



**ЗВО:** Національний університет  
«Львівська політехніка»  
**Навчальний рік:** 2024/2025  
**Семестр:** весняний

**Кафедра систем**  
автоматизованого проектування  
**Викладач:**  
Юрій ПАТЕРЕГА

**Навчальна дисципліна:** Розробка  
кросплатформених додатків  
**Лабораторна робота № 1:**  
Розроблення додатку для імітації кодового  
замка

**Група:** ПП-26  
**Студент:** Ігор ЯКІБ'ЮК

### Мета роботи

Розробити додаток на основі бібліотеки tkinter для кодового замка, який містить:

- дисплей для введення коду;
- клавіатуру з цифрами «0»...«9»;
- кнопку «Back» для видалення останнього символу;
- кнопку «Enter» для перевірки правильності коду.

### Інструкція до виконання роботи

1. Вибрати код замка згідно із варіантом:

№	1	2	3	4	5	6	7	8
Код	3011	1378	2049	2597	3293	3257	8810	5273
№	9	10	11	12	13	14	15	16
Код	6943	2074	5240	9286	1349	1225	3903	2156
№	17	18	19	20	21	22	23	24
Код	3520	7864	9607	7858	3593	3442	5949	2962
№	25	26	27	28	29	30	31	32
Код	2533	7841	6782	9971	1912	5417	6322	7759

2. Створити головне вікно:

- Імпортувати бібліотеку tkinter.
- Створити головне вікно (Tk).
- Встановити заголовок вікна та зафіксувати його розмір.

3. Додати дисплей:

- Додати мітку (Label) для відображення введеного коду.
- Встановити білий фон та шрифт Arial, 14 пт.

4. Створити клавіатуру:

- Додати фрейм (Frame) для групування кнопок.
- Додати кнопки (Button) для цифр «0»...«9».
- Додати кнопки (Button) для операцій «Back» та «Enter».

5. Реалізувати оброблення подій:

5.1. Функція введення цифр handle\_digit():

- Додавання символів до текстової змінної code.
- Відображення введеного коду на дисплеї.
- Обмеження довжини введення до 4 символів.

5.2. Функція видалення останнього символу handle\_back():

- Видалення останнього введеного символу.
- Оновлення дисплея.

5.3. Функція перевірки коду handle\_enter():

- Порівняння введеного коду зі збереженим паролем.
- Виведення відповідного повідомлення («Вірно!» або «Невірно!»).

- Очищення поля після перевірки.

6. Запустити головний цикл обробки подій (mainloop).

