test\_c(self):

self.app.display.insert(0, "20%")

self.app.clearButton.invoke()

self.assertEqual(self.app.display.get(), "0", "After 'C' click text should have been reset to '0'")

def test\_all\_num\_buttons(self):

test\_cases = {

"oneButton" : "1", "twoButton" : "2", "threeButton" : "3",

"fourButton" : "4", "fiveButton" : "5", "sixButton" : "6",

"sevenButton" : "7", "eightButton" : "8", "nineButton" : "9",

"zeroButton" : "0", "plusButton" : "+", "dotButton" : ".",

"minusButton" : "-", "percentageButton" : "%", "divideButton" : "/",

"timesButton" : "\*"

}

for key in test\_cases:

self.app.display.delete(0)

getattr(self.app, key).invoke()

self.assertEqual(self.app.display.get(), test\_cases[key], "After click on " + key + " text should have been" + test\_cases[key])

if \_\_name\_\_ == '\_\_main\_\_':

unittest.main()

Далі було написано програмний продукт “Калькулятор”:

from tkinter import \*

from tkinter import messagebox

calculator = Tk()

calculator.title("Calculator")

calculator.resizable(0, 1)

class Application(Frame):

def \_\_init\_\_(self, master, \*args, \*\*kwargs):

Frame.\_\_init\_\_(self, master, \*args, \*\*kwargs)

self.createWidgets()

def replaceText(self, text):

self.display.delete(0, END)

self.display.insert(0, text)

def appendToDisplay(self, text):

self.entryText = self.display.get()

self.textLength = len(self.entryText)

if self.entryText == "0":

self.replaceText(text)

else:

self.display.insert(self.textLength, text)

def calculateExpression(self): # python's calculate function

self.expression = self.display.get()

self.expression = self.expression.replace("%", "/ 100")

try:

self.result = eval(self.expression)

self.replaceText(self.result)

except:

messagebox.showinfo("ERROR", "Invalid input", icon="warning", parent=calculator)

def clearText(self): # clears imput on pressing C on Calculator

self.replaceText("0")

def createWidgets(self):

self.display = Entry(self, font=("Helvetica", 16), borderwidth=0, relief=RAISED, justify=RIGHT)

self.display.insert(0, "0")

self.display.grid(row=0, column=0, columnspan=5)

# This is the First Row

self.sevenButton = Button(self, font=("Helvetica", 11), text="7", borderwidth=0,

command=lambda: self.appendToDisplay("7"))

self.sevenButton.grid(row=1, column=0)

self.eightButton = Button(self, font=("Helvetica", 11), text="8", borderwidth=0,

command=lambda: self.appendToDisplay("8"))

self.eightButton.grid(row=1, column=1)

self.nineButton = Button(self, font=("Helvetica", 11), text="9", borderwidth=0,

command=lambda: self.appendToDisplay("9"))

self.nineButton.grid(row=1, column=2)

self.timesButton = Button(self, font=("Helvetica", 11), text="\*", borderwidth=0,

command=lambda: self.appendToDisplay("\*"))

self.timesButton.grid(row=1, column=3)

self.clearButton = Button(self, font=("Helvetica", 11), text="C", borderwidth=0,

command=lambda: self.clearText())

self.clearButton.grid(row=1, column=4)

# This is the Second Row

self.fourButton = Button(self, font=("Helvetica", 11), text="4", borderwidth=0,

command=lambda: self.appendToDisplay("4"))

self.fourButton.grid(row=2, column=0)

self.fiveButton = Button(self, font=("Helvetica", 11), text="5", borderwidth=0,

command=lambda: self.appendToDisplay("5"))

self.fiveButton.grid(row=2, column=1)

self.sixButton = Button(self, font=("Helvetica", 11), text="6", borderwidth=0,

command=lambda: self.appendToDisplay("6"))

self.sixButton.grid(row=2, column=2)

self.divideButton = Button(self, font=("Helvetica", 11), text="/", borderwidth=0,

command=lambda: self.appendToDisplay("/"))

self.divideButton.grid(row=2, column=3)

self.percentageButton = Button(self, font=("Helvetica", 11), text="%", borderwidth=0,

command=lambda: self.appendToDisplay("%"))

self.percentageButton.grid(row=2, column=4)

