Реферат

Тема: Описати створення графічного інтерфейсу за допомогою Tkinter

Тулкіт tkinter доступний у вигляді окремого вбудованого модуля, який містить усі необхідні графічні компоненти – кнопки, текстові поля тощо.

Базовим моментом у побудові графічних програм є створення вікна. Потім у вікно додаються всі інші компоненти графічного інтерфейсу.

from tkinter import \*

root = Tk()

root.title("Графическая программа на Python")

root.geometry("400x300")

root.mainloop()

Для створення графічного вікна застосовується конструктор Tk(), визначений у модулі tkinter. за допомогою методу title() можна встановити заголовок вікна. За допомогою методу geometry() – розмір вікна. Для встановлення розміру метод geometry() передається рядок у форматі "Ширина x Висота". Для відображення вікна треба викликати в нього метод mainloop(), який запускає цикл обробки подій для взаємодії з користувачем. За замовчуванням вікно позиціонується у лівий верхній кут екрана. Але ми можемо змінити його положення, передавши потрібні значення метод geometry():

from tkinter import \*

root = Tk()

root.title("Графическая программа на Python")

root.geometry("400x300+300+250")

root.mainloop()

Тепер рядок у методі geometry має такий формат: "Ширина x Висота + координата X + координата Y". Тобто при запуску вікно перебуватиме на 300 пікселів праворуч і 250 пікселів вниз від верхнього лівого кута екрана.

Для створення кнопки використається конструктор Button(). У цьому архітекторі за допомогою параметра text можна встановити текст кнопки. Щоб зробити елемент видимим, він викликається метод pack().

Button (master, options)

Параметр master надає посилання на батьківський контейнер. Якби створювалося кілька вікон, тоді ми могли б передати в конструктор Button посилання на потрібне вікно.

Другий параметр options є набір насправді набір параметрів, які ми можемо встановити на їх ім'я. Деякі з них:

justify: встановлює вирівнювання тексту. Значення LEFT вирівнює текст по лівому краю, CENTER – по центру, RIGHT – по правому краю

padx: відступ від меж кнопки до її тексту праворуч та ліворуч

pady: відступ від меж кнопки до її тексту зверху та знизу

bg/background: фоновий колір кнопки

fg/foreground: колір тексту кнопки

Іноді може знадобитися змінити атрибути кнопки, наприклад текст. Однак у тієї ж кнопки немає такого методу зміни тексту. І виникає питання, як його можна змінити?

Для зміни тексту ми можемо використати проміжний компонент StringVar. Цей компонент дозволяє створити прив'язку до рядка. Він має два методи:

get(): повертає рядок із StringVar

set(str): встановлює рядок у StringVar

Для зв'язку об'єкта StringVar з текстом візуального елемента цього елемента в конструкторі треба встановити параметр textvariable.

Крім компонента StringVar, нам доступно ще ряд подібних компонентів для інших типів даних:

IntVar

BooleanVar

DoubleVar

Але є й інший спосіб, який допоможе, якщо нам треба змінювати не тільки текст, але й інші параметри кнопки або іншого компонента. Це спосіб полягає у виклику елемента методу config(), в якому і встановлюється потрібний параметр.

from tkinter import \*

clicks = 0

def click\_button():

global clicks

clicks += 1

btn.config(text="Clicks {}".format(clicks))

root = Tk()

root.title("GUI на Python")

root.geometry("300x250")

btn = Button(text="Clicks 0", background="#555", foreground="#ccc",

padx="20", pady="8", font="16", command=click\_button)

btn.pack()

root.mainloop()

Метод place() дозволяє точніше налаштувати параметри позиціонування. Він приймає такі параметри:

height і width: встановлюють відповідно висоту та ширину елемента в пікселях

x та y: встановлюють зміщення елемента по горизонталі та вертикалі в пікселях відповідно щодо верхнього лівого кута контейнера.

bordermode: визначає формат межі елемента. Може приймати значення INSIDE (за замовчуванням) та OUTSIDE

Слід зазначити, що з використанні методу place() не треба використовувати метод pack(), щоб зробити елемент видимим.

Метод grid застосовує інший підхід до позиціонування елементів, ніж метод place, дозволяючи помістити елемент у певну комірку умовної сітки або гриду.

Метод grid застосовує такі параметри:

column: номер стовпця, відлік починається з нуля

row: номер рядка, відлік починається з нуля

columnspan: скільки стовпців повинен займати елемент

rowspan: скільки рядків повинен займати елемент

ipadx та ipady: відступи по горизонталі та вертикалі відповідно від меж елемента до його тексту

padx та pady: відступи по горизонталі та вертикалі відповідно від меж осередку гриду до меж елемента

sticky: вирівнювання елемента в осередку, якщо осередок більший за елемент. Може приймати значення n, e, s, w, ne, nw, se, sw, які вказують відповідний напрямок вирівнювання

from tkinter import \*

root = Tk()

root.title("GUI на Python")

root.geometry("300x250")

for r in range(3):

for c in range(3):

btn = Button(text="{}-{}".format(r,c))

btn.grid(row=r, column=c, ipadx=10, ipady=6, padx=10, pady=10)

root.mainloop()

Текстові позначки у Python представлені елементом Label. Цей елемент дозволяє виводити статичний текст без можливості редагування.

