

Поняття фреймворку. Tkinter.

Фреймворк, відмінності від бібліотек

Фреймворк – це бібліотека, яка повністю визначає структуру програми; програмне забезпечення, визначаюче структуру програмної системи; програмне забезпечення, яке значно полегшує роботу програміста та дозволяє об'єднати різні компоненти проекту. Фреймворк також може використовувати велику кількість бібліотек для власної роботи.

Відмінність від бібліотеки та, що при імпорті бібліотеки встановлюється набір додаткових функцій для користування в файлі програми. При імпорті фреймворку змінюється архітектура програми. В деяких випадках створюється набір папок, необхідних для проекту.

Прикладами крупних фреймворків на мові Python є:

- Django – фреймворк для ефективного створення сайтів та веб-додатків. До його сильних сторін належать такі можливості:
 - Аутентифікація
 - Маршрутизація URL-адрес
 - Робота з базами даних: PostgreSQL, MySQL, SQLite, Oracle
 - Підтримка веб-серверів та інш.
- CherryPy – мікрофреймворк для розробки додатків для Android. Є частиною фреймворку TurboGears. Має готові інструменти для аутентифікації, кешування, кодування, статичного контенту.
- TurboGears – фреймворк для створення веб-додатків, які працюють з даними. Вважається альтернативою Django.
- Pyramid – фреймворк, який має великий набір функцій. Є каркасом для як малих, так і великих додатків.
- Bottle – фреймворк для роботи над малими та середніми проектами. При цьому код фреймворка складається лише з одного файлу.
- Tkinter – стандартний фреймворк для створення графічного інтерфейсу користувача. Може бути встановлений разом з інтерпретатором мови Python, тому часто не потребує попереднього встановлення. Фреймворк реалізований на мові Tcl, але має адаптацію для роботи з Python.

Tkinter

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()
```

Створення елементів відбувається після створення головного вікна. Всі вони є об'єктами. Від класу залежить тип елемента, а від переданих при виклику параметрів – зовнішній вигляд та функціонал віджета.

Для створення однострочного поля з текстом використовується клас Label. Його параметрами можуть бути:

- Назва вікна, у яке буде поміщений віджет. У разі, якщо вікно не вказано, віджет буде поміщений у головне вікно. Цей параметр завжди передається на першому місці.
- Width – довжина строки (у символах).
- Text – строка всередині віджета.
- Font – шрифт та розмір тексту
- Bg – колір фону віджета
- Fg – колір напису
- Justify – положення тексту в віджеті

Також є багато інших параметрів, які регулюють більш тонке налаштування.

```
lab = tk.Label(win, width='15', bg='#0bb1bd', text='Приклад тексту', justify='center')
```

За допомогою функцій строкове поле може отримувати нові або змінювати старі поля.

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.

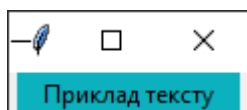
```
lab.pack()
```

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()
```

Як результат ми маємо такий код і результат виконання:

```
1 import tkinter as tk
2
3 win = tk.Tk()
4 lab = tk.Label(win, width='15', bg='#0bb1bd',
5 | | | | text='Приклад тексту', justify='center')
6
7 lab.pack()
8
9 win.mainloop()
```



Контрольні питання:

- 1) Чим фреймворк відрізняється від звичайної бібліотеки?
- 2) Як додати фреймворк до програми?
- 3) Які основні кроки до створення простішого графічного інтерфейсу користувача за допомогою фреймворка Tkinter?