# This is the Third Row

self.oneButton = Button(self, font=("Helvetica", 11), text="1", borderwidth=0,

command=lambda: self.appendToDisplay("1"))

self.oneButton.grid(row=3, column=0)

self.twoButton = Button(self, font=("Helvetica", 11), text="2", borderwidth=0,

command=lambda: self.appendToDisplay("2"))

self.twoButton.grid(row=3, column=1)

self.threeButton = Button(self, font=("Helvetica", 11), text="3", borderwidth=0,

command=lambda: self.appendToDisplay("3"))

self.threeButton.grid(row=3, column=2)

self.minusButton = Button(self, font=("Helvetica", 11), text="-", borderwidth=0,

command=lambda: self.appendToDisplay("-"))

self.minusButton.grid(row=3, column=3)

self.equalsButton = Button(self, font=("Helvetica", 11), text="=", borderwidth=0,

command=lambda: self.calculateExpression())

self.equalsButton.grid(row=3, column=4, rowspan=2)

self.zeroButton = Button(self, font=("Helvetica", 11), text="0", borderwidth=0,

command=lambda: self.appendToDisplay("0"))

self.zeroButton.grid(row=4, column=0, columnspan=2)

self.dotButton = Button(self, font=("Helvetica", 11), text=".", borderwidth=0,

command=lambda: self.appendToDisplay("."))

self.dotButton.grid(row=4, column=2)

self.plusButton = Button(self, font=("Helvetica", 11), text="+", borderwidth=0,

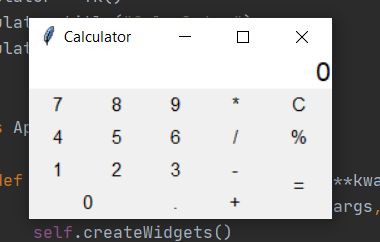
command=lambda: self.appendToDisplay("+"))

self.plusButton.grid(row=4, column=3)

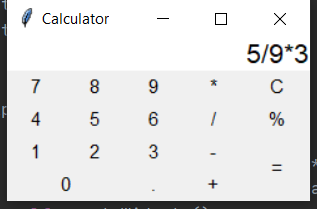
#app = Application(calculator).grid()

#calculator.mainloop()

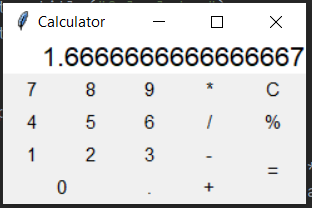
Після цього було отримано наступний результат:

**

*Рис. 1. Стартове вікно калькулятору*

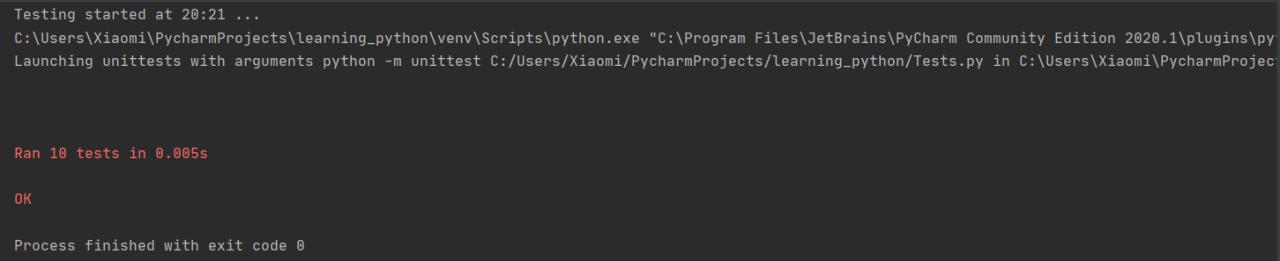
**

*Рис. 2. Введений вираз у калькулятор*

**

*Рис. 3. Розрахований результат введеного виразу*

Після цього були запущені вищезгадані тести, які показали наступний результат:

**

*Рис. 4. Результат тестування*

**Висновок**: у ході даної роботи було ознайомлено з відповідними модулями, створено програмний продукт “Калькулятор” та протестовано його.