ЖИТОМИРСЬКА ПОЛІТЕХНІКА
ДЕРЖАВНИЙ УНІВЕРСИТЕТ

@ZTUEDUUA

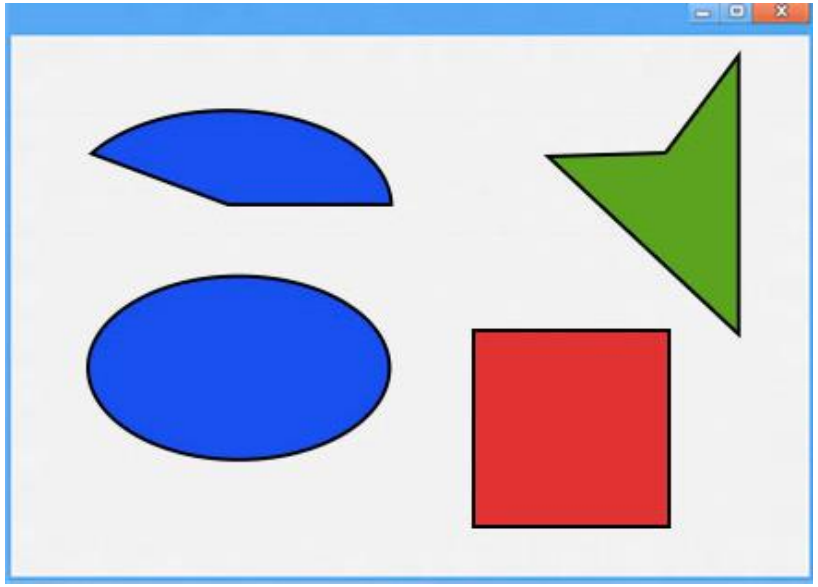
АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРИЗОВАНИХ ІНФОРМАЦІЙНО-ВИМІРЮВАЛЬНИХ СИСТЕМ

Лекція 14
Тема: Графіка в Python: Tkinter і Canvas

1.Позиціювання елементів в Python. Python grid.

2.Canvas: Малюємо лінії в Python.

3.Canvas: малюємо кольорові прямокутники в Python.



1.Позиціювання елементів в Python. Python grid

Для позиціювання елементів, під час роботи з Tkinter використовують різні методи:

- pack();
- place();
- grid.

Метод grid досволяє розмістити елемент в конкретну комірку умовної сітки або гріда.

Використовується ряд прикладів:

- column – це номер стовпчика, відраховується з нуля;
- row – це номер рядка, відраховується з нуля;
- columnspan – вказує число стовпчиків, використаних елементом;
- ipadx і ipady – відступи по горизонталі і вертикалі від границь до тексту компонента;
- Sticky – визначає вирівнювання елемента в комірці в випадку коли комірка більша компонента.

Для початку роботи з Tkinter бібліотеку спочатку потрібно імпортувати:

```
from tkinter import *
```

Створимо грід із дев’яти кнопок:

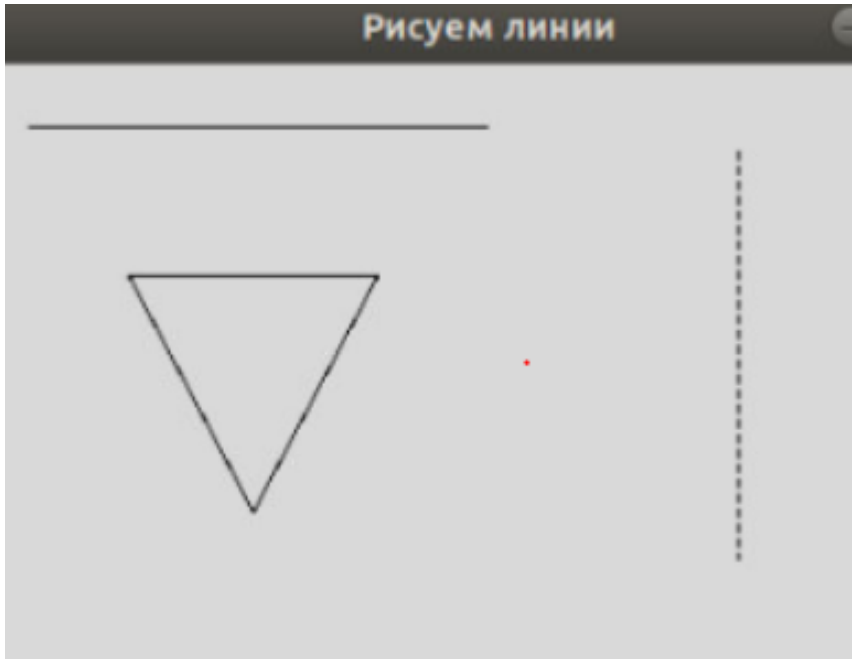
```
from tkinter import *

root = Tk()
root.title('GUI на Python')
root.geometry('300x250')

for r in range(3):
    for c in range(3):
        btn = Button(text="{}-{}".format(r, c))
        btn.grid(row = r, column=c, ipadx=10, ipady=6, padx=10, pady=10)

root.mainloop()
```

Результат:

2.Canvas: Малюємо лінії в Python

Віджет Canvas представляє функціональність, за допомогою якої розробник може намалювати (drawn) графіку в Tkinter.

