
# Homework #9

**01286121 Computer Programming**

**Software Engineering Program,**

**Department of Computer Engineering,**

**School of Engineering, KMITL**

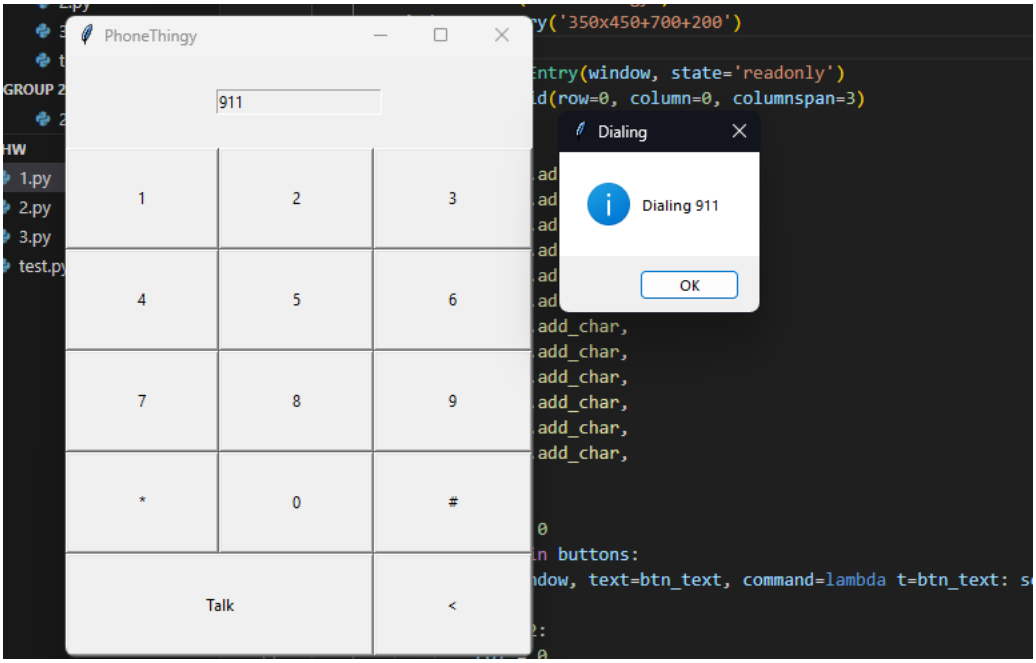
By

66011149 Phatthadon Sornplang

1. Create a mobile phone user interface as follows:

```
from tkinter import *
from tkinter import messagebox

class Phone:
    def __init__(self):
        window = Tk()
        window.title("PhoneThingy")
        window.geometry('350x450+700+200')

        self.inpus = Entry(window, state='readonly')
        self.inpus.grid(row=0, column=0, columnspan=3)

        buttons = {
            "1": self.add_char,
            "2": self.add_char,
            "3": self.add_char,
            "4": self.add_char,
            "5": self.add_char,
            "6": self.add_char,
            "7": self.add_char,
            "8": self.add_char,
            "9": self.add_char,
            "*": self.add_char,
            "0": self.add_char,
            "#": self.add_char,
        }

        row, col = 1, 0
        for btn_text in buttons:
            Button(window, text=btn_text, command=lambda t=btn_text: self.add_char(t)).grid(row=row, column=col,
sticky='nsew')
```

```
            col += 1
            if col > 2:
                col = 0
                row += 1
        Button(window, text="Talk", command=self.dial_number).grid(row=5, column=0, columnspan=2, sticky='nsew')
        Button(window, text="<", command=self.delete_rightmost_char).grid(row=5, column=2, columnspan=1,
sticky='nsew')

        window.columnconfigure((0, 1, 2), weight=1)
        window.rowconfigure((0, 1, 2, 3, 4, 5), weight=1)
        window.mainloop()

    def add_char(self, char):
        self.inpus.config(state=NORMAL)
        self.inpus.insert(END, char)
        self.inpus.config(state="readonly")

    def dial_number(self):
        number = self.inpus.get()
        messagebox.showinfo("Dialing", f"Dialing {number}")

    def delete_rightmost_char(self):
        text = self.inpus.get()
        if text:
            self.inpus.config(state=NORMAL)
            self.inpus.delete(len(text) - 1)
            self.inpus.config(state="readonly")

Phone()
```




2. Create a GUI for your Python project and submit it as homework No.9

x = input("Enter some text: ")

import tkinter as tk

from tkinter import messagebox

class InfoApp:

    def __init__(self, root):

        root.title('Information')

        self.lbl_name = tk.Label(root, text="Name:")

        self.lbl_name.grid(row=0, column=0, padx=10, pady=10, sticky="w")

        self.ent_name = tk.Entry(root)

