

20. Grafika

Python nam omogoča, da rezultate programa prikažemo tudi grafično. Za grafiko je na voljo več modulov. Ogleдали si bomo dva.

20.1. Modul turtle

Modul turtle, ki omogoča delo z želvjo grafiko smo že spoznali. Spomnimo se, kako lahko shranimo sliko z zaslona na datoteko.

```
zaslon = getscreen()
zaslon.getcanvas().postscript(file=f)
bye()
```

Zdaj pa še drugi, veliko bolj pomemben in zmogljiv grafični vmesnik.

20.2. Modul tkinter

V ta namen moramo odpreti grafično okno, ter vanj postaviti platno, na katerega lahko rišemo. Delamo s modulom tkinter, ki je objektno orientiran.

20.3. Okno s platnom

```
from tkinter import *
window = Tk()
window.title("Okno s platnom.")
canvas = Canvas(window, width=300, height=300)
canvas.pack()
# sem nalagamo grafične elemente
window.mainloop()
```

20.4. Osnovni elementi

Sliko sestavimo iz osnovnih grafičnih elementov (črta, pravokotnik, elipsa, poligon, lok, besedilo, ...), ki jim lahko nastavimo različne lastnosti (barva, rob, debelina, ...). Pregled likov in njihovih lastnosti dobimo [tukaj](#).

- **canvas.create_arc(x0, y0, x1, y1, opcije...)** ... nariše krožni lok/odsek/izsek
 - start
 - extent
 - fill
 - outline
 - width
 - style (PIESLICE, CHORD, ARC)
- **canvas.create_line(x0, y0, x1, y1, ..., opcije...)** ... nariše lomljeno črto
- **canvas.create_rectangle(x0, y0, x1, y1, opcije...)** ... nariše pravokotnik
- **canvas.create_oval(x0, y0, x1, y1, opcije...)** ... nariše elipso
- **canvas.create_polygon(x0, y0, x1, y1, ..., opcije...)** ... nariše poligon
- **canvas.create_text(x0, y0, opcije...)** ... nariše besedilo

20.5. Primer

Sestavite funkcijo, ki nariše graf funkcije f na intervalu (a,b) .

```
from tkinter import *

def graf(f, a, b):
    w = 600
    h = 400
    window = Tk()
    canvas = Canvas(window, width=w, height=h)
    canvas.pack()

    xmin = 10
    xmax = w - 10
    k = (xmax - xmin) / (b - a)
    n = (b * xmin - a * xmax) / (b - a)
    x0 = n
    y0 = h / 2

    s = []
    x = a
    while x <= b:
        s.append((k * x + n, y0 - k * f(x)))
        x += 1 / k

    canvas.create_line(0, y0, w, y0, arrow=LAST)
    canvas.create_line(x0, 0, x0, h, arrow=FIRST)
    canvas.create_line(x0 + k, y0 - 5, x0 + k, y0 + 5)
    canvas.create_line(x0 - 5, y0 - k, x0 + 5, y0 - k)
    canvas.create_text(x0 - 7, y0 - k, text='1', anchor=E)
    canvas.create_text(x0 + k, y0 + 7, text='1', anchor=N)
    canvas.create_line(s, fill='red')
    window.mainloop()

# graf(sin, -pi, pi)
# graf(exp, -10, 2)
# graf(tan, -5, 5)
graf(lambda x: sin(10 * x + 2), -2 * pi, pi)
```

20.6. Primer

Sestavite funkcijo, ki nariše nekaj osnovnih grafičnih elementov.

```
def risanje():
    window = Tk()
    canvas = Canvas(window, width=400, height=300)
    canvas.pack()

    canvas.create_rectangle(10, 50, 150, 170)
    canvas.create_oval(10, 50, 150, 170)
    canvas.create_line(10, 50, 150, 170, 250, 100, fill='blue',
                      width=4, smooth=True)
    canvas.create_arc(100, 100, 300, 200, style=CHORD,
                      fill='yellow', outline='blue', start=180, extent=180)
    canvas.create_polygon(10, 50, 150, 170, 250, 100)
    canvas.create_text(200, 150, text='besedilo', anchor=NW,
                      fill='red', font=('Arial', 16, 'bold italic'))

    window.mainloop()
```

20.7. Primer

Oglejte si funkcijo, ki nariše pujsa.

```
def pujs():
    window = Tk()
    canvas = Canvas(window, width=400, height=400, background='white')
    canvas.pack()
    # nogi (nogi narišemo pred telesom, ker želimo da jih na sliki telo
    prekriva)
    canvas.create_rectangle(140, 280, 160, 350, fill='pink',
    outline='black', width=2)
    canvas.create_rectangle(240, 280, 260, 350, fill='pink',
    outline='black', width=2)
    # telo
    canvas.create_oval(100, 100, 300, 300, fill='pink', outline='black',
    width=2)
    # ocesi
    canvas.create_oval(140, 150, 170, 170, width=2)
    canvas.create_oval(230, 150, 260, 170, width=2)
    canvas.create_oval(150, 160, 160, 170, width=2, fill='black')
    canvas.create_oval(240, 160, 250, 170, width=2, fill='black')
    # rilec
    canvas.create_oval(160, 190, 240, 250, width=2)
    canvas.create_oval(180, 210, 187, 217, width=2)
    canvas.create_oval(213, 210, 220, 217, width=2)
    canvas.create_arc(180, 200, 220, 240, start=200, extent=140,
    style='arc', outline='brown', width=5, dash=1)
    # usesi
    canvas.create_arc(50, 120, 150, 280, start=90, extent=120, width=2,
    fill='pink')
    canvas.create_arc(250, 120, 350, 280, start=-30, extent=120, width=2,
    fill='pink')
    # obrvi
    canvas.create_line(140, 140, 170, 140, width=2)
    canvas.create_line(230, 140, 260, 140, width=2)
    # rep
    canvas.create_line(200, 100, 180, 60, 200, 40, 220, 60, 200, 70, 170,
    30, width=4)
    # korito
    canvas.create_polygon(50, 330, 70, 400, 330, 400, 350, 330,
    fill='brown', outline='black', width=4)
    # napis
    canvas.create_text((200, 15), text='To je pujsek pri koritu.',
    justify='center', font=('Arial', 12, 'bold italic'))
    window.mainloop()
    # pujs()
```

20.8. Naloge

1. Sestavite funkcijo, ki nariše n dotikajočih se krogov s središči na navidezni krožnici polmera r .
2. Sestavite funkcijo, pri prikazu šest kock z 1,2,3,4,5,6 pikami.
3. S pomočjo dokumentacije raziščite delovanje metode `create_bitmap` in jo vključite na platno, tako da narišete vrsto desetih vprašajev.