Лабораторна роботи №12

Основи створення віконного візуального інтерфейсу програм. Елементи керування їх властивості. Опрацювання подій.

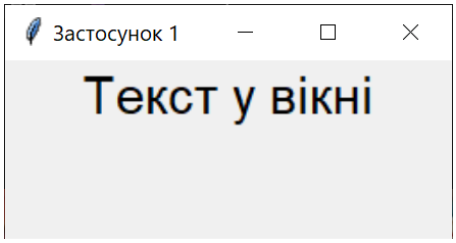
Мета роботи: *знайомство з базовими можливостями створення візуального інтерфейсу користувача засобами бібліотеки Tkinter.*

Короткі теоретичні відомості

Пересічний користувач комп'ютера сьогодні звик забезпечувати свої обчислювальні потреби за допомогою програм з графічним інтерфейсом, а не шляхом спілкування з програмою за допомогою текстових команд через консоль. Мова програмування ***Python*** дозволяє створювати програми з візуальним інтерфейсом. Для цього в ***Python*** використовується спеціальна бібліотека компонентів ***tkinter***, вона доступна як окремий вбудований модуль, що постачається разом з інтерпретатром.

Назва "***Tkinter***" є скороченням від "Tk interface", ця бібліотека створена у 1988 році для мови Tcl. ЇЇ зачинатель – Джон Остерхаут (John Ousterhout), професор computer science з Берклі. Згодом Tk було адаптовано для широкої низки динамічних мов, зокрема, для Ruby, Perl та для Python у 1994 році. Tkinter - кросплатформний, один і той самий код працюватиме однаково на різних платформах (Mac OS, Linux і Windows). Він поширюється за BSD-ліцензією, тому бібліотека може бути використана як для власних потреб, так і для комерційної розробки.

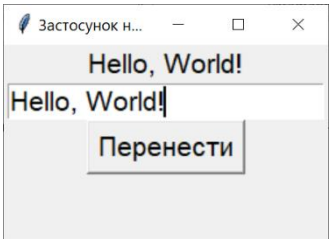
Для створення вікна використовуємо функцію ***`Tk()`*** з модуля ***`tkinter`*** (подібні функції модуль має і для створення інших візуальних компонентів). Щоб вивести вікно на екран в його об'єкта викликається метод ***`mainloop()`***. Наприклад:




```
import tkinter

root = tkinter.Tk()  # об'єкт вікна

root.title('Застосунок 1')

root.geometry('250x100')  # розміри  вікна

# об'єкт текстова мітка

label = tkinter.Label(text='Текст у вікні',

                      font=('Arial',22))

label.pack()  # додаємо мітку до вікна

root.mainloop() # відображення вікна на екрані
```

В нашому прприкладі створене вікно присвоюється змінній root, від її імені ми керуємо параметрами вікна. Окрім вікна в прикладі вище створено текстовий підпис ***Label***, його додаємо до вікна методом ***pack()***.

Для вводу тексту можна скористатися візуальним компонентом ***Entry***, але для зчитування введених символів, йому потрібно передати спеціальну текстову змінну (див. код далі). Реакцію віконного інтерфейсу на дії користувача програмують з використанням подій візуальних компонентів. Найпростіший вид взаємодії користувача з програмою – натискання кнопок, створених для виконання того чи іншого завлдання. В tkinter з цією метою застосовують об’єкт Button, а для опрацювання кліку по кнопці в її конструктор передають наперед створену для цього функцію, як це зроблено в коді далі:




```
# така форма імпорту дозволяє не писати назву модуля
from tkinter import * # перед функціями
root = Tk()  # об'єкт вікна
root.title('Застосунок на Tkinter')
# розміри та розташування вікна
root.geometry('250x150+250+150')
# об'єкт для зберігання введеного користувачем тексту
user_input = StringVar(root)
user_input.set(value='Введіть текст тут')

label = Label(text='Текст у вікні', font = ('Arial', 16))
label.pack()  # додаємо мітку до вікна
text = Entry(textvariable=user_input,
             font = ('Arial', 16)) # Поле для вводу
text.pack() # додаємо поле до вікна

# функція для обробки кліку по кнопці
def btn_click():
    label["text"] = user_input.get()

button = Button(text="Перенести",
        font = ('Arial', 16), command=btn_click) # кнопка
```

```
button.pack() # додаємо кнопку до вікна
root.mainloop() # відображення вікна на екрані
```

Дизайн вікна з прикладу, звісно ж, потребує покращення, проте базові програмні конструкції, потрібні для розв'язування задачі з цієї лабораторної роботи в нашому прикладі розглянуто.

До розробки графічного інтерфейсу на ***Python*** можна використовувати спеціальні візуальні редактори, найбільш популярні з них:

PAGE – Python Automatic GUI Generator. Це безкоштовний візуальний редактор з відкритим кодом для ***Tkinter***, який дозволяє створювати графічні інтерфейси шляхом перетягування віджетів на полотно.

