



Загальний порядок створення графічного інтерфейсу

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки і санітарно-гігієнічних норм

Мова Python забезпечує роботу з різними графічними бібліотеками. За допомогою таких бібліотек можна створювати програми з розвинутим графічним інтерфейсом користувача. Достатньо потужні графічні можливості має модуль `tkinter`, який і розглядатиметься далі.

Модуль `tkinter` входить до стандартного дистрибутива Python. Основним призначенням цього модуля є створення графічних інтерфейсів для програм, а завдяки наявності об'єкта полотно (`Canvas`) його можна застосовувати й для малювання. За допомогою модуля `tkinter` можна створювати також графічні функції. Але для побудови графіків потужніші можливості в модуля `matplotlib` (який тут не розглядається).

Бібліотека `tkinter` має стандартний набір об'єктів (їх ще називають віджетами, або компонентами), із яких, власне, і створюється графічний інтерфейс. Прикладами віджетів є кнопка, смуга прокручування, прапорець, перемикач та інші.

Кожному об'єкту в бібліотеці відповідає свій клас. Наприклад, об'єкту кнопка відповідає клас `Button`, а об'єкту перемикач — клас `Radiobutton`. Базовим класом бібліотеки є клас `Tk`, за допомогою якого створюється головне вікно.

Усі віджети в програмі створюються за однаковою принципом — шляхом виклику конструктора відповідного класу, який має таку загальну структуру:

<ім'я змінної> = <назва класу> [(параметри)]

Наприклад, створити кнопку можна за допомогою інструкції: `but1 = Button()`. Першим параметром у цьому конструкторі вказується ім'я батьківського віджета, у якому буде розміщено цей об'єкт. Якщо цей параметр не вка-

зано, то об'єкт буде розміщено в головному вікні. Далі можуть міститися інші параметри, які визначають його конфігурацію, наприклад, назву — `text`, колір — `bg`, шрифт — `font` та інші.

Щоб скористатися можливостями графічного інтерфейсу, слід передбачити розміщення у вікні необхідних графічних об'єктів, а також ті дії, які мають виконуватися в процесі взаємодії користувача з цими об'єктами. Безумовно, це збільшує обсяг програмного коду й ускладнює його структуру, але суттєво спрощує користувачеві взаємодію з програмою. А інколи без використання графічних об'єктів просто не можна



обійтися.

Наприклад, неможливо уявити роботу банкомата або платіжного терміналу без висвітлення на екрані різних компонентів, за допомогою яких практично будь-який користувач може взаємодіяти з програмами, які ці при-

строї обслуговують. Використовуючи графічні об'єкти, користувачі навіть можуть і не знати про сутність програми, із якою він взаємодіє. Таким прикладом є програма управління сучасною пральною машиною.

Для створення графічного інтерфейсу користувача необхідно знати перелік об'єктів бібліотеки tkinter, їх призначення й можливості, порядок розміщення об'єктів у вікні та події, які можуть виникати під час взаємодії з ними. Саме в такій послідовності й розроблено цей підрозділ.

Опишемо алгоритм створення графічного інтерфейсу мовою Python.

1. Імпортуйте модуль tkinter і створіть головне вікно.

```
from tkinter import *  
назва вікна=Tk()
```

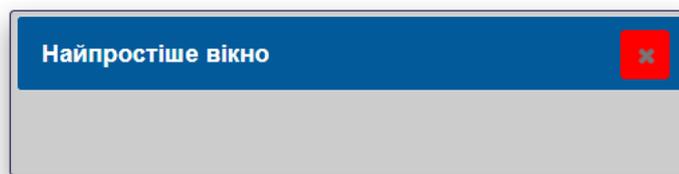
Змінна, яку пов'язують з об'єктом цього вікна, зазвичай має ім'я root. Найпростіша інструкція для створення головного вікна має таку структуру: root = Tk()

Створення порожньої форми

```
1 from tkinter import *  
2 root = Tk() # створення вікна  
3 root.title("Найпростіше вікно") # заголовок вікна  
4 root.geometry("400x100") # розмір вікна  
5 root.mainloop() # вікно залишається на екрані поки його не закриємо
```

Всі наступні команди потрібно розміщувати перед командою root.mainloop()

Результат:



2. Створіть у програмі віджети й визначте їх властивості.

