

Уроки 31 і 32

Проект "Піцерія". Етап 1

Потрібно знати команди виведення елементів керування, опрацювання подій та користувацькі функції.

Почати

Згадайтеся, що буде виводити наведена програма, і перевірте свою здогадку в Python.

Яку команду потрібно додати, щоб програма відображала те значення, яке введе користувач?

```
def f():  
      
    x=int(input("Введіть x:"))  
  
f()  
print(x)
```

Введіть команду на місці пропуску.

Сьогодні ми створимо електронний бланк замовлення для піцерії!

Власник ресторану може вводити ціни, а клієнт — вказувати кількість кожної страви. Програма має автоматично визначити вартість кожного найменування в замовленні, після натискання кнопки **Розрахувати** — вивести сумарну вартість.

Найменування	Ціна, грн.	Кількість	Вартість, грн.
Піца	75	0	0
Морозиво	12	0	0
Тістечко	16	0	0
Сік	8	0	0
Вартість замовлення:		0 грн.	Розрахувати



Демонстрація роботи проекту.

Найменування	Ціна, грн.	Кількість	Вартість, грн.
Піца	75	2	150
Морозиво	12	1	12
Тістечко	16	0	
Сік	8	0	

Вартість замовлення: 0 грн. Розрахувати

Вкажіть, які елементи керування необхідно вибрати для оформлення інтерфейсу програми.

Найменування	Ціна, грн.	Кількість	Вартість, грн.
Піца	75	0	0
Морозиво	12	0	0
Тістечко	16	0	0
Сік	8	0	0

Вартість замовлення: 0 грн. Розрахувати

Оберіть відповіді. Перекладіть з англійської слово **scale**. Також зауважте, що ціну може змінювати користувач, а вартість обчислюється тільки автоматично.

Розглянемо створення інтерфейсу програми покроково. Почнемо з першого рядка.

Заповніть пропуски в коді відображення написів.



```
name=  (text= "Найменування", font=("Arial", 12, "bold"))
name.place(x=  ,y=  )
price=  (text= "Ціна, грн.", font=( ,  ,  ))
price.place(x=  ,y=  )
number=  (text= "Кількість", font=("Arial", 12, "bold"))
number.place(x=  ,y=  )
cost=  (text= "Вартість, грн.", font=("Arial", 12, "bold"))
cost.place(x=  ,y=  )
```

Заповніть пропуски.



Вправа 1.

- 1) Створіть графічне вікно розміром **440x300**.
- 2) Розташуйте в ньому написи першого рядка та першого стовпця, враховуючи, що відступи між рядками – **40** пікселів.



Для введення кількості страв необхідно використати елемент керування «шкала» з конструктором **Scale**.

Шкала – це елемент керування, призначений для вибору чисел із діапазону.

Елемент керування «шкала» може мати такі властивості (усі вони необов'язкові):

orient – орієнтація (HORIZONTAL – горизонтальна, VERTICAL – вертикальна)

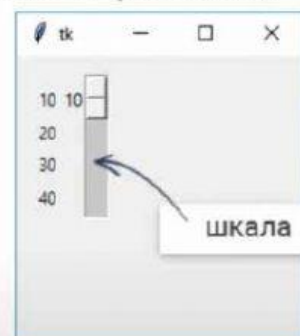
length – довжина шкали (у пікселях)

from_ – початкове значення шкали

to – кінцеве значення шкали

tickinterval – інтервал відображення міток шкали

resolution – крок переміщення повзунка шкали



```
sc = Scale(orient=VERTICAL, length=80, from_=10, to=40, tickinterval=10)
```

Заповніть пропуски в коді створення горизонтальної шкали довжиною **50** пікселів, що надає можливість вибирати значення від **0** до **10**.

sc1= [] (orient= [], length= [], from_= [], to= [])

Заповніть пропуски.