Qt Designer – візуальний редактор для створення графічних інтерфейсів за допомогою бібліотеки PyQt, яка є зв'язкою Python та інструментарію Qt GUI. Тут можна створювати графічні інтерфейси за допомогою Qt Designer, а потім конвертувати їх у код на Python за допомогою інструменту pyuic.

Glade – візуальний редактор для створення графічного інтерфейсу користувача за допомогою інструментарію GTK+, який є популярним інструментарієм для створення десктопних додатків у Linux. Glade дозволяє створювати графічні інтерфейси шляхом перетягування віджетів на полотно, а потім експортувати їх на різні мови програмування, включаючи Python.

Використання візуального редактора може заощадити ваш час і зусилля при створенні графічних інтерфейсів, оскільки дозволяє спроектувати інтерфейс без необхідності писати весь код вручну. Однак важливо відзначити, що візуальні редактори не завжди можуть генерувати найбільш ефективний або гнучкий код, тому вам може знадобитися відредагувати згенерований код вручну, щоб досягти бажаної функціональності.

Щоб використовувати ***PAGE*** у ***PyCharm***, спочатку потрібно встановити ***PAGE,*** завантаживши останню версію PAGE з офіційного сайту: https://sourceforge.net/projects/page/ та дотримуючись інструкцій з інсталяції, наданих на сайті, щоб встановити її.

Далі потрібно:

- створити новий проект у PyCharm;
- створити новий файл Python;
- створити файл інтерфейсу користувача за допомогою PAGE;
- експортвати файл інтерфейсу у форматі "Python";
- імпортуват файл інтерфейсу в PyCharm;
- використстати цей файл інтерфейсу у Python-коді.

Завдання

Використовуючи модуль ***Tkinter*** створити та запрограмувати візуальний інтерфейс до програми розв'язування простої обчислювальної задачі для розв'язання задачі згідно варіанту. У звіт подати:

- умову задачі варіанта;
- малюнок або схему вікна програми;
- складений Python-код;
- результати обчислень для деякого набору вхідних даних у вигляді знімка екрану з вікном програми.

1. Дано три цілих числа. Знайти кількість додатних чисел в цьому наборі.

2. Дано два числа. Вивести спочатку більше, а потім менше з них.

3. Дано дві змінні дійсного типу: A, B. Перерозподілити значення даних змінних так, щоб A виявилось меншим із заданих значень, а B - більше. Вивести нові значення змінних A і B.

4. Дано дві змінні цілого типу: A і B. Якщо їхні значення не рівні, то присвоїти кожній змінної суму цих значень, а якщо рівні, то присвоїти змінним нульові значення. Вивести нові значення змінних.

5. Дано три числа. Знайти найменше з них.

6. Дано три числа. Знайти середнє з них (тобто число, розташоване між найменшим і найбільшим).

7. Дано три числа. Вивести спочатку найменше, а потім найбільше з даних чисел.

8. Дано три числа. Знайти суму двох більших з них.

9. На числовій осі розташовані три точки: A, B, C. Визначити, яка з точок B і C розташована ближче до A, і вивести цю точку і її відстань від точки A.

10. Дано координати точки, що не лежить на координатних осях OX та OY. Визначити номер координатної чверті, в якій знаходиться дана точка.

11. Дано цілочисельні координати трьох вершин прямокутника, сторони якого паралельні координатним осям. Знайти координати його четвертої вершини.

12. Дано номер року (додатне ціле число). Визначити кількість днів в цьому році, враховуючи, що звичайний рік нараховує 365 днів, а високосний.

13. Дано два числа. Вивести спочатку більше, а потім менше з них.

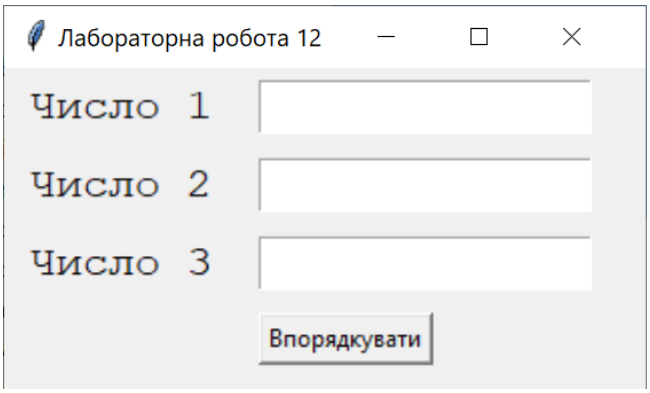
14. Дано три числа. Знайти середнє з них (тобто число, розташоване між найменшим і найбільшим).

15. Дано три числа. Вивести спочатку найменше, а потім найбільше з даних чисел.

16. Дано три цілих числа. Записати ці числа у порядку зростання.

Приклад виконання завдання

Варіант № 16. Умова задачі: Дано три цілих числа. Записати ці числа у порядку зростання.



