**Муниципальное бюджетное общеобразовательное учреждение “Школа №133”**

Гайд по созданию калькулятора на языке программирования Python с использованием библиотеки Tkinter

Выполнил: Приказчиков Я.Э., учащийся 9 “В” класса

Руководитель: Данилович С.В., учитель информатики

Нижний Новгород

2024

**Оглавление**

**Стр. 1.** Титульный лист.

**Стр. 2.** Оглавление.

**Стр. 3.** Введение.

**Стр. 4.** Описание проекта.

**Стр. 5.** Что такое библиотека в языке программирования Python и почему я выбрал именно Tkinter?

**Стр. 7.** Написание кода на языке программирования Python при помощи библиотеки Tkinter для калькулятора.

**Стр. 13.** Компиляция получившегося кода при помощи компилятора Pyinstaller.

**Стр. 14.** Публикацияполученногообучающего пособия на платформу GitHub.

**Стр. 15.** Вывод.

**Стр. 16.** Список литературы.

**Введение**

В наше время не многие способны делать сложные математические вычисления без многофункционального программного калькулятора. Программы такого рода часто используют различные специалисты, основной работой которых является вычисление каких-либо числовых выражений. Не для кого не секрет, что это довольно сложный процесс, который необходимо максимально упростить, чтобы увеличить как КПД работников подобных направлений, так и уменьшить дневную мозговую нагрузку. Чтобы всего этого достичь, необходимо создать обучающее пособие для разработчика, который будет писать код для обсуждаемого нами калькулятора. Именно его, по итогам проведённой работы, мы и сможем представить будущим реализаторам.

**Описание проекта**

**Актуальность проекта:** Специалистам, которые непосредственно связаны с вычислительными процессами, необходима программа многофункционального калькулятора. Она сможет упростить работу и тем самым увеличит производительность труда и его продолжительность.

**Проблема проекта:** Разработчикам необходимо создать программу для сложных математических вычислений, но для этого необходимо обучающее пособие, которое позволит осуществить поставленную задачу.

**Цель проекта:** Создать обучающее пособие (гайд) по созданию стандартного калькулятора для разработчиков, которые в последующем его доработаю под необходимые специалистам задачи на языке программирования Python с использованием библиотеки Tkinter.

**Задачи проекта:**

**1.** Узнать о том, что такое библиотека и почему я выбрал именно Tkinter.

**2.** Написать код на языке программирования Python при помощи библиотеки Tkinter для калькулятора.

**3.** Произвести компиляцию данного кода для его последующего запуска в виде файла с расширением “.exe”

**4.** Поместить получившееся в результате проведённой работы обучающее пособие на общедоступную платформу GitHub.

**Что такое библиотека в языке программирования Python и почему я выбрал именно Tkinter?**

Библиотеки — это наборы функций и инструментов, написанные другими людьми ранее, которые помогают в написании кода и расширяют возможности языка программирования Python. Это значит, что написанную часть кода можно использовать в разных проектах. Существует множество разных библиотек на «Пайтон», которые предназначены для решения определенных задач. Установленные библиотеки Python — это ваши помощники, которые ускоряют работу над проектом. Например, библиотека Matplotlib содержит функции для создания графиков и диаграмм, а библиотека Pandas используется для работы с табличными данными.

Кроме того, в Python уже встроены некоторые библиотеки, которые можно использовать для создания проектов. Они не требуют установки и доступны сразу. Некоторые из них:

Random — используется для генерации случайных чисел.

Math — содержит математические функции, такие как sin, cos, sqrt и др.

Tkinter — используется для создания графических интерфейсов.

Продолжая данную тему, я могу сказать, что библиотека Tkinter была выбрана мною не случайно. В первую очередь, она подходила мне по её изначально заложенному предназначению - создание графических интерфейсов, что и является направлением моей работы. Помимо этого, данная библиотека содержит необходимый функционал и возможности для дальнейшего усовершенствования моего итогового продукта. Некоторые из них:

Функциональность — большое количество интерактивных элементов и атрибутов для создания полноценного работающего интерфейса.

Кроссплатформенность — возможность работать на разных операционных системах.

Широкая поддержка — данная библиотека собрала целое сообщество интересующихся ею пользователей.

Простота — для написания кода на Tkinter нужно всего лишь импортировать её в код, она изначально установлена в системе Visual Studio Code.

**Написание кода на языке программирования Python при помощи библиотеки Tkinter для калькулятора**

Прежде всего, работа должна начинаться с подготовки области работы и среды её выполнения.

Для этого необходимо открыть программу для написания кодов различного уровня сложности Visual Studio Code и создания в появившейся области файла с расширением .py — на этом подготовка к написанию кода завершена.

Первые строчки нашего кода начинаются с команды на импортирование необходимой нам библиотеки Tkinter:

from tkinter import \*

После этого нам необходимо сразу описать объект и его поведение используя метод class и определить работу функции def следующими командами:

class Main(Frame):

def \_\_init\_\_(self, root):

super(Main, self).\_\_init\_\_(root)

self.build()

Классы (class) помогают кратко и понятно описывать объекты и их поведение независимо от основного кода программы.

Такой метод делает код более читаемым и понятным, а также помогает избежать многих ошибок и багов.

Функция def, используемая в данном коде, создаёт объект и присваивает ему определённое имя.

После описания объекта нам необходимо настроить окно будущего калькулятора и все необходимые параметры для его корректной работы.

Для этого необходимо использовать следующие команды:

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

root = Tk()

root["bg"] = "#FFF"

root.geometry("485x550")

root.title("Калькулятор")

root.resizable(False, False)

app = Main(root)

app.pack()

root.mainloop()

С самой первой строки мы указываем условие с помощью if, которое будет применять все параметры, если классом объекта является main.