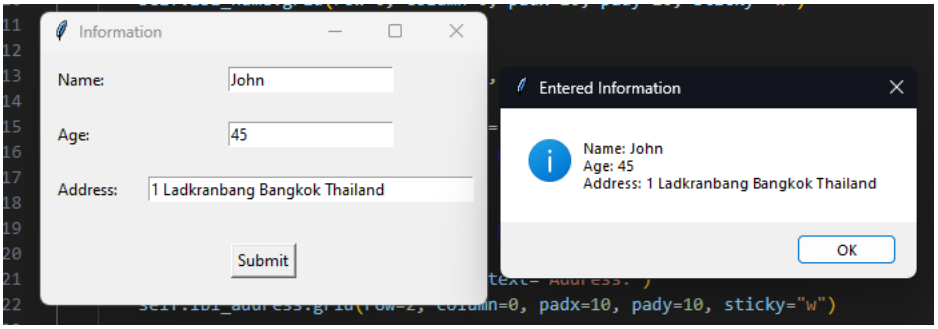
        self.ent_name.grid(row=0, column=1, padx=10, pady=10)

        self.lbl_age = tk.Label(root, text="Age:")

        self.lbl_age.grid(row=1, column=0, padx=10, pady=10, sticky="w")

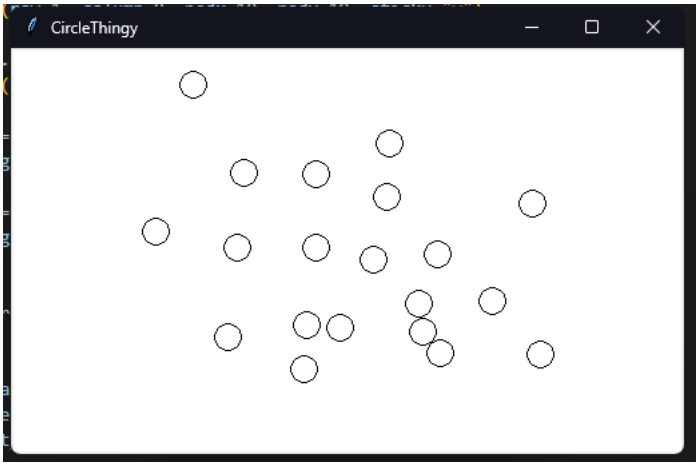
        self.ent_age = tk.Entry(root)

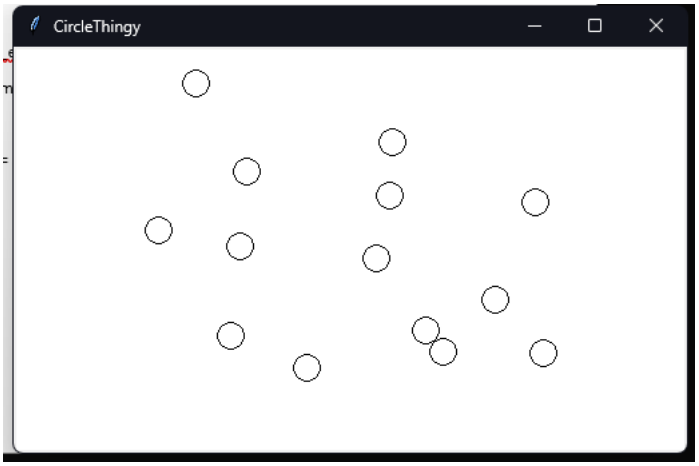
```
        self.ent_age.grid(row=1, column=1, padx=10, pady=10)

        self.lbl_address = tk.Label(root, text="Address:")
        self.lbl_address.grid(row=2, column=0, padx=10, pady=10, sticky="w")

        self.ent_address = tk.Entry(root, width=40)
        self.ent_address.grid(row=2, column=1, padx=10, pady=10)

        self.btn_submit = tk.Button(root, text="Submit", command=self.show_info)
        self.btn_submit.grid(row=3, column=0, columnspan=2, pady=20)

    def show_info(self):
        name = self.ent_name.get()
        age = self.ent_age.get()
        address = self.ent_address.get()


        messagebox.showinfo('Entered Information', f'Name: {name}\nAge: {age}\nAddress: {address}')

if __name__ == '__main__':
    root = tk.Tk()
    app = InfoApp(root)
    root.mainloop()
```



3. Write a program that displays a new circle at the position of the left mouse click and removes the circle at the position of the right mouse click.

```
from tkinter import *
from tkinter import messagebox


class CircleThingy:
    def __init__(self):
        self.window = Tk()
        self.window.title("CircleThingy")
        self.canvas = Canvas(self.window, width=500, height=300, bg='white')
        self.canvas.pack(padx=0, pady=0)


        self.prev = None
        self.canvas.bind('<Button-1>', self.draw)
        self.canvas.bind('<Button-3>', self.dele)


        self.window.mainloop()


    def draw(self, move_event):
        self.canvas.create_oval(move_event.x + 10, move_event.y + 10, move_event.x - 10, move_event.y - 10,
width=1, fill='white')
        self.prev = move_event


    def dele(self, move_event):
        item = self.canvas.find_closest(move_event.x, move_event.y)
        x1, y1, x2, y2 = self.canvas.coords(item)


        if x1 <= move_event.x <= x2 and y1 <= move_event.y <= y2:
            self.canvas.delete(item)


        self.prev = move_event



CircleThingy()
```


CircleThingy


CircleThingy