Урок №51 8 клас

Події та обробники подій. Вікно повідомлення

10.1 Події та обробка подій

Ми вже навчилися створювати статичні (не змінні) вікна та налаштовувати їх властивості. Але програми не будуються лише на статиці, їм властива і динаміка, тобто певні зміни. А зміни можливі Розглянемо поняття як події.

Подія *— це зміна властивостей об'єкта, взаємодія між ними, утворення нового або знищення існуючого.* Кожна подія містить оцінку часу, що вказує, коли вона відбувається і місця, де вона відбувається.

В роботі із вікнами **на мові Python існують такі події:**

• **Button-1** — клік лівою клавішею миші по будь-якій області об’єкта;

• **Button-3** — клік правою клавішею миші по будь-якій області об’єкта;

• **KeyPress** — натискання будь-якої клавіші на клавіатурі;

• **Motion** — переміщення курсора миші по області об’єкта;

• **Destroy** — закриття вибраного вікна. Звичайно, це не весь перелік, але й цього невеликого списку нам буде досить.

З будь-якою подією, яка може відбутися з формою, можна пов’язати фрагмент програми, який буде виконуватися одразу після настання цієї події. Такий фрагмент програми називають **обробником події**.

У Python обробником подій є **функція**, що являє собою набір команд.

Тобто спочатку потрібно створити функцію, щоб до неї можна було звертатися при виконанні певної події. **Функцію потрібно оголошувати на початку**, після підключення модулів, аби не вийшло непорозумінь з інтерпретатором.

1) СТВОРЕННЯ ФУНКЦІЇ

Функції створюються наступним чином:

```
def назва_функції(аргументи):
    команда
    ...
```

Назву функції ми вигадуємо самостійно, а до команд можуть відноситися створення об’єктів, елементів управління, або змінення властивостей певних об’єктів.

Саме зі зміною властивостей вікна ми і будемо працювати.

Приклад 1 (створення функції)

Давайте створимо функцію, яка б змінювала розміри нашого вікна Window на 500x500 пікселів та встановлювала **зелений** колір фону. Назвемо нашу функцію *change*, на місце аргументу встановлюємо *event* (з англійської - подія), це означатиме, що функція пов'язана з певною подією:

```
from tkinter import *          #підключення модуля
def change(event):          #створюємо функцію change з аргументом event
    Window.geometry("500x500")  #прописуємо команди зміни властивості вікна
    Window["bg"]="green"
    ...
```

2) СТВОРЕННЯ ПОДІЇ

Тепер створимо подію та присвоїмо їй дану функцію. Для цього використовують наступною конструкцією:

назва_об'єкта_до_якого_застосовується_подія.**bind**(**"<*подія*>"**, *назва_функції*)

Зверніть увагу на те, як потрібно записувати подію. **Назву події потрібно вносити в відповідну конструкцію "<>", про це не треба забувати.** Наприклад: "<Button-1>", "<Button-3>" і т.д.

Приклад 1 (продовження: створення події)

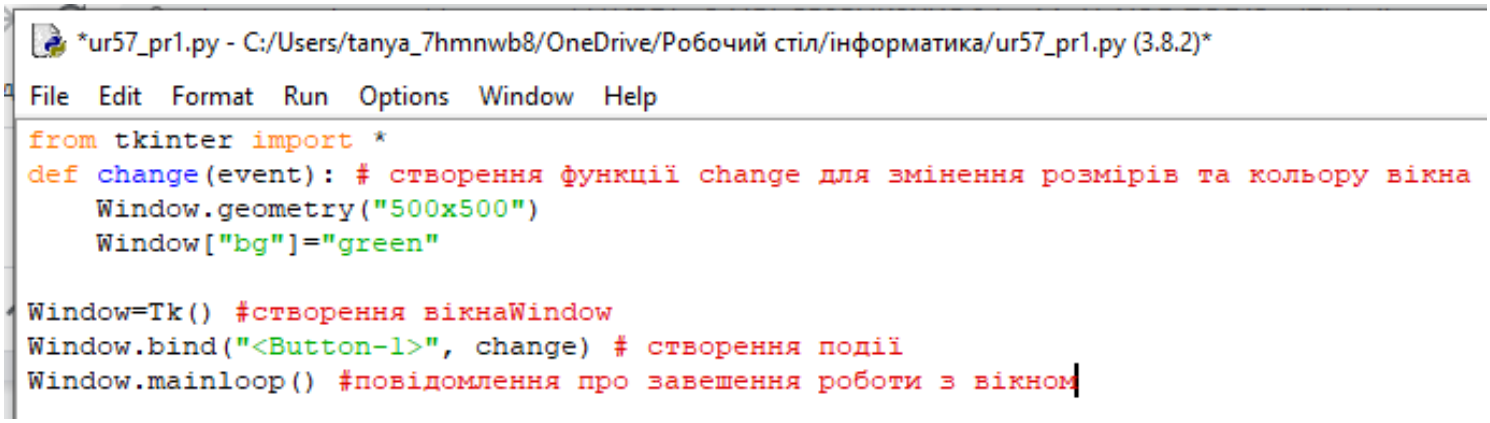
Отже, створимо подію натисканням лівою клавішею миші (**Button-1)**по області вікна Window, при якій буде виконуватися функція *change*.

