

Урок 21. Поняття об'єкта, його властивості та методи

Вивчення нового матеріалу

Слайд № 1

Ви вже знайомі з графічним режимом роботи в середовищі Python з використанням модуля **turtle**. Але є і модулі зі значно ширшими можливостями, які дають змогу не лише малювати зображення, але й створювати справжні Windows-програми. Зокрема, з цією метою використовують модуль **tkinter**.

Для підключення модуля **tkinter** використовують таку команду:

```
from tkinter import *
```



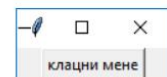
Розпочнімо знайомство з модулем **tkinter**.

Слайд № 2

Особливістю Windows-програм є те, що вони працюють у спеціальних **вікнах** і взаємодіють із користувачем через **елементи керування** — кнопки, текстові поля, прапорці тощо.

Створимо найпростішу програму, що виводить у вікні кнопку.

```
from tkinter import *  
button1=Button(text="Клацніть мене")  
button1.pack()
```

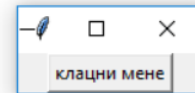


Алекс покаже, як буде працювати ця програма.

Слайд № 3

Запустимо програму, натиснувши F5.

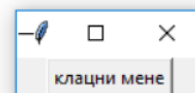
```
from tkinter import*  
button1=Button(text="клацни мене")  
button1.pack()
```



Після запуску програми з'являється вікно з кнопкою. Клацніть її.

Слайд № 4

```
from tkinter import*  
button1=Button(text="клацни мене")  
button1.pack()
```



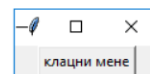
Нічого не відбувається. Створений код лише розташовує кнопку у вікні.

Вправа № 1



Вправа 1. У новому файлі введіть програмний код та запустіть його на виконання. У результаті у вікні буде відображено кнопку.

```
from tkinter import *  
button1=Button(text="Клацніть мене")  
button1.pack()
```

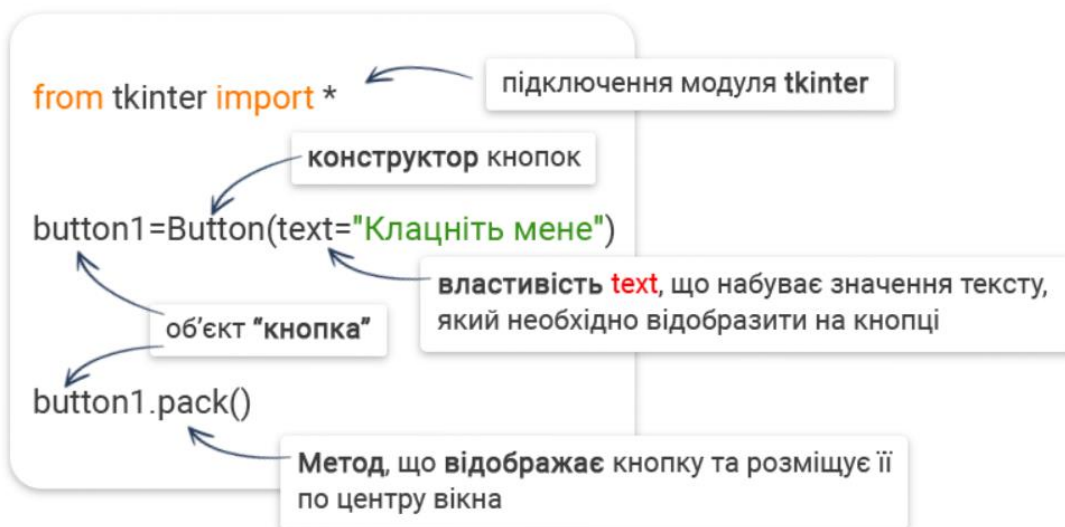


Виконайте вправу в середовищі Python.

Вивчення нового матеріалу

Слайд
№ 5

Проаналізуємо команди програмного коду.

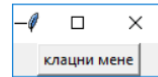


Для об'єкта "кнопка" було задано значення **властивості** та застосовано **метод**.

