

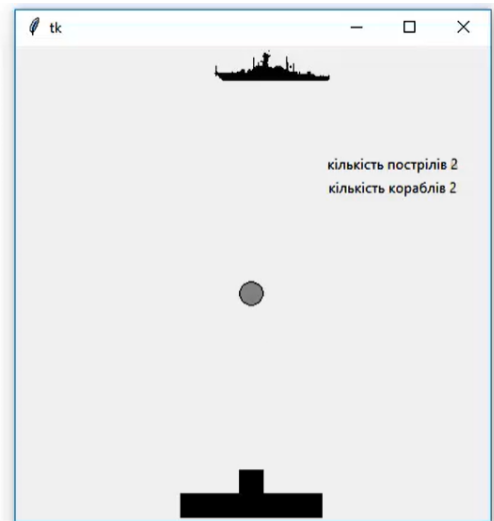
## Урок 37. Гра “Морський бій”. Етап 3

### Вивчення нового матеріалу

#### Слайд № 1

Продовжимо створювати гру “Морський бій”. Сьогодні ми вдосконалимо код таким чином, щоб переміщувався не один корабель, а 10, один за одним. Причому значення швидкості кожного корабля буде випадковим числом. Також додамо написи, що будуть виводити кількість пострілів.

Оскільки згідно умови гри ми маємо випустити усього **10 ядер**, обмежимо їх кількість. Також будемо виводити поточне значення кількості пострілів. Коли ядра закінчатся, введемо повідомлення “Ядра закінчилися!”



Напис, що відображатиме кількість кораблів, додамо на наступному уроці, а решту із зображеного на відео — на цьому уроці.

#### Слайд № 2

Для виведення напису в графічному вікні в Python використовують таку команду:

```
canvas.create_text(x, y, text='Повідомлення')
```

**x, y** — координати верхнього лівого кута напису

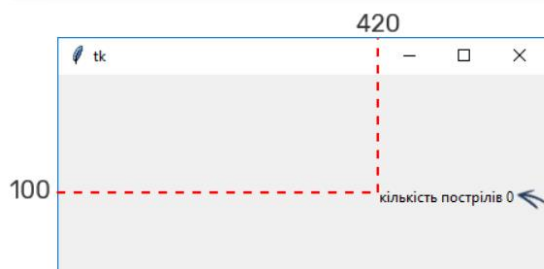


Запам'ятайте команду виведення текстового повідомлення.

### Слайд № 3

Для підрахунку кількості пострілів будемо використовувати змінну **shots**.  
Отже, для виведення напису про кількість зроблених пострілів скористаємося такими командами:

```
shots=0 #початкове значення  
canvas.create_text(420, 100, text='кількість пострілів' + str(shots)) #виведення повідомлення
```



функція, що перетворює  
значення числової змінної на  
текст

значення змінної **shots**



Функція **str** — це те саме, що й функція **int**, тільки навпаки.

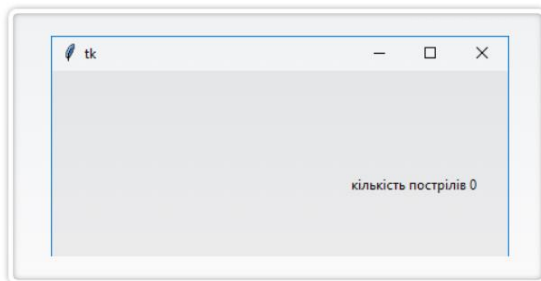
### Вправа

#### Вправа № 1



**Вправа 1.** Відкрийте файл програми, створений на попередніх уроках, та додайте в ньому команди для виведення повідомлення про кількість пострілів.

```
shots=0 #початкове значення  
canvas.cleate_text(420, 100, text='кількість пострілів' + str(shots)) #виведення повідомлення
```



Згадайтеся самостійно, де потрібно розташувати ці команди.  
Наразі кількість пострілів завжди буде **0**.