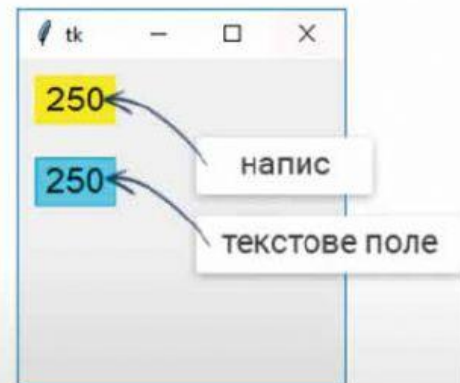
Для надання кольорового фону текстовим полям та написам, які містить вікно програми, необхідно скористатися властивістю **bg**, значенням якої є код кольору.

Наприклад, код для виведення кольорового напису:

```
lbl = Label(text="250", bg="yellow", font="12")  
lbl.place(x=10, y=10, width=50, height=30)
```

Код для виведення кольорового текстового поля:

```
ent = Entry(bg="cyan", justify="center", font="12")  
ent.place(x=10, y=60, width=50, height=30)  
ent.insert(END, "250")
```



Вправа 2.

1) Додайте у створеному вікні:

- текстові поля ціни з іменами **P1, P2, P3, P4**;
- шкали **S1, S2, S3, S4**;
- написи вартостей — **C1, C2, C3, C4** та загальної вартості — **C5**;
- інші написи нижнього рядка і кнопку.

2) Запустіть проект на виконання та перевірте роботу шкал.



Коди фонових кольорів можете вибрати на власний смак. Інформація тут:

http://www.science.smith.edu/dftwiki/index.php/Color_Charts_for_TKinter

1. Для виведення першого текстового поля з ціною піци скористайтесь кодом:

```
p1=Entry(font="Arial 12", bg="sky blue", justify="center")
p1.insert(END, "75")
p1.place(x=150, y=60, width=60, height=30)
```

2. Для виведення першої шкали скористайтесь кодом:

```
s1 = Scale(orient=HORIZONTAL, length=50, from_=0, to=10)
s1.place(x=230, y=50)
```

3. Для виведення першого напису з вартістю скористайтесь кодом:

```
c1=Label(text=0, font="Arial 12", bg="deep sky blue")
c1.place(x=310, y=60, width=60, height=30)
```

Обчислимо вартість першої страви – піци. Ціну піци зберігатимемо у змінній **x1**, кількість піц – у змінній **k1**, а вартість – у змінній **y1**.

Укажіть типи цих змінних.

x1 –

k1 –

y1 –

Найменування	Ціна, грн.	Кількість	Вартість, грн.
Піца	75.5	2	151

Як ви думаєте, у результаті виконання якої з перелічених подій має змінюватися вартість піци?

- ☐ Клацання ціни в текстовому полі
- ☐ Клацання напису з вартістю
- ☐ Пересування повзунка шкали
- ☐ Натискання кнопки "Розрахувати"

Найменування	Ціна, грн.	Кількість	Вартість, грн.
Піца	75	0	0
Морозиво	12	0	0
Тістечко	16	0	0
Сік	8	0	0

Вартість замовлення: 0 грн.

Щоб надати змінній **k1** значення положення повзунка шкали **s1**, використовуємо таку конструкцію:

змінна, що одержує значення положення повзунка (текстовий рядок)

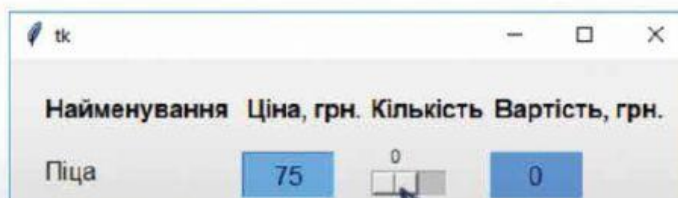
```
def s1_click(val):
```

```
    k1=int(val)
```

```
    s1 = Scale(orient=HORIZONTAL, length=50, from_=0, to=10, command=s1_click)
```

надання змінній **k1** цілочисельного значення положення повзунка

прив'язка функції **s1_click** до події зміни положення повзунка



Зауважте, що змінна **val** матиме тип **str**, тому для перетворення її на число пишемо **int(val)**. Для інших шкал код буде аналогічним.

