**№5**

**Лабораторная работа**

**«Использование элементов Кнопка Надпись Поле ввода»**

**Вариант 0**

**Цель работы:** получить практические навыки по использованию графических элементов Кнопка, Надпись, Поле ввода.

**Краткая теория:** Практическое использование графических элементов рассмотрим на примере создания простого калькулятора с графическим интерфейсом на Python 3.

**Индивидуальное задание:** создание функций подсчета sin, cos, tan.

**ХОД РАБОТЫ**

Для реализации графического интерфейса воспользуемся стандартным пакетом Tkinter, входящим в состав Python 3 (если установлен Python, то дополнительно не надо ничего устанавливать).

**Реализация на языке программирования Python**

Для реализации проекта, понадобятся модули: tkinter и math, а также sys. Поэтому первым делом импортируем их.

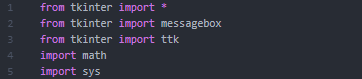


Рисунок 1 – импорт модулей

Далее создаем окно программы



Рисунок 2 – создание окна программы

Прописываем калькулятору логику и способность считать.

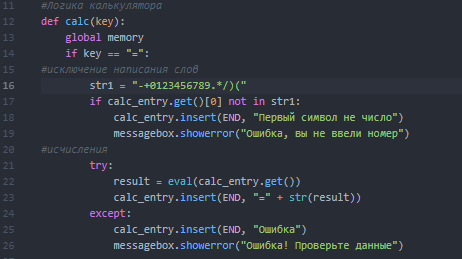


Рисунок 3 – логика калькулятора

Этой строкой кода разрешается пользователю вводить только символы -+0123456789.\*/)(, а остальные исключаются, запрещаются вводить.

В этом блоке кода используется функция *eval* — это, компилятор внутри компилятора. Она будет производить расчеты в программе. По сути, производится обработка функции, что сработает при нажатии на кнопку "=".

Создаем функцию очистки поля ввода. Она будет срабатывать при нажатии на кнопку «C».



Рисунок 4 – функция очистки поля

Создаем функцию изменения минуса на плюс.

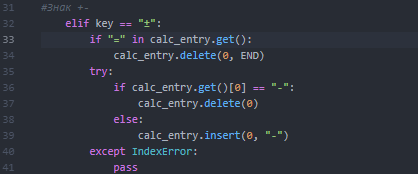


Рисунок 5 – код функции смены знака

Следующая функция — число pi. При нажатии на кнопку «П» программа выведет 3.14159265359, то есть число Pi. Для этого потребуется библиотека math.



Рисунок 6 – функция «Pi»

При нажатии на кнопку «Exit» окно Tkinter будет уничтожено и процесс остановлен. В этой функции требуется библиотека sys.



Рисунок 7 - функция выхода из программы.

Нужно ввести число, которое нужно возвести в степень. Далее программа выводит \*\*. В Python этот символ означает возведение в степень 2\*\*6 (возведение 2 в степень 6).



Рисунок 8 - функция возведения в степень.

Далее с помощью модуля math создаем функции sin, cos, tan

|  |  |  |  |
| --- | --- | --- | --- |
| Работу выполнил: | Светашев В.А. | Дата: | Подпись: |
| Работу принял: | Малибашев А.В. |  |  |

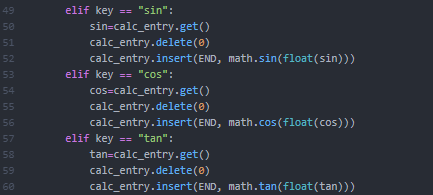


Рисунок 9 – код функций sin, cos, tan

Следующие две функции — «скобки» ) и (. При нажатии на кнопку ")" выводится ), аналогично поступаем со второй функцией.

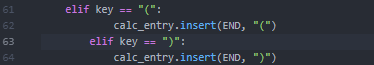


Рисунок 10 – функции «скобки»

Функция, которая отвечает за очистку поля ввода при нажатии на кнопку "=".

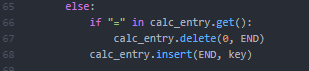


Рисунок 11 – код функции очистки поля ввода

Создаем список с именами будущих кнопок калькулятора. Алгоритмы работы которых были прописаны выше.