## Слайд № 4

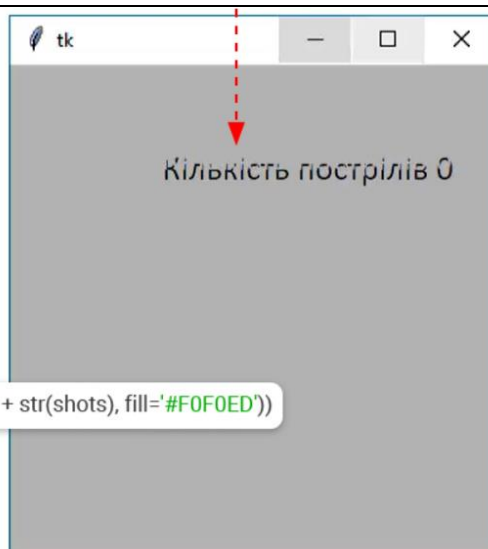
Під час виведення кожної наступної кількості (0,1,2,...) цифра накладатиметься на попередній напис.

Щоб «затерти» попередній напис, накладемо на нього ще один такий самий, але кольором фону.

Це можна зробити за допомогою команди

```
canvas.create_text(420, 100, text='кількість пострілів' + str(shots), fill='#F0F0ED')
```

Тут **#F0F0ED** — код кольору вікна.



Також, щоб затерти попередній напис, можна намалювати прямокутник із заливкою кольору **#F0F0ED**.

## Вправи

## Вправа № 2



**Вправа 2.** Змініть програмний код, додавши команди виведення повідомлення про кількість пострілів та повідомлення про закінчення ядер.

```
shots=0
canvas.create_text(420,100,text='кількість пострілів '+str(shots))
b=0
def ball(event):
    global b, shots
    if shots<10:
        canvas.delete(b)
        b=canvas.create_oval(240,460,260,480,fill='gray')
        canvas.create_text(420,100,text='кількість пострілів '+str(shots),fill='#F0F0ED')
        shots=shots+1
        canvas.create_text(420,100,text='кількість пострілів '+str(shots))
    else:
        canvas.create_text(50,100,text='Ядра закінчилися!')
canvas.bind_all("<space>",ball)
```



Не забудьте оголосити змінну **shots** у функції **ball** як **global**. Запустіть програму та перевірте правильність її виконання.

### Вправа № 3



### Вправа 3. Додайте зовнішній цикл для переміщення 10 кораблів.

Запустіть програму та перевірте правильність її виконання.

```
ship_image=PhotoImage(file='ship.gif')
for z in range(10):
    s=canvas.create_image(500, 0,anchor=NW, image=ship_image)
    for y in range(0,200):
        canvas.move(s,-3,0)
        canvas.move(b,0,-5)
        tk.update()
        time.sleep(0.02)
```



Тепер можна спробувати справді пограти в гру!

### Вивчення нового матеріалу

### Слайд № 5

Щоб працювати з випадковими числами, спочатку необхідно підключити модуль **random**.

```
import random
```

Для отримання (**генерування**) цілого випадкового числа **n** у діапазоні  **$a \leq n \leq b$**  використовують таку команду:

```
random.randint(a,b)
```



Слово **random** у перекладі з англійської означає «випадковий».

## Вправа № 4



**Вправа 4.** Додайте до програми команди, які задають випадкову швидкість руху корабля.

Не забудьте замінити крок у команді **canvas.move**.

```
import time
from tkinter import*
import random
...
```

```
ship_image=PhotoImage(file='ship.gif')
for z in range(10):
    s=canvas.create_image(500, 0,anchor=NW, image=ship_image)
    v=random.randint(2,5)
    for y in range(0,200):
        canvas.move(s,-v,0)
        canvas.move(b,0,-5)
    tk.update()
    time.sleep(0.02)
```



Запустіть програму та переконайся, що кораблі рухаються з різною швидкістю.

## Вправа № 5



**Вправа 5.** Змініть код так, щоб усі кораблі переміщалися за межі вікна проекту.

Запустіть програму і переконайтеся у правильності її виконання.

```
ship_image=PhotoImage(file='ship.gif')
for z in range(10):
    s=canvas.create_image(500, 0,anchor=NW, image=ship_image)
    v=random.randint(2,5)
    for y in range(300):
        canvas.move(s,-v,0)
        canvas.move(b,0,-5)
    tk.update()
    time.sleep(0.02)
```



Змінити треба тільки одне число!