## 1. Теоретичні відомості

## 2. Лістинг коду програми

```
#root
import tkinter as tk
root = tk.Tk()
root.title("Кодовий замок")
root.geometry("450x450+350+0")
root.configure(bg="#1b1c1c")

root.grid_rowconfigure(0, weight=1)
root.grid_columnconfigure(0, weight=1)
root.grid_columnconfigure(1, weight=1)

VALIDE_CODE="9971"
LIST_CODE=list(range(95, 106))

label_var = tk.StringVar()

#specialFunc
def changeColor():
    label.config(fg="#34ebe8")
    label_var.set("")

def on_key(event):
    if event.keycode in LIST_CODE:
        label_var.set(label_var.get() + event.char)
    if event.keycode == 13:
        onClickVerify()
    if event.keycode == 8:
        onClickDel()

#onClickfn
def onClickNum(num):
    label_var.set(label_var.get() + str(num))

def onClickDel():
    label_var.set(label_var.get()[:-1])

def onClickVerify():
    if (VALIDE_CODE==label_var.get()):
        label.config(fg="green")
        label_var.set("Approved")
    else:
        label.config(fg="red")
        label_var.set("Disaprovved")
        label.after(1000, changeColor)

#label
label = tk.Label(
    root,
    textvariable=label_var,
    bg="black",
    fg="#34ebe8",
    font=("Arial", 14, "normal"),
    width=20,
    height=10,
    anchor="center"
)
```

```

label.grid(row=0, column=0, padx=10, pady=10)

#frame for btns
frame = tk.Frame(
    root,
    bg="black",
)
frame.grid(column=1,row=0)

#buttons
for i in range(1, 10):
    button = tk.Button(
        frame,
        bg="#34ebe8",
        font=("Arial", 14, "normal"),
        text=str(i),
        width=4,
        height=2,
        command=lambda num=i: onClickNum(num),
    )
    button.grid(row=(i-1)//3, column=(i-1)%3,padx=1, pady=1)

button_0 = tk.Button(
    frame,
    text=0,
    bg="#34ebe8",
    fg="black",
    font=("Arial", 14, "normal"),
    width=4, height=2,
    command= lambda num=0 : onClickNum(num)
)
button_0.grid(row=3, column=1,padx=1, pady=1)

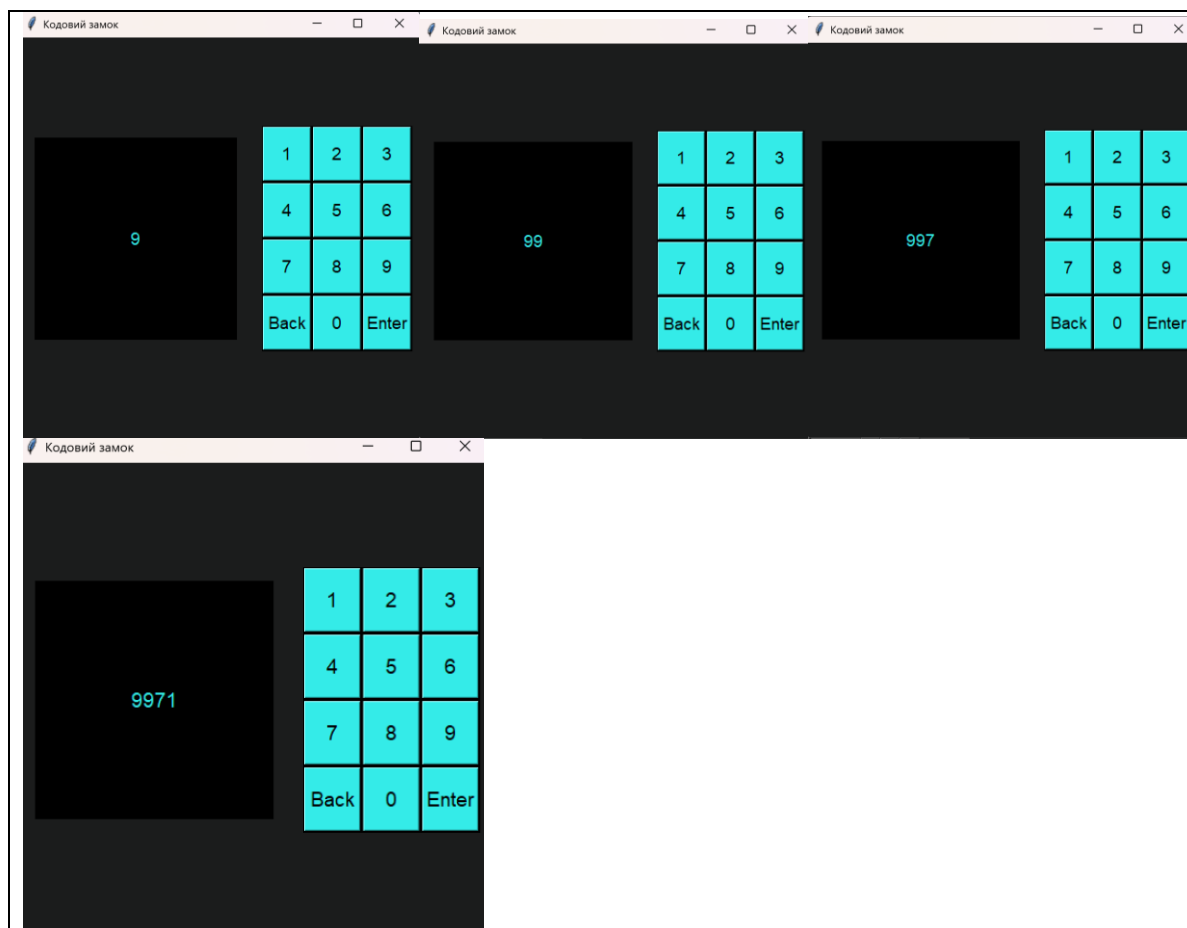
button_back = tk.Button(
    frame,
    text="Back",
    bg="#34ebe8",
    fg="black",
    font=("Arial", 14, "normal"),
    width=4, height=2,
    command=onClickDel
)
button_back.grid(row=3, column=0,padx=1, pady=1)

button_enter = tk.Button(
    frame,
    text="Enter",
    bg="#34ebe8",
    fg="black",
    font=("Arial", 14, "normal"),
    width=4, height=2,
    command=onClickVerify
)
button_enter.grid(row=3, column=2, padx=1, pady=1)
root.bind("<KeyPress>", on_key)
root.mainloop()

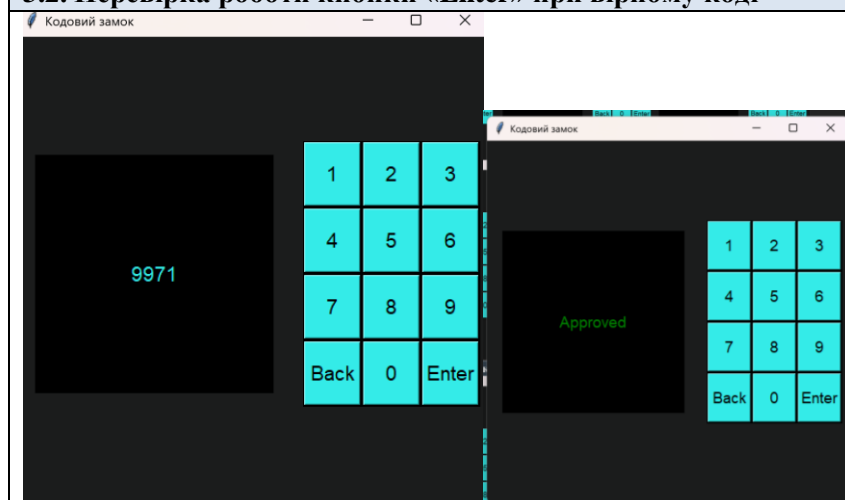
```