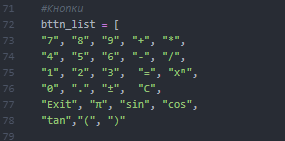


Рисунок 12 – список имен кнопок калькулятора

Создаем сами кнопки калькулятора

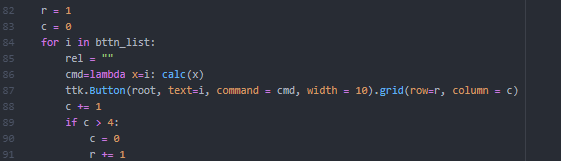


Рисунок 13 – код стилизации кнопок

В каждом калькуляторе есть, поле ввода, в которое пользователь вводит нужные данные для программы. Это могут быть цифры, функции и математические операции. Их можно вводить, как с клавиатуры, так и при нажатии на кнопку в калькуляторе. В Python Tkinter поле ввода называется *Entry.*



Рисунок 14 – код поля ввода

И последняя строка кода — это «закрытие» окна tkinter.



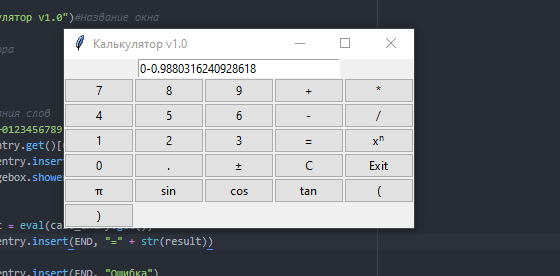
Рисунок 15 – закрытие программы

**ЛИСТИНГ ПРОГРАММЫ**

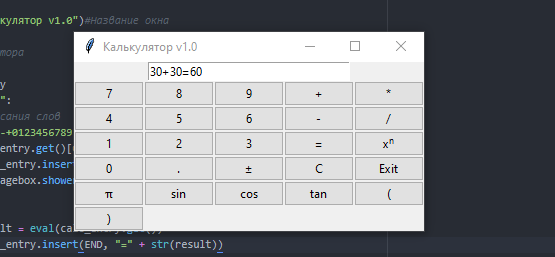
|  |
| --- |
| from tkinter import \*  from tkinter import messagebox  from tkinter import ttk  import math  import sys  #Создание окна программы  root = Tk()  root.title("Калькулятор v1.0")#Название окна  #Логика калькулятора  def calc(key):  global memory  if key == "=":  #исключение написания слов  str1 = "-+0123456789.\*/)("  if calc\_entry.get()[0] not in str1:  calc\_entry.insert(END, "Первый символ не число")  messagebox.showerror("Ошибка, вы не ввели номер")  #исчисления  try:  result = eval(calc\_entry.get())  calc\_entry.insert(END, "=" + str(result))  except:  calc\_entry.insert(END, "Ошибка")  messagebox.showerror("Ошибка! Проверьте данные")  #Очистить поле  elif key == "C":  calc\_entry.delete(0, END)  #Знак +-  elif key == "±":  if "=" in calc\_entry.get():  calc\_entry.delete(0, END)  try:  if calc\_entry.get()[0] == "-":  calc\_entry.delete(0)  else:  calc\_entry.insert(0, "-")  except IndexError:  pass  elif key == "π":  calc\_entry.insert(END, math.pi)  elif key == "Exit":  root.after(1,root.destroy)  sys.exit  elif key == "xⁿ":  calc\_entry.insert(END, "\*\*")  elif key == "sin":  sin=calc\_entry.get()  calc\_entry.delete(0)  calc\_entry.insert(END, math.sin(float(sin)))  elif key == "cos":  cos=calc\_entry.get()  calc\_entry.delete(0)  calc\_entry.insert(END, math.cos(float(cos)))  elif key == "tan":  tan=calc\_entry.get()  calc\_entry.delete(0)  calc\_entry.insert(END, math.tan(float(tan)))  elif key == "(":  calc\_entry.insert(END, "(")  elif key == ")":  calc\_entry.insert(END, ")")  else:  if "=" in calc\_entry.get():  calc\_entry.delete(0, END)  calc\_entry.insert(END, key)  #Кнопки  bttn\_list = [  "7", "8", "9", "+", "\*",  "4", "5", "6", "-", "/",  "1", "2", "3", "=", "xⁿ",  "0", ".", "±", "C",  "Exit", "π", "sin", "cos",  "tan","(", ")"  ]  r = 1  c = 0  for i in bttn\_list:  rel = ""  cmd=lambda x=i: calc(x)  ttk.Button(root, text=i, command = cmd, width = 10).grid(row=r, column = c)  c += 1  if c > 4:  c = 0  r += 1  calc\_entry = Entry(root, width = 33)  calc\_entry.grid(row=0, column=0, columnspan=5)  root.mainloop() |

**КОНТРОЛЬНЫЙ ВАРИАНТ**

Задаем получаем ответ:



Складываем



|  |  |  |  |
| --- | --- | --- | --- |
| Работу выполнил: |  | Дата: | Подпись: |
| Работу принял: |  |  |  |