Пригадаємо, що кожен віджет має свій клас. Будь-який віджет повинен бути пов'язаний із відповідним іменем змінної. Наприклад, кнопку Button будемо пов'язувати зі змінною but за допомогою інструкції: but=Button(root). Тут аргумент root означає, що кнопка буде розміщена в головному вікні. За замовчуванням кнопку також буде розміщено в головному вікні. Але якщо кнопку необхідно розмістити в іншому вікні, то слід вказати його ім'я.

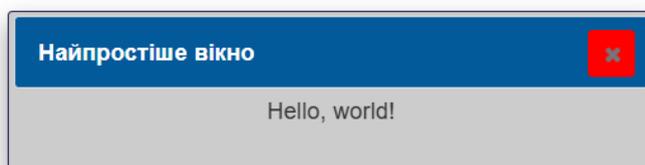


Будь-який віджет має властивості. Наприклад, кнопка може мати розмір, колір, напис. Конкретні значення властивостей віджета вказуються як значення відповідних параметрів після назви вікна (наприклад, після `root`) або встановлюються після їх створення. Наприклад, для кнопки, якій присвоєно ім'я `but`, можна встановити властивість напис за допомогою інструкції: `but=Button (root, text="Виконати")`.

Напис на формі

```
label = Label (root, text="Hello, world!") # вказуємо, де створюється напис з іменем label, і його текстовий вміст
label.pack() # розміщення на формі
```

Результат:



Кнопка

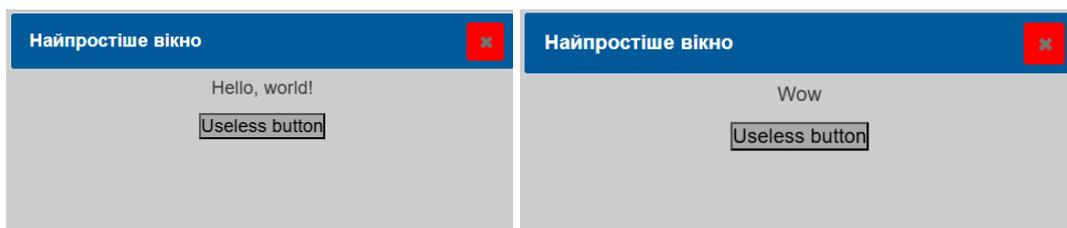
```
кнопка=Button(root, text="Useless button", command=reply) # вказуємо, де створюється кнопка кнопка,
# її текстовий вміст та назва функції, яка викликається при натисненні цієї кнопки
кнопка.pack(side=TOP, padx=5, pady=5) # розміщення кнопки на формі: вирівнювання згори, відступи
# по 5 пікселів по x та y
```

Функція, яка викликається цією кнопкою змінює текст напису на "Wow". Це описано так:

```
def reply():
    label.configure(text="Wow")
```

Зверніть увагу, що визначення функції `reply` має розміщуватись до опису кнопки, бажано на початку програми.

Результат:

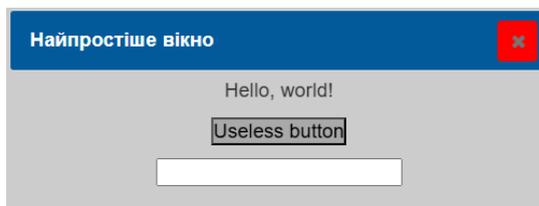


Поле введення



```
textBox = Entry (root) # вказуємо, де створюється поле введення інформації  
textBox.pack(padx=5, pady=5) # розміщення на формі з відступами по 5 пікселів по горизонталі та вертикалі  
імя=textBox.get() # інформація з поля введення записується у змінну імя
```

Результат:



3. Визначте події та порядок їх опрацювання.

Для кожного компонента можна передбачити деякі події. Наприклад, кнопка може бути натиснута або змінено її розмір. У програмі необхідно передбачити опрацювання кожної події, наприклад, які дії слід виконати під час натискання кнопки. Опрацювання подій часто реалізується у вигляді функцій, які викликаються під час виникнення відповідної події.

Функції у цьому випадку мають таку загальну структуру:

```
def <назва функції> (<назва події>):  
    <тіло функції>
```

Функція зазвичай створюється на початку програмного коду. Щоб можна було викликати функцію опрацювання події, необхідно зв'язати цю функцію із самою подією. Зв'язування реалізується за допомогою методу bind. Наприклад, зв'язати подію натиснення кнопки Button_1 із функцією func_1 можна за допомогою інструкції: but.bind("<Button_1>").

Прив'язуємо до кнопки подію (можна послідовно прив'язувати кілька подій до різних об'єктів).

```
назва_кнопки.bind("<подія>", назва_функції)
```

4. Вказуємо місце розташування віджетів у нашому вікні.

Найпоширеніші менеджери – pack(), grid() (сітка) та place() (абсолютне позиціонування).

У головному вікні розміщуються віджети, вони можуть бути навіть у строго визначених координатах вікна. Щоб розмістити віджети без визначення координат, використайте метод pack(). Наприклад, розмістити кнопку у вікні можна за допомогою інструкції: but.pack() або за допомогою методу place():
назва_кнопки.place(x="число", y="число")

5. Вказуємо про завершення роботи з вікном.

Головне вікно відображається на екрані за допомогою методу mainloop(). Виконайте інструкцію root.mainloop(), яка має бути останньою в програмі.

```
назва_вікна.mainloop()
```



Онлайн середовище: <https://ed-info.github.io/epython/>

Задача 1. На рис. зображено код програми, за допомогою якої відображається головне вікно з двома кнопками. Одна з них має напис *Множення*, а друга — *Додавання*.

Якщо натиснути кнопку *Множення*, то викличеться функція `func2`, введуться й помножаться два числа.

Якщо натиснути кнопку *Додавання*, то викличеться функція `func1`, введуться й додадуться два інших числа.

```
1  from tkinter import*
2  def func1(eve):
3      a=int(input("Увести a: "))
4      b=int(input("Увести b: "))
5      print("a+b = ", a+b)
6  def func2(fff):
7      c=int(input("Увести c: "))
8      d=int(input("Увести d: "))
9      print("c*d = ", c*d)
10
11  root=Tk()
12  but1=Button(root)
13  but1["text"] = "Додавання"
14  but1.bind("<Button>", func1)
15  but2=Button (root)
16  but2["text"]="Множення"
17  but2.bind("<Button>", func2)
18  but2.pack()
19  but1.pack()
20  root.mainloop()
```

Після запуску програми на виконання на екрані з'явиться вікно з двома кнопками (рис. 2) та вікно, у яке необхідно ввести дані для опрацювання. Вікно для



введення даних можна перемістити в інше місце екрана так, щоб ці вікна не перетиналися.

Натиснемо спочатку, наприклад, кнопку Додавання й уведемо два числа — результат додавання чисел відобразиться на екрані. Потім натиснемо кнопку Множення й уведемо два числа — з'явиться результат множення. Ще раз натиснемо кнопку Додавання й уведемо два числа. Можливий результат описаних взаємодій із програмою наведено на рис. 3.

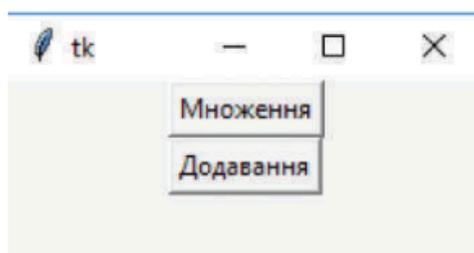


Рис. 2. Вікно з двома кнопками

```
Увести a: 8
Увести b: 45
a+b = 53
Увести c: 7
Увести d: 9
c*d = 63
Увести a: 33
Увести b: 9
a+b = 42
```

Рис. 3. Результат взаємодій із програмою

Створювати графічний інтерфейс користувача можна й на основі об'єктно-орієнтованого підходу, обов'язковою складовою якого є клас. Такий підхід у деяких випадках є ефективнішим за метод процедурного програмування.

Онлайн середовище: <https://ed-info.github.io/epython/>

Задача 2. На рис. нижче зображено код програми, яка реалізує ті самі функції, що й програма, яку зображено у задачі 1. Але вона побудована з використанням класу.