Для створення елемента Label застосовується конструктор, який приймає два параметри:

Label(master, options)

Елемент Entry є полем для введення тексту. Конструктор Entry приймає такі параметри:

Entry(master, options)

master – посилання на батьківське вікно

Елемент Entry має низку методів. Основні з них:

insert(index, str): вставляє в текстове поле рядок за певним індексом

get(): повертає введений текстове поле текст

delete(first, last=None): видаляє символ з індексу first. Якщо вказано параметр last, видалення проводиться до індексу last. Щоб видалити до кінця, в якості другого параметра можна використовувати значення END.

Елемент Checkbutton є прапорцем, який може перебувати в двох станах: зазначеному і невідзначеному.

Відмінністю Checkbutton є можливість прив'язки до компонента IntVar через параметр variable. У зазначеному стані прив'язаний компонент IntVar має значення 1, а невідзначеному - 0. У результаті через IntVar ми можемо отримувати значення, вказане користувачем.

Елемент Radiobutton являє собою перемикач, який може перебувати у двох станах: зазначеному або невідзначеному. Але на відміну від Checkbutton перемикачі можуть створювати групу, з якої миттєво ми можемо вибрати тільки один перемикач.

from tkinter import \*

languages = [("Python", 1), ("JavaScript", 2), ("C#", 3), ("Java", 4)]

def select():

l = language.get()

if l == 1:

sel.config(text="Выбран Python")

elif l == 2:

sel.config(text="Выбран JavaScript")

elif l == 3:

sel.config(text="Выбран C#")

elif l == 4:

sel.config(text="Выбран Java")

root = Tk()

root.title("GUI на Python")

root.geometry("300x280")

header = Label(text="Выберите курс", padx=15, pady=10)

header.grid(row=0, column=0, sticky=W)

language = IntVar()

row = 1

for txt, val in languages:

Radiobutton(text=txt, value=val, variable=language, padx=15, pady=10, command=select)\

.grid(row=row, sticky=W)

row += 1

sel = Label(padx=15, pady=10)

sel.grid(row=row, sticky=W)

root.mainloop()

Для створення ієрархічного меню tkinter і Python застосовується віджет Menu.

Меню може містити багато елементів, причому ці елементи можуть представляти меню і містити інші елементи. Залежно від того, який тип елементів ми хочемо додати в меню, буде відрізнятися метод, який використовується для їхнього додавання. Зокрема, нам доступні такі методи:

add\_command(options): додає елемент меню через параметр options

add\_cascade(options): додає елемент меню, який, у свою чергу, може представляти підменю

add\_separator(): додає лінію-розмежувач

add\_radiobutton(options): додає в меню перемикач

add\_checkbutton(options): додає в меню прапорець

Відмінною рисою елементів меню є здатність реагувати на натискання користувача. Для цього у кожного елемента меню можна встановити параметр command, який встановлює посилання на функцію, що виконується при натисканні.

1. Що таке tkinter?
2. Як підключити tkinter в скрипт?
3. Чи потрібно імпортувати щось для підключення tkinter?
4. Який клас застосовується для створення вікна?
5. Який метод дає можливість вказати назву вікна?
6. Чи існує метод для зміни назви додатків?
7. Який метод дає можливість задати розмір вікна?
8. Який метод запускає цикл обробки подій для взаємодії з користувачем?
9. Який метод є альтернативою pack?
10. Що означає параметр padx?
11. Як розмістити елемент UI у потрібному нам вікні?
12. Що означає параметр pady?
13. Що означає параметр fg?
14. Що означає параметр bg?
15. Який метод елемента розміщує його у вікні?
16. Який компонент створює прив’язку до тексту кнопки?
17. Яким методом можна встановити значення для StringVar?
18. Яким методом можна отримати значення для StringVar?
19. Як називається змінна, в яку необхідно передати елемент з прив’язаним значенням для тексту кнопки?
20. Що означає justify=CENTER для кнопки?
21. Який метод дає можливість більш точно розміщувати елементи у вікні: pack чи place?
22. Чи має Button параметр master?
23. Який метод вставляє в Entry рядок?
24. Який метод видаляє рядок з Entry?
25. Чи правда, що меню може містити лише 2 елемента?
26. Що означає параметр x метода place?
27. Що означає параметр y метода place?
28. Що таке Entry?
29. Чи має Label параметр master?
30. Чи має Entry параметр master?
31. Який клас використовується для текстових позначень на UI?
32. Чи може Radiobutton утворювати групу з якої миттєво ми можемо вибрати тільки один перемикач?
33. Для чого потрібен метод config?
34. Чи має Button параметр command?
35. Скільки елементів групи для Radiobutton можуть мати увімкнений стан?
36. Який клас дає можливість створити меню?
37. Чи треба використовувати метод pack після place?
38. Чи можна створити прив’язку до значення поля Entry?
39. Яких значень може набувати Checkbutton?

Джерела:

https://metanit.com/python/tutorial