Вікно для роботи програми, відповідно до поставленої задачі повинномати три тектсові поля ***Entry*** для вводу чисел. Запуск розв'язування задачі, очевидно, буде виконуватися за допомогою кнопки ***Button***, та функції, що опрацьовуватиме її натискання. Для позначення полів з числами використаємо підписи ***Label***. Орієнтовна конструкція вікна програми:

Створимо віконний інтерфейс з програмного коду, використовуючи функцію ***Tk()*** та конструктори відповідних елементів керування. Для додавання компоненнтів до вікна скористаємося методом ***grid(row, column)***, який дає можливість компонувати елементи вікна у вигляді уявної прямокутної таблиці. В методі, викликаному від імені компонента вказуємо рядок (***row***) та стовпець (***column***) цієї таблиці, в яких буде виводитися цей компонент.

З огляду на те, що використання візуальних редакторів для віконного інтерфейсу потребуватиме додаткової конвертації на ***Python***, та, часто-густо, ще й редагування імпортованого коду, додаткових візуальних знарядь ми не використовуватимемо, а писатимемо код програми «з нуля».

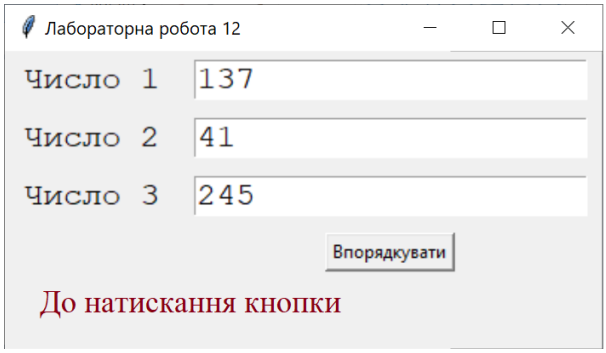
Орієнтовний код програми:

```
from tkinter import *
window = Tk()  # об'єкт вікна
window.title('Лабораторна робота 12')  # заголовок вікна
window.geometry('440x200')  # розміри  вікна
# об'єкти для зчитування даних з текстових полів
num1_str = StringVar(window)
num2_str = StringVar(window)
num3_str = StringVar(window)
```

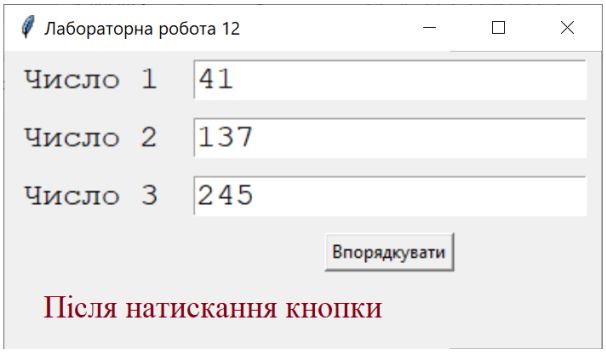
```
label1 = Label(text = 'Число 1', font = ('Courier',16))
# підпис до поля 1
label1.grid(row = 0, column = 0, padx = 10, pady = 5 )
# додаємо мітку до вікна
text1 = Entry(textvariable = num1_str, font =
('Courier',16))
text1.grid(row = 0, column = 1, padx = 10, pady = 5)
label2 = Label(text = 'Число 2', font = ('Courier',16))
# підпис до поля 2
label2.grid(row = 1, column=0, padx = 10, pady = 5)
# додаємо мітку до вікна
text2 = Entry(textvariable = num2_str,
                            font = ('Courier',16))
text2.grid(row = 1, column = 1, padx = 10, pady = 5)
label3 = Label(text = 'Число 3', font = ('Courier',16))
# підпис до поля 3
label3.grid(row = 2, column=0, padx = 10, pady = 5)
# додаємо мітку до вікна
text3 = Entry(textvariable = num3_str,
                               font = ('Courier',16))
text3.grid(row = 2, column = 1, padx = 10, pady = 5)
# функція-обробник натискання кнопки
def button_click():
    a = int(num1_str.get())
    b = int(num2_str.get())
    c = int(num3_str.get())
    if c < b :
        tmp = c; c = b; b = tmp
    if b < a :
        tmp = a; a = b; b = tmp
```

```
    if c < b :
        tmp = c; c = b; b = tmp
    # записуємо впорядковані значення
    # в текстові поля назад
    num1_str.set(a.__str__())
    num2_str.set(b.__str__())
    num3_str.set(c.__str__())
button = Button(text = 'Впорядкувати',
                                command = button_click)
button.grid(row = 3, column = 1, padx = 10, pady = 5)
window.mainloop() # відображення вікна на екрані
```

Результат роботи програми



До натискання кнопки



Після натискання кнопки

Запитання для самоконтролю

1. Який функціонал містить модуть ***Tkinter***?
2. Яка дія функції ***Tk()***?
3. Які засоби має бібліотека ***Tkinter*** для створення візуальних компонентів?
4. Які функції потрібно використати, щоб створити текстове поле?
5. Які функції потрібно використати, щоб створити кнопку?
6. Який механізм ***Tkinter*** використовується для опрацювання подій компонентів візуального інтерфейсу?
7. Чи існують засоби для візуальної розробки віконного інтерфейсу з використанням ***Tkinter***?