После этого нам необходимо создать объект, используя команду root=Tk() и назначить для него параметры. Некоторыми из них являются:

root["bg"] = "#FFF" — устанавливает цвет заднего фона окна приложения если указать атрибут bg.

root.geometry("485x550") — устанавливает размеры окна в пикселях.

root.title("Калькулятор") — устанавливает название окна нащего приложения.

root.resizable(False, False) — устанавливает запрет на изменение размеров окна пользователем если использовать атрибут false.

Данная часть кода помещается в конец и используется для того, чтобы после какой-либо корректировки кода, находящегося между нею и описанием поведения объекта в начале, можно было сразу запустить его и посмотреть на изменения и возможные ошибки.

Следующим шагом является создание поля, на котором будут выводиться как сами вычисления, так и их результаты.

Для этого необходимо написать следующий код:

def build(self):

self.formula = "0"

self.lbl = Label(text=self.formula, font=("SF Pro Text", 30, "bold"),

bg="#FFF", foreground="#000")

self.lbl.place(x=11, y=50)

Здесь мы настраиваем визуальную составляющую текста. При желании можно поменять сам шрифт, его размер, цвет и цвет фона, который находится за ним при помощи команды foreground.

После этого нам необходимо создать физические кнопки, которые будет нажимать пользователь, используя метод build, который мы уже назначили ранее.

Для этого необходимо написать следующие команды:

btns = [ "C", "DEL", "\*", "=",

"1", "2", "3", "/",

"4", "5", "6", "+",

"7", "8", "9", "-",

"(", "0", ")", "X^2"]

Здесь мы использовали атрибут btns и вручную указали все необходимые нам кнопки с их примерным расположением.

Не стоит забывать о визуальной составляющей данных кнопок, поэтому следующим шагом является настройка их внешнего вида.

Для этого необходимо воспользоваться следующими командами:

x = 10

y = 140

for bt in btns:

com = lambda x=bt: self.logicalc(x)

Button(text=bt, background="#FFF",

font=("SF Pro Text", 30, "bold"),

command=com).place(x=x, y=y,

width=115,

height=80)

x += 117

if x > 400:

x = 10

y += 81

Данный процесс аналогичен изменению визуальной составляющей поля ввода и результатов вычислений, с который мы проводили работу ранее. Мы всё также можем изменить шрифт, его размер и цвет, но также здесь мы можем настроить расположение самих кнопок по осям x и y в окне приложения.

Завершающим шагом написания данного кода является настройка создание логики для стабильной работы калькулятора и выполнения им всех заявленных функций.

Для этого нам необходима следующая кодировка:

def logicalc(self, operation):

if operation == "C":

self.formula = ""

elif operation == "DEL":

self.formula = self.formula[0:-1]

elif operation == "X^2":

self.formula = str((eval(self.formula))\*\*2)

elif operation == "=":

self.formula = str(eval(self.formula))

else:

if self.formula == "0":

self.formula = ""

self.formula += operation

self.update()

def update(self):

if self.formula == "":

self.formula = "0"

self.lbl.configure(text=self.formula)

**Компиляция получившегося кода при помощи компилятора Pyinstaller**

Мы завершили написание кода и теперь смело можем делать его компиляцию, чтобы наш код мог открываться на всевозможных операционных системах с расширением .exe.

Для этого необходимо воспользоваться компилятором Pyinstaller и начать с ним работу в командной строке PowerShell. В ней изначально будет указан путь к файлу, который не относится к нашему проекту, поэтому необходимо написать следующий код:

C:\Users\yaros>cd C:\Users\yaros\Desktop\Калькулятор

Здесь мы указываем путь через команду cd к корневой папке с нашим калькулятором.

После этого необходимо установить данный компилятор и запустить его для нашего калькулятора следующими командами:

C:\Users\yaros\Desktop\Калькулятор>pip install -U pyinstaller

C:\Users\yaros\Desktop\Калькулятор>pyinstaller calculator.py

Теперь мы можем лицезреть полноценный код, адаптированный для запуска на всех операционных системах и полностью готовый к использованию его по прямому назначению. Безусловно, мы не стримились сделать продвинутый вариант калькулятора, поэтому и сделали базовый вариант программы, дабы другие разработчики усовершенствовали его для узконаправленных специалистов.

**Публикация полученного обучающего пособия на платформу GitHub**

Завершающим шагом в данном проекте является публикация данного обучающего пособия на площадке GitHub, поскольку есть вероятность, что разработчики, которые увидят статью с ним заинтересуются и начнут работу по усовершенствованию этого приложения. Кроме разработчиков, данная статья, безусловно, будет полезна начинающим кодерам, которые хотят начать свой путь с чего-то простого.

Статья находится здесь: https://github.com/FlowCreatorr

**Вывод:**

Таким образом, мы провели работу над написанием кода программы калькулятора в Visual Studio Code на языке программирования Python в связке с библиотекой Tkinter и компилятором Pyinstaller и создали полноценный гайд, который подробно описывает процесс создания с некоторыми пояснениями и будет полезен как для начинающих кодеров, так и для продвинутых разработчиков, которые в перспективе могут основательно взяться за проект и усовершенствовать его, дабы его функционал удовлетворял все потребности узконаправленных специалистов.

**Список литературы:**

1. <https://habr.com/ru/articles/481074/>

2. <https://pyinstaller.org/en/stable/index.html>

3. <https://habr.com/ru/sandbox/115348/>