Слайд
№ 6

У нашому прикладі об'єктом є кнопка.

```
from tkinter import *  
button1=Button(text="Клацніть мене")  
button1.pack()
```



Об'єкт у програмуванні, так само, як і об'єкт у навколишньому світі, — це щось таке, що можна розглядати цілісно, окремо від інших об'єктів.



Вікно програми — це теж об'єкт, тільки він створюється автоматично.

Слайд
№ 7

Пригадаємо програмний код ще раз:

```
from tkinter import *
```

конструктор кнопок

```
button1=Button(text="Клацніть мене")
```

властивість кнопки

об'єкт "кнопка"

```
button1.pack()
```



Зауважте, що **button1** — це назва об'єкта, **Button** — назва конструктора.

Слайд
№ 8

Крім властивостей, об'єкти мають **методи** — міні-програми, що пов'язані з об'єктами.

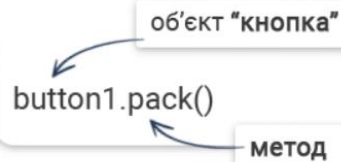
У нашому коді методом є **pack()**, що відображує кнопку.

```
from tkinter import *  
button1=Button(text="Клацніть мене")
```

об'єкт "кнопка"

button1.pack()

метод



Запам'ятайте, що таке **метод**.

Слайд
№ 9

Отже, об'єкти мають **властивості** та **методи**.

Для присвоєння значень властивостям використовують такий синтаксис:

об'єкт=конструктор(властивість=значення)

Для виклику (виконання) **методу об'єкта** використовують такий синтаксис:

об'єкт.метод()



Синтаксис — це правила запису тих чи інших конструкцій.

Отже, об'єкти мають **властивості** та **методи**.

Для присвоєння значень властивостям використовують такий синтаксис:

об'єкт=конструктор(властивість=значення)

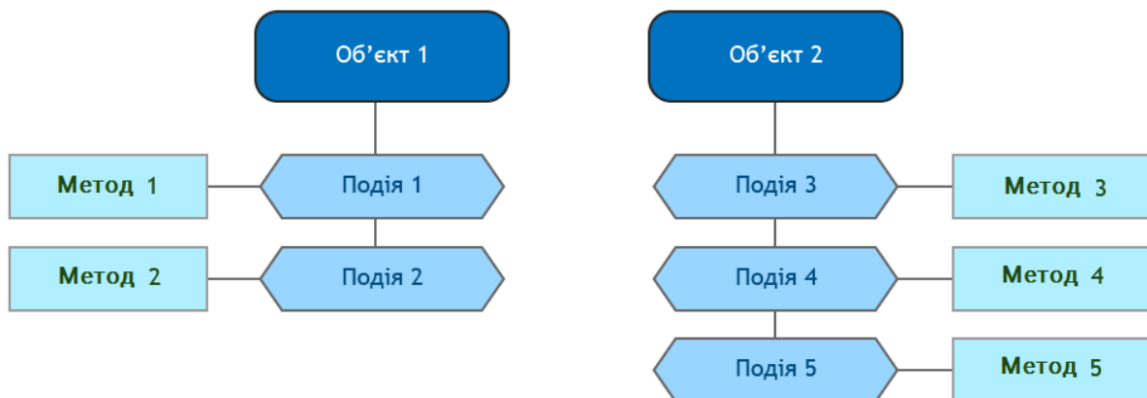
Для виклику (виконання) методу об'єкта використовують такий синтаксис:

об'єкт.метод()



Запам'ятайте синтаксис присвоєння значень властивостям та виконання методів об'єкта.

Об'єкти також можуть реагувати на події, що стаються в програмі.

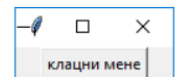


Уся об'єктно-орієнтована програма — це набір об'єктів, пов'язаних із ними подій та методів, що опрацьовують ці події.

Завдання № 8

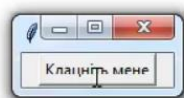
Наша програма зараз не виконує ніяких дій. Як ви думаєте, що потрібно зробити, щоб програма реагувала на натискання кнопки?