Тобто маємо:

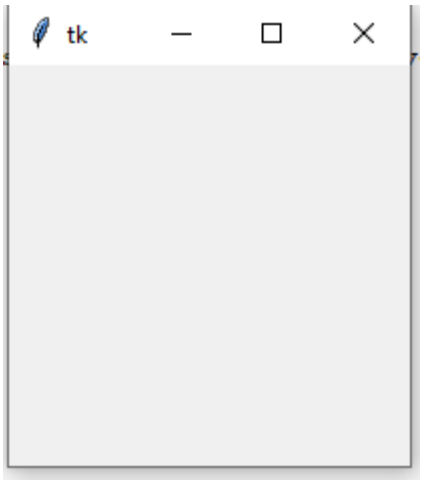
Window.bind("<Button-1>", change)

Приклад 1 (готовий код)

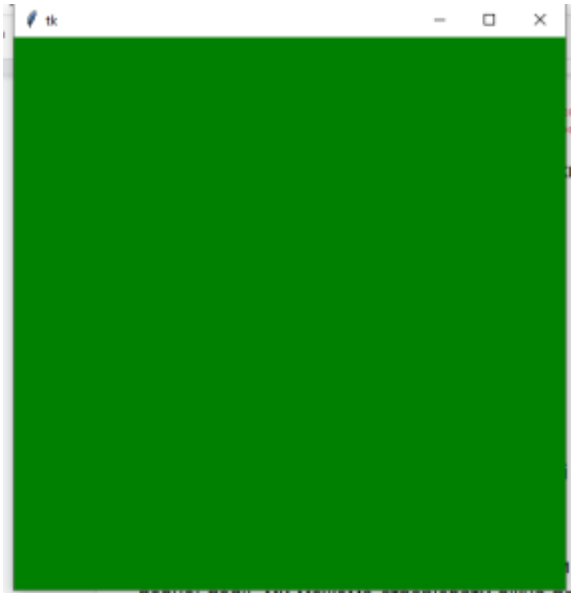
Об'єднаємо ці конструкції в одну програму. Створимо новий файл в IDLE та внесемо в нього наступний код:



При запуску даного коду, маємо звичайне вікно:



Але варто нам клацнути лівою кнопкою миші (саме цю подію ми створили) в області даного вікна як вікно змінить розмір і колір фону:



10.2 Вікно повідомлення

Ми можемо створювати вікна повідомлення, тобто командою функції буде створити вікно з відповідним повідомленням для користувача.

Вікна з повідомленням зручні тим, що вони допомагають донести інформацію користувачеві, при цьому не займаючи місця на головному вікні.

Вікно повідомлення складається тільки із інформації та кнопки “Ок”.

Зверніть увагу! В більш нових версіях потрібно підключати клас **messagebox** окремо, тому якщо ви користуєтеся версією 3.5 і більше, то на початку повинен бути присутній такий рядок коду:

```
from tkinter import messagebox
```

Функцією створення вікна повідомлення є

messagebox.showinfo()

і застосовується вона наступним чином:

messagebox.showinfo(“заголовок вікна”, “зміст повідомлення”)

Приклад 2

Застосуємо для нашого вікна з прикладу 1 функцію створення вікна-повідомлення із заголовком “**Повідомлення**”, та текстом “**Я програмую на Python**”.

От, що у нас вийшло:

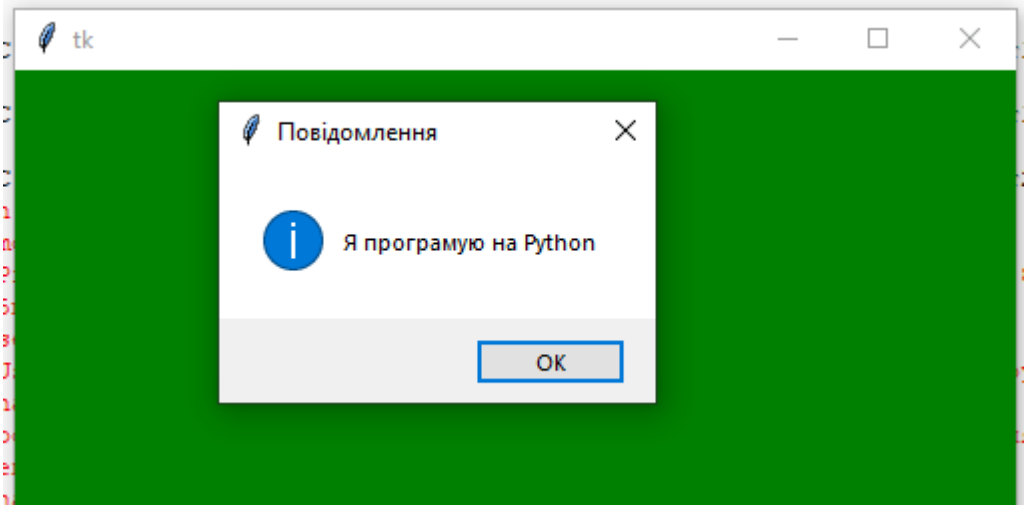
ur57_pr2.py - C:/Users/tanya_7hmnwb8/OneDrive/Робочий стіл/інформатика/ur57_pr2.py (3.8.2)

File Edit Format Run Options Window Help

```
from tkinter import *
from tkinter import messagebox #підключаємо клас messagebox
def change(event): # створення функції change для змінення розмірів та кольору вікна
    Window.geometry("500x500")
    Window["bg"]="green"
    messagebox.showinfo("Повідомлення", "Я програмую на Python") # створення вікна повідомлення

Window=Tk() #створення вікнаWindow
Window.bind("<Button-1>", change) # створення події
Window.mainloop() #повідомлення про завешення роботи з вікном
```

Після запуску даного коду при натисканні лівою кнопкою миші на область вікна маємо:



Контрольні питання

1. Дайте визначення поняттю подія.
2. Наведіть приклади подій в мові програмування Python.
3. Що таке обробник подій?
4. Що є обробником подій на мові програмування Python?
5. Як встановлюється подія в Python?