

Практичне завдання

§ 39. Візуалізація даних

Вправа 39.2 Побудова графіка

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно гігієнічних норм



Завдання. Побудувати графік випадіння опадів за кожний місяць по Україні відповідно до зразка (рис. 39.4). При натисненні кнопки **Створити список** має бути сформований список опадів на кожний місяць, де значенням є випадкове число від 40 до 120.

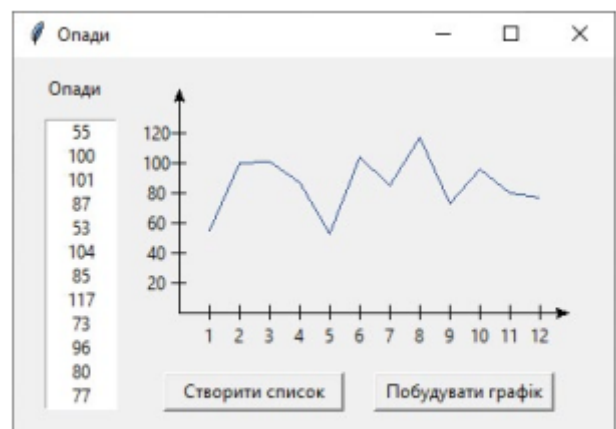


Рис.39.4

При натисненні кнопки **Побудувати графік** має здійснюватися побудова графіка за даними зі списку.

Створення списку опадів

1. Завантажте файл **Вправа_39_2.py** із матеріалів до даного параграфу.
У наданому файлі здійснено імпорт необхідних модулів та створено елементи інтерфейсу програми: список **Lbox** та дві кнопки **Btn1** та **Btn2**.
2. Створіть обробник події натиснення кнопки **Створити список**.

```
def Btn1_click():  
    p=[]  
    #очищення списку від попередніх значень  
    Lbox.delete(0,END)  
    for i in range(12):  
        #створення елемента, що є випадковим числом  
        p.append(randint(40,120))  
        #додавання елемента масиву до списку Lbox  
        Lbox.insert(END,p[i])
```

3. В конструкторі кнопки **Btn1** додайте команду виклику обробника події **Btn1_click**.
4. Запустіть програму та перевірте створення списку.

Створення осей

5. Додайте команди створення та розташування у вікні графічного полотна:

```
canvas=Canvas(tk,width=300,height=250)
canvas.place(x=80,y=10)
```

6. Додайте команди створення координатних осей:

```
canvas.create_line(30,160,290,160,arrow=LAST) #x
canvas.create_line(30,160,30,10,arrow=LAST) #y
```

7. Додайте команди створення штрихів та підписів на осі **x**.

```
#штрихи та підписи на осі x
x=[]
for i in range(12):
    x.append(20*i+50)
    #створення штрихів
    canvas.create_line(x[i],155,x[i],165)
    #створення підписів
    canvas.create_text(x[i],175,text=i+1)
```

У даному коді створено масив **x**, що буде містити значення координат штрихів по осі **x**. Значення по **x** будуть змінюватися через кожні 20 рх, отже, перша координата відносно полотна буде знаходитися на відстані 30 (тут розташовано початок координат) +20=50 рх. Отже, загальна формула для значень масиву **x** має бути **20*i+50**, де **i** змінюється від 0 до 11 включно.

8. Додайте команди створення штрихів та підписів на осі **y**.

Оскільки максимальне значення, можливе по осі **y** — це 120, а ціна поділки дорівнює 20, то для створення штрихів та підписів достатньо 6-ти ітерацій.

Масив **y** буде мати такі значення [20,40,...,120].

Отже, для його створення введіть такий код:

```
y=[]
for i in range(6):
    y.append(20*(i+1))
```

9. Нижче у тілі циклу додайте команди створення штрихів та підписів по осі **y**.

```
canvas.create_line(25,160-y[i],35,160-y[i])
canvas.create_text(15,160-y[i],text=y[i])
```

10. Запустіть програму на виконання та перевірте правильність побудови осей.

Побудова графіка

11. Створіть обробник події кнопки **Побудувати графік**.

Оскільки графік складається з відрізків, то для його побудови доцільно використовувати метод **create_line**. Також неважко помітити, що таких відрізків, що сполучають 12 точок, буде 11. Для даного методу **x1=x[i]**, **y1=160-p[i]**, **x2=x[i+1]**, **y2=160-p[i+1]**. Отже, **x[i]** та **p[i]** — це поточні елементи масивів, а **x[i+1]** та **p[i+1]** — це наступні елементи масивів.

Враховуючи попередні пояснення, уведіть наведений нижче код:

```
def Btn2_click():
    for i in range(11):
        canvas.create_line(x[i],160-p[i],x[i+1],
                           160-p[i+1],fill="blue")
```

12. В конструкторі кнопки Btn2 додайте команду виклику обробника події **Btn2_click**.

13. Оскільки створення масиву **p** відбувалося в іншому обробнику подій, змінна **p** має бути оголошена глобальною. Додайте оголошення змінної **p** в якості глобальної в тілі обробника події **Btn1_click**:

```
global p
```

14. Запустіть програму на виконання та перевірте правильність побудови графіка.

Сформууйте список ще раз, натиснувши кнопку **Створити список** та побудуйте графік, натиснувши кнопку **Побудувати графік**.

В результаті новий графік буде накладатися на попередній. Аби цього не відбувалося, потрібно очистити полотно від попереднього зображення.

Щоб очистити полотно від зображення, використовують метод **delete**, що має таку конструкцію: **полотно.delete(ALL)**.

15. Додайте команду **canvas.delete(ALL)** до обробника події кнопки **Btn2_click** перед оператором циклу та перевірте, як буде працювати програма. В

результаті натиснення кнопки **Побудувати графік** буде видалено все, що було побудовано до графіка.

Для видалення лише одного об'єкта, а в нашому випадку перед побудовою нового графіка ми маємо видаляти попередній, в команді створення відрізка графіка необхідно додати властивість **tags=ім'я**, а в якості параметра методу **delete** вказати ім'я даного тегу, який ми бажаємо видаляти.

16. В команді створення відрізка графіка додайте властивість **tags**.

```
canvas.create_line(x[i],160-p[i],x[i+1],160-p[i+1],  
fill="blue",tags="g")
```

17. Змініть параметр методу **delete** наступним чином:

```
canvas.delete("g")
```

18. Запустіть програму та перевірте правильність її виконання.