- ☐ Задати для кнопки інший текст
- ☐ Записати на початку програми команду на кшталт "Якщо кнопку натиснуто, виконати певну дію"
- ☒ Запрограмувати подію натискання кнопки



Слайд № 12

Запрограмуємо, наприклад, подію натискання кнопки. У результаті настання цієї події виводитиметься повідомлення "Привіт!"
Для цього ми маємо створити окрему **функцію** (підпрограму), а потім **прив'язати** її до події.



```
>>>  
== RESTART: C:/Users/Ihor/AppData  
>>> Привіт!
```



Функція — це майже те саме, що метод, тільки прив'язується вона до подій, а не до об'єктів.

Слайд № 13

Функція — це фрагмент коду, що виконує те чи інше завдання.

У найпростішому випадку **функція** має такий вигляд:

```
def ім'я функції():  
    тіло функції
```

Ім'я функції користувач вигадує самостійно.

Тіло функції — це команди, які виконуватимуться в разі виклику функції.



«Викликати функцію» означає вказати, що її потрібно виконати в разі настання певної події чи в певному місці програми.

Слайд № 14

Створимо функцію виведення тексту "Привіт!" та додамо до конструктора кнопки **button1** параметр, що прив'язуватиме цю функцію до події натискання кнопки.

```
from tkinter import *
```

ім'я функції

```
def button_clicked():  
    print("Привіт!")
```

```
button1=Button(text="Клацніть мене", command=button_clicked)  
button1.pack()
```

ім'я функції

параметр
виклику функції

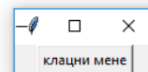


Параметр **command** прив'язує функцію до **стандартної** події. Для кнопки такою подією є натискання, для прапорця – встановлення тощо.

Слайд № 15

Запустимо програму, натиснувши клавішу **F5**.

```
Python 3.6.5 Shell  
File Edit Shell Debug Options Window Help  
Py 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64  
bit (AMD64)] on win32  
Type "copyright", "credits" or "license()" for more information.  
>>>  
===== RESTART: C:\Users\ольга\Documents\проект_1.py =====  
>>>
```



Після запуску програми з'являється вікно з кнопкою. Натисніть її.

Слайд № 16

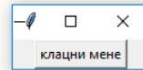
File Edit Shell Debug Options Window Help

Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>

===== RESTART: C:\Users\ольга\Documents\проект_1.py =====
=====

>>> Привіт!

з'явилося повідомлення



З'явилося повідомлення "Привіт!". Натисніть кнопку ще раз.

Слайд № 17

File Edit Shell Debug Options Window Help

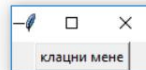
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>

===== RESTART: C:\Users\ольга\Documents\проект_1.py =====
=====

>>> Привіт!

Привіт!

з'явилося ще
повідомлення



Повідомлення з'явилося ще раз. Натисніть кнопку ще раз.

Слайд № 18

File Edit Shell Debug Options Window Help

Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [AMD64] on win32

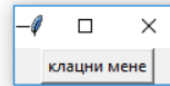
Type "copyright", "credits" or "license()" for more information
>>>

===== RESTART: C:\Users\ольга\Documents\проект_1
=====

>>> Привіт!

Привіт!

Привіт!



І знову з'явилося повідомлення. Воно з'являтиметься стільки разів, скільки ми будемо натискати кнопку.

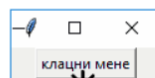
Вправа

Вправа № 2



Вправа 2. Додайте до наявного коду команди, які в разі натискання кнопки виводять повідомлення "Привіт!"

```
from tkinter import *  
def button_clicked():  
    print("Привіт!")  
button1=Button(text="Кляцніть мене", command=button_clicked)  
button1.pack()
```



>>> Привіт!



Змініть програму в середовищі Python.

Слайд № 19

Сьогодні ми зробили перші кроки в **об'єктно-орієнтованому програмуванні** – програмуванні з використанням **об'єктів**.

Скорочено його позначають як **ООП**.

Зараз це найбільш поширена методика програмування.



Методика ООП є панівною з кінця 1980-х років.