Починати освоєння Canvas краще всього з ліній – примітивних геометричних елементів. В Canvas лінію з потрібним розміром можна створити через метод create_line(). Метод mainloop використовується для виклику вікна віджета.



Приклад коду:

lines.py

```python
from tkinter import Tk, Canvas, Frame, BOTH


class Example(Frame):

    def __init__(self):
        super().__init__()
        self.initUI()

    def initUI(self):
        self.master.title("Рисуем линии")
        self.pack(fill=BOTH, expand=1)

        canvas = Canvas(self)
        canvas.create_line(15, 25, 200, 25)
        canvas.create_line(300, 35, 300, 200, dash=(4, 2))
        canvas.create_line(55, 85, 155, 85, 105, 180, 55, 85)

        canvas.pack(fill=BOTH, expand=1)
```

```python
def main():
    root = Tk()
    ex = Example()
    root.geometry("400x250+300+300")
    root.mainloop()


if __name__ == '__main__':
    main()
```

3.Canvas: малюємо кольорові прямокутники в Python

colours.py

```
from tkinter import Tk, Canvas, Frame, BOTH


class Example(Frame):

    def __init__(self):
        super().__init__()
        self.initUI()

    def initUI(self):
        self.master.title("Цвета")
        self.pack(fill=BOTH, expand=1)

        canvas = Canvas(self)
        canvas.create_rectangle(
            30, 10, 120, 80,
            outline="#fb0", fill="#fb0"
        )

        canvas.create_rectangle(
            150, 10, 240, 80,
            outline="#f50", fill="#f50"
        )

        canvas.create_rectangle(
            270, 10, 370, 80,
            outline="#05f", fill="#05f"
        )

        canvas.pack(fill=BOTH, expand=1)


def main():
    root = Tk()
    ex = Example()
    root.geometry("400x100+300+300")
    root.mainloop()


if __name__ == '__main__':
    main()
```

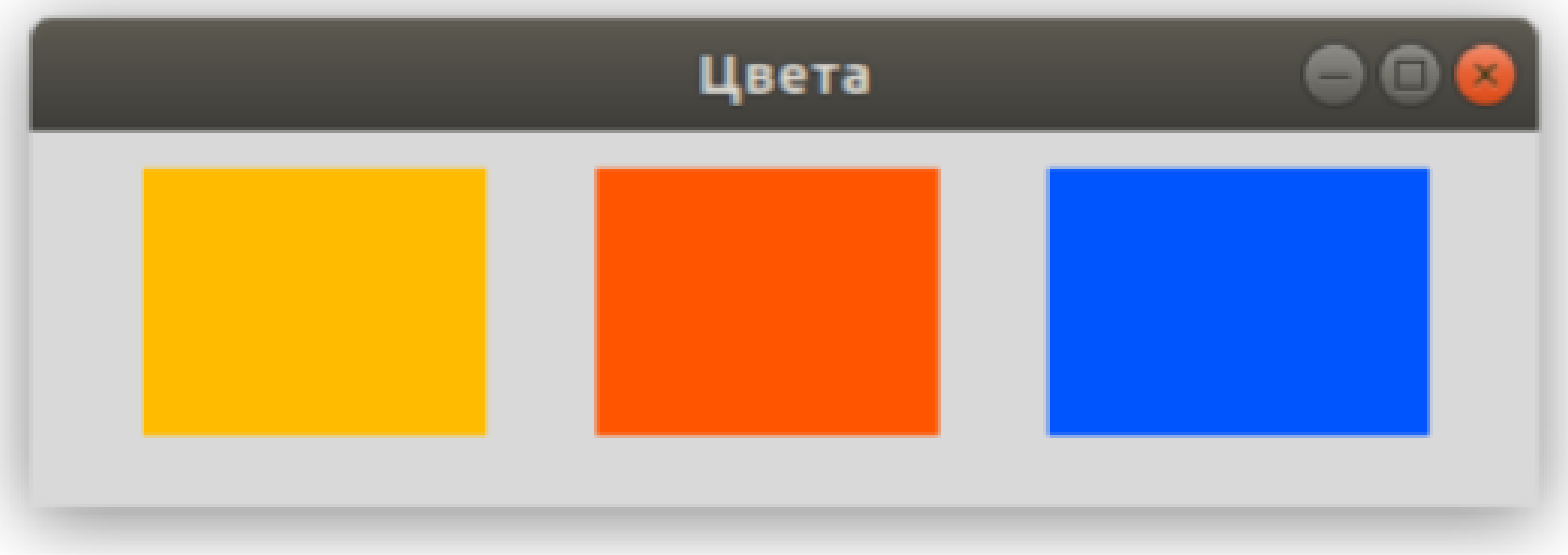
Так створюється віджет **Canvas:**

```python
canvas = Canvas(self)
```

Далі створюється прямокутник, для чого використовується create_rectangle().
Тут потрібно прописати 4 параметри:

- x;
- y;
- Координатне положення верхньої лівої точка(лівого верхнього кута)
- Координатне положення нижньої правої точки(нижнього правого кута)

Готовий результат:



@ZTUEDUUA

ЖИТОМИРСЬКА ПОЛІТЕХНІКА

ДЕРЖАВНИЙ УНІВЕРСИТЕТ

- **Розвиваємо лідерів**
- **Створюємо інновації**
- **Змінюємо світ на краще**