Принципова різниця між цими програмами полягає лише в тому, що числа в другій програмі не вводяться за допомогою клавіатури.



```
1 from tkinter import*
2 class K19_04:
3     def __init__(self, a, b, c, d):
4         self.s=a+b
5         self.p=c*d
6         self.but1=Button(root)
7         self.but1["text"]="Додавання"
8         self.but1.pack()
9         self.but2=Button(root)
10        self.but2["text"]="Множення"
11        self.but2.pack()
12        self.but1.bind("<Button-1>", self.func1)
13        self.but2.bind("<Button-1>", self.func2)
14        def func1(self, rkt):
15            print("a+b = ", self.s)
16        def func2 (self, rkt):
17            print("c*d = ", self.p)
18 root = Tk()
19 obj=K19_04(3, 9, 6, 7)
20 root.mainloop()
```

```
c*d = 42
a+b = 12
c*d = 42
c*d = 42
a+b = 12
```

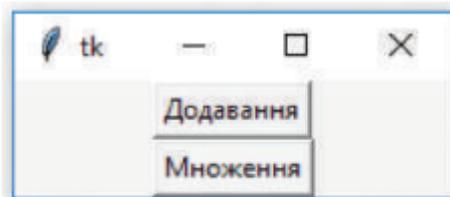


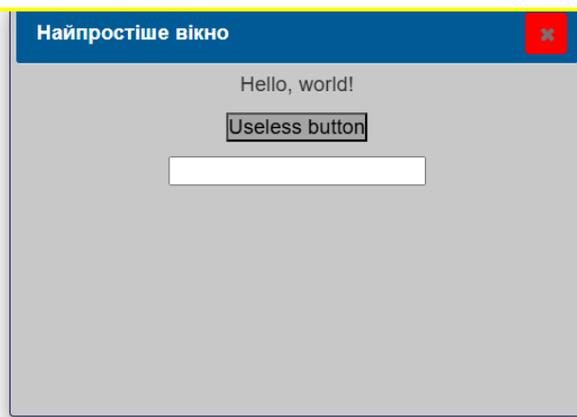
Рис. 6. Виконання програми на основі ООП будови графічного інтерфейсу

Задача 3. Створити проект, у якому є 1 кнопка, 1 поле введення інформації та 1 напис. Ім'я користувача вводиться у текстове поле. При клацанні кнопки даний текст заноситься у напис на формі.



Внесіть зміни у наведений код, щоб написи відповідали тим, що наведені на зображенні.

```
1 def reply():
2     imya=textBox.get()
3     label.configure(text=imya)
4
5 from tkinter import *
6 root = Tk()
7 root.title("Найпростіше вікно")
8 root.geometry("400x300")
9
10 label = Label (root, text="Hello, world!")
11 label.pack()
12
13 кнопка=Button(root, text="Useless button", command=reply)
14 кнопка.pack(side=TOP, padx=5, pady=5)
15
16 textBox = Entry (root)
17 textBox.pack(padx=5, pady=5)
18
19 root.mainloop()
20
```





Запитання для перевірки знань

- 1 Для чого призначений модуль tkinter?
- 2 Які основні об'єкти містить модуль tkinter?
- 3 За допомогою якої інструкції віджети розміщуються у вікні?
- 4 Яку загальну структуру має функція опрацювання події?
- 5 Які переваги й дефекти має графічний інтерфейс користувача?
- 6 Поясніть загальний порядок створення графічного інтерфейсу користувача.
- 7 Поясніть особливості створення графічного інтерфейсу користувача на основі об'єктно-орієнтованого підходу.



Завдання для самостійного виконання

- 1 Розробіть програму, за допомогою якої у головному вікні розміщується кнопка та здійснюється введення з клавіатури трьох чисел. Після натиснення кнопки обчислюється середнє арифметичне уведених чисел.
- 2 Розробіть програму, за допомогою якої у головному вікні розміщується кнопка, після натиснення якої генеруються два цілі випадкові числа, множаться й результат виводиться на екран.
- 3 Розробіть програму, за допомогою якої у головному вікні розміщуються дві кнопки з назвами Увести та Ділення. Після натиснення першої кнопки два числа вводяться з клавіатури, а після натиснення другої перше число ділиться на друге.
- 4 Розробіть програму, за допомогою якої у головному вікні розміщуються три кнопки. Після натиснення першої кнопки вводиться ціле число, після натиснення другої — генерується випадкове число, а після натиснення третьої — вони складаються.
- 5 Розробіть програму на основі об'єктно-орієнтованого підходу, за допомогою якої у головному вікні розміщуються кнопки Менше і Більше. Після натиснення першої кнопки з клавіатури вводяться два числа й повідомляється, яке з них менше. Після натиснення другої кнопки з клавіатури вводяться два інших числа й повідомляється, яке з них більше.