### 3. Перевірка та тестування програми

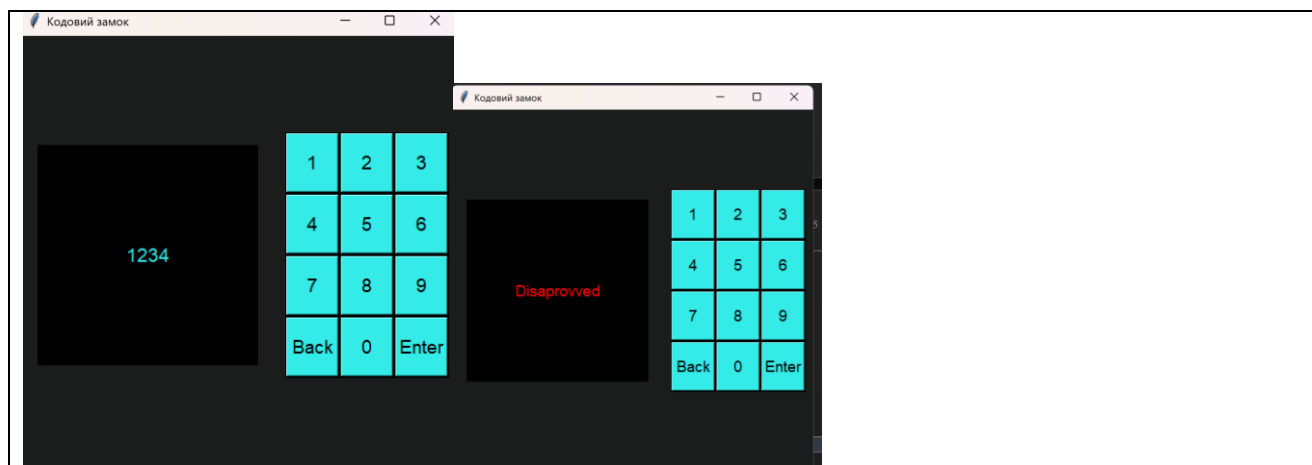
#### 3.1. Перевірка введення коду (на прикладі коду, який задано згідно із варіантом)



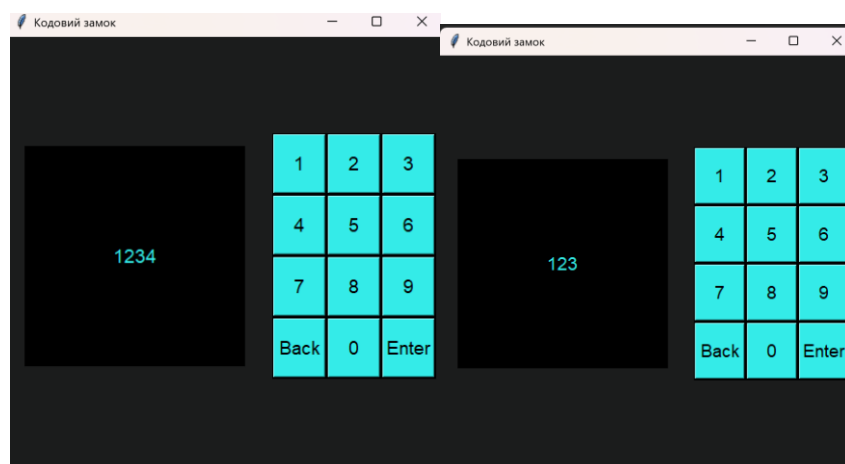
### 3.2. Перевірка роботи кнопки «Enter» при вірному коді



### 3.3. Перевірка роботи кнопки «Enter» при невірному коді



### 3.4. Перевірка роботи кнопки «Back»



### Висновок

Висновок має відповісти на запитання «Що зроблено?», «Як зроблено?», «Що це дало?».

На даній лабораторній роботі я створив кросплатформений додаток з використанням Python та як основи графічного інтерфейсу tkinter. Для написання використовував середовище pytho, всі необхідні дії виносив у функції, а також максимально структурував та оптимізував код. Виконавши індивідуальне практичне завдання, засвоїв використання бібліотеки tkinter у створенні десктопних додатків. За дизайн замка було взято прототип реального сейфу. Як додатковий функціонал реалізовано: можливість введення та підтвердження, а також видалення даних з клавіатури. Виконана робота продемонстрована вище.