1. uzdevums

Nodrukāt secīgi skaitļus no 1 līdz 15 pēc kārtas un katru trešo.

Kods:

```
# Programmas nosaukums: 1. uzd MPR8

# 1. uzdevums MPR8

# Uzdevuma formulējums: Nodrukāt secīgi skaitļus no 1 līdz 15 pēc kārtas un katru trešo.

# Versija 1.0


print("Programma nodrukā secīgi skaitļus no 1 līdz 15 pēc kārtas un katru trešo.\n")

for x in range(1, 16, 3):
    print(x)
```

Testa piemēri:

1)

```
Programma nodrukā secīgi skaitļus no 1 līdz 15 pēc kārtas un katru trešo.

1
4
7
10
13
```

2. uzdevums

Nodrukāt secīgi skaitļus no 15 līdz 1 pēc kārtas un katru trešo.

Kods:

```
# Programmas nosaukums: 2. uzd MPR8

# 2. uzdevums MPR8

# Uzdevuma formulējums: Nodrukāt secīgi skaitļus no 15 līdz 1 pēc kārtas un katru trešo.

# Versija 1.0


print("Programma nodrukā secīgi skaitļus no 15 līdz 1 pēc kārtas un katru trešo.\n")
```

```
for x in range(15, 1, -3):
    print(x)
```

Testa piemēri:

1)

```
Programma nodrukā secīgi skaitļus no 15 līdz 1 pēc kārtas un katru trešo.

15
12
9
6
3
```

3. uzdevums

Novilkt taisnes nogriezni.

