

## Лабораторна робота №13

**Тема:** Створення графічного інтерфейсу. Tkinter.

**Мета роботи:** навчитися створювати простіший графічний інтерфейс за допомогою вбудованої бібліотеки Tkinter.

### 1. Теоретичні відомості

*Tkinter* – це фреймворк для Python, призначений для роботи з бібліотеками Tk. Бібліотека Tk містить компоненти графічного інтерфейса користувача, написані на мові програмування Tcl.

*Графічний інтерфейс користувача (GUI)* — це тип інтерфейсу, який дає змогу користувачам взаємодіяти з електронними пристроями через графічні зображення та візуальні вказівки, на відміну від текстових інтерфейсів, заснованих на використанні тексту, текстовому наборі команд та текстовій навігації. Виконання дій у GUI— це безпосередня маніпуляція з графічними елементами.

Tkinter імпортується так само, як і інші бібліотеки.

```
import tkinter as tk
```

Далі, щоб створити програму з GUI, потрібно виконати такі дії:

- 1) Створити головне вікно.
- 2) Створити віджети (елементи) цього вікна.
- 3) Приєднати до елементів властивості (функції).
- 4) Вибрати події, на які буде реагувати програма.
- 5) Розташувати елементи на вікні.
- 6) Запустити цикл обробки подій.

Для створення вікна потрібно створити об'єкт класу Tk.

```
win = tk.Tk()
```

- `.geometry()` – зміна розміру вікна.

Його параметр – це строка типу ‘ширина x висота + відступ зверху + відступ зліва’.

```
win = tk.Tk()  
win.geometry('300x300+200+100')
```

- `.title()` – зміна заголовку вікна.
- `.overridedirect()` – прибирає строку назви вікна.
- `.destroy()` – закриває вікно, може використовуватися в функціях.

- `.configure()` – параметри вікна.
- `.resizable()` – дозволяє або забороняє розширення вікна.

### Створення віджетів (елементів)

Створення елементів відбувається після створення головного вікна. Всі вони створюються методом вікна з прив'язкою до змінної.

```
lab = tk.Label(width='50', text='Введіть текст') #Створюємо напис
ent = tk.Entry(width=50) #Поле для вводу
but = tk.Button(text='Переробити') #Кнопку
rez = tk.Label(width='50', text='') #Та пусте поле для напису
```

Віджети зачасто мають такі ж параметри, як і вікно.

Поле Entry має кілька власних методів:

- `.get()` – отримати введений текст. У випадку, якщо аргументи не приведені, то метод зчитує весь текст. Якщо приведені 2 числових аргументи, то метод зчитує зріз з введеної строки.
- `.delete()` – видаляє введений текст. Аргументи аналогічні до методу `get`.
- `.insert()` – вставляє строку в поле Entry. Перший аргумент – позиція (індекс) вставлення строки (якщо текст повинен бути в кінці – END), другий аргумент – строка, що вставляється.

Для додавання функціоналу віджетів треба додати функцію та прив'язати її до події.

Для редагування якоїсь властивості елемента вказується сам елемент, назва властивості у квадратних дужках та цьому виразу присвоюється значення.

```
def vyvid(event):
    text = ent.get()
    rez['text'] = str(text)
```

Властивість елемента обов'язково повинна бути строкового типу.

Для того, щоб указати до чого прив'язана функція та при якій події вона виконується, використовують метод `bind()`. У всередині дужок ми пишемо подію та функцію, яка виконується. Наприклад:

```
but.bind('<Button-1>', vyvid)
```

У данному випадку ми прив'язуємо виконання функції до натискання лівою кнопкою миші на кнопку.

Основні події в Python:

<Button-1> – натискання лівої кнопки миші.

<Button-2> – настикання середньої кнопки миші (коліщатко).

<Button-3> – натискання правої кнопки миші.

<Enter> – введення курсору на зону елемента.

<Leave> – виведення курсору з зони елемента.

<Key> – натискання на будь-яку клавішу.

<Return> – натискання на Enter.

<BackSpace> – натискання пробілу.

<Shift\_L> – натискання будь-якого Shift.

<Control\_L> – натискання будь-якого Ctrl.

Також у «скобках» < > можна написати будь-яку клавішу – на цю клавішу і буде прив’язана функція. Звертайте увагу на регістр: <r> та <R> – це різні клавіші. Найбільш ефективно це прив’язувати до полей вводу тексту.

Щоб віджет з’явився у вікні, потрібно його отуди помістити. Найпростішим методом для цього є метод `pack()`. Він належить усім віджетам. При його використанні без додаткових параметрів, розміщення віджетів буде послідовним у стовпець на верхній частині вікна.

```
lab.pack()
```

Останній крок – зациклення вікна. Цей крок обов’язковий, так як без нього вікно буде закриватися одразу ж після першого кадру. Для цього використовується метод `mainloop()`. Цей метод доступний лише вікнам.

```
win.mainloop()
```

Він запускає цикл, завдяки якому вікно не закривається.

## 2. Хід роботи

- 1) Виконати завдання на мові Python згідно з варіантом. Написати програму та перевірити її виконання.
- 2) Скласти звіт, який має мати:
  - найменування та мету роботи
  - завдання за варіантом
  - код програми
  - результат виконання програми
  - стислі відповіді на контрольні питання

### 3. Завдання

Створіть програму з графічним інтерфейсом користувача, яка буде виконувати задачу згідно з вашим варіантом. Колір та розміри обрати за власним бажанням.

*Варіант 1*

Матиме 2 поля вводу, одну кнопку та поле виводу. Нехай введені строки конкатенуються, якщо це строки, та складаються, якщо їх можливо перевести в числа. Результат вивести.

*Варіант 2*

Виконує роль калькулятора з двома полями вводу. Якщо введена строка, вивести помилку. Калькулятор повинен мати додавання, віднімання, множення та ділення.

*Варіант 3*

Отримує строку та сортирує слова за алфавітом (функція `sort()`). Сторовану строку вивести в поле для виводу тексту.

*Варіант 4*

Отримує строку та виводить в поле для виводу 3 випадкових слова з неї.

*Варіант 5*

Виводить тип даних, який було введено до поля для вводу (спочатку все отримується як строка, потрібно перевести).

### 4. Контрольні питання

- 1) Що таке GUI?
- 2) Основні дії при створенні програми з графічним інтерфейсом.
- 3) Перечислити методи та функції, які обов'язково використовуються для створення програм з графічним інтерфейсом
- 4) Перечислити 5 (на Вашу думку) найвживаніших подій.