шкала **s1**

Які ще три дії, крім обчислення кількості **k1**, потрібно виконати у функції **s1_click**, що опрацьовує зміщення повзунка шкали?

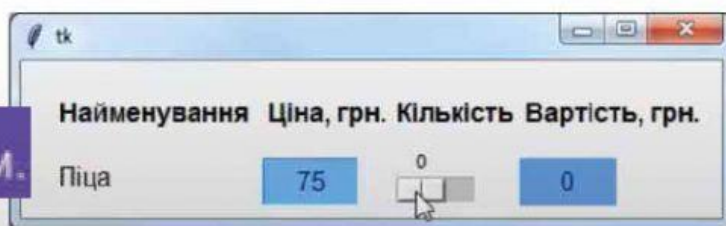
```
def s1_click(val):  
    k1=int(val)  
    ???
```

Впорядкуйте дії у правильній послідовності, а зайву дію поставте в кінці.

1. зчитування ціни з текстового поля
2. обчислення вартості **y1**
3. виведення вартості в написі
4. запис ціни в текстове поле

Перетягніть кроки алгоритму вгору/вниз.

Зайвий крок поставте останнім.

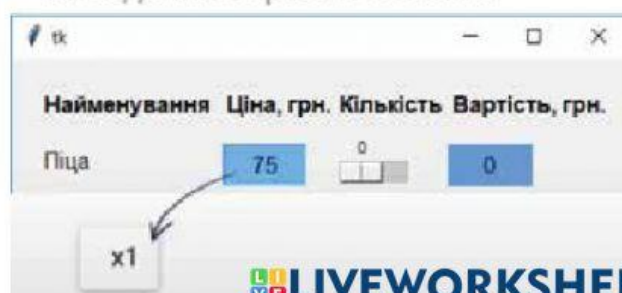


Заповніть пропуски в коді присвоювання змінній **x1**, що має дійсний тип, значення текстового поля **p1**.

```
[ ] = [ ] ([ ] - [ ] 0)
```

- зчитування ціни з текстового поля
- обчислення вартості **y1**
- виведення вартості в написі

Заповніть пропуски.



Оберіть формулу для визначення вартості піци **y1**, якщо **x1** – ціна піци, **k1** – кількість піц.

- ☐ $y1 = x1/k1$
- ☐ $y1 = x1 + k1$
- ☐ $y1 = x1 * k1$
- ☐ $y1 = k1/x1$

- зчитування ціни з текстового поля
- обчислення вартості **y1**
- виведення вартості в написі



Оберіть одну правильну відповідь.

Залишилося вивести вартість **y1** у написі **c1**. Для цього використаємо таку конструкцію:

запис значення **y1**
до рядка **var1**

```
def s1_click(val):
```

```
...  
var1.set(y1)
```

встановлення
зв'язку напису
зі змінною **var1**

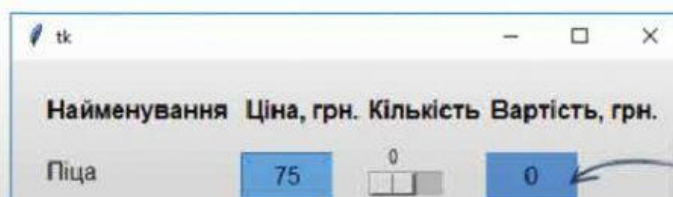
```
var1=StringVar()
```

```
c1 = Label(text=0, font="Arial 12", bg="#61a1ff", textvariable=var1)
```

створення змінної
текстового типу

set

— метод запису
значення до рядка
типу **StringVar**



напис **c1**

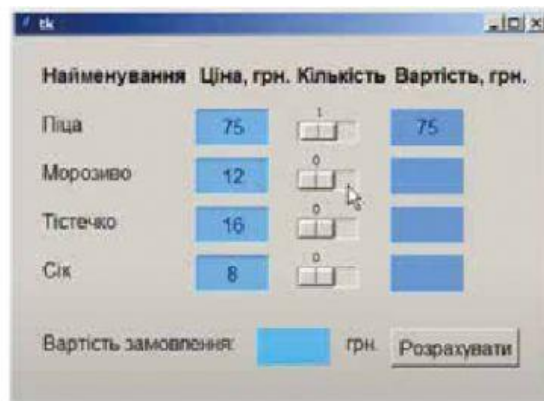
Зауважте, що метод виведення значення змінної в написі розташовано в тілі функції, що опрацьовує подію переміщення повзунка шкали.



Вправа 3. Додайте функції, що опрацьовують події пересування повзунків шкал **s1**, **s2**, **s3**, **s4**.

У результаті вартості страв мають підраховуватися та відображатися в написах **c1**, **c2**, **c3**, **c4**.

Виконайте завдання вправи у середовищі Python.



1. Створіть функцію опрацювання події пересування повзунка шкали **s1**.

```
def s1_click(val):
    k1=int(val)
    x1=float(p1.get())
    y1=x1*k1
    var1.set(y1)
```

2. Додайте прив'язку даної функції до події пересування повзунка шкали **s1**.

```
s1 = Scale(orient=HORIZONTAL,length=50,from_=0,to=10,command=s1_click)
```

3. Додайте код, що відображатиме в написі **c1** вартість усіх замовлених піц.

```
var1=StringVar()
c1=Label(text=0,font="Arial 12",bg="#61a1ff",textvariable=var1)
```

4. Аналогічні дії виконайте для інших елементів керування.

Виконайте дії у вказаній послідовності. Щоб запрограмувати інші шкали, створіть копії функції **s1_click**, змініть назви змінних та додайте прив'язки.

Заповніть пропуски у фрагменті коду визначення загальної вартості замовлення у разі натискання кнопки **Розрахувати**.

```
...
def btn_click():
    global y1,y2,y3,y4
    y= 
    var5. 
...
var5= 
lbl5=Label(text=0, font="Arial 12", bg="cyan", textvariable=  )
lbl5.place(x=200, y=240, width=60, height=30)
...
btn=Button(text="Розрахувати", font="Arial 12", command=  )
btn.place(x=310,y=240)
```

Заповніть пропуски.

Як ви думаєте, чому в обробнику натискання кнопки **Розрахувати** змінні **y1, y2, y3, y4** було оголошено як **global**?

- ☐ тому що значення цих змінних визначаються в інших функціях
- ☐ тому що обробник натискання кнопки **Розрахувати** — глобальна функція

```
def btn_click():  
    global y1, y2, y3, y4  
    y=y1+y2+y3+y4
```

Де ще ці змінні потрібно оголосити з декларацією **global**?

- ☐ в обробниках подій переміщення повзунків шкал
- ☐ на початку програми
- ☐ перед створенням кожного напису вартості

Виберіть по одній відповіді на кожне запитання.



Вправа 4.

Запрограмуйте натискання кнопки **Розрахувати**.

Перевірте роботу програми за умови вибору кожної страви. Запустіть програму ще раз і залиште необраною принаймні одну страву.

Сформууйте код обробника події натискання кнопки **Розрахувати**.
Перевірте роботу програми.

1. Оголосіть текстову змінну **var5**: **var5=StringVar()**.
2. Прив'яжіть змінну **var5** до напису загальної вартості замовлення за допомогою параметра **textvariable**.
3. Оголосіть змінні **y1, y2, y3, y4** з декларацією **global** у функції **btn_click**.
4. Кожну зі змінних **y1, y2, y3, y4** оголосіть з декларацією **global** в обробнику переміщення повзунка відповідної шкали.
5. У функції **btn_click** присвойте змінній **y** суму значень змінних **y1, y2, y3, y4**.
6. Виведіть значення **y** в написі: **var5.set(y)**

Запрограмуйте подію натискання кнопки **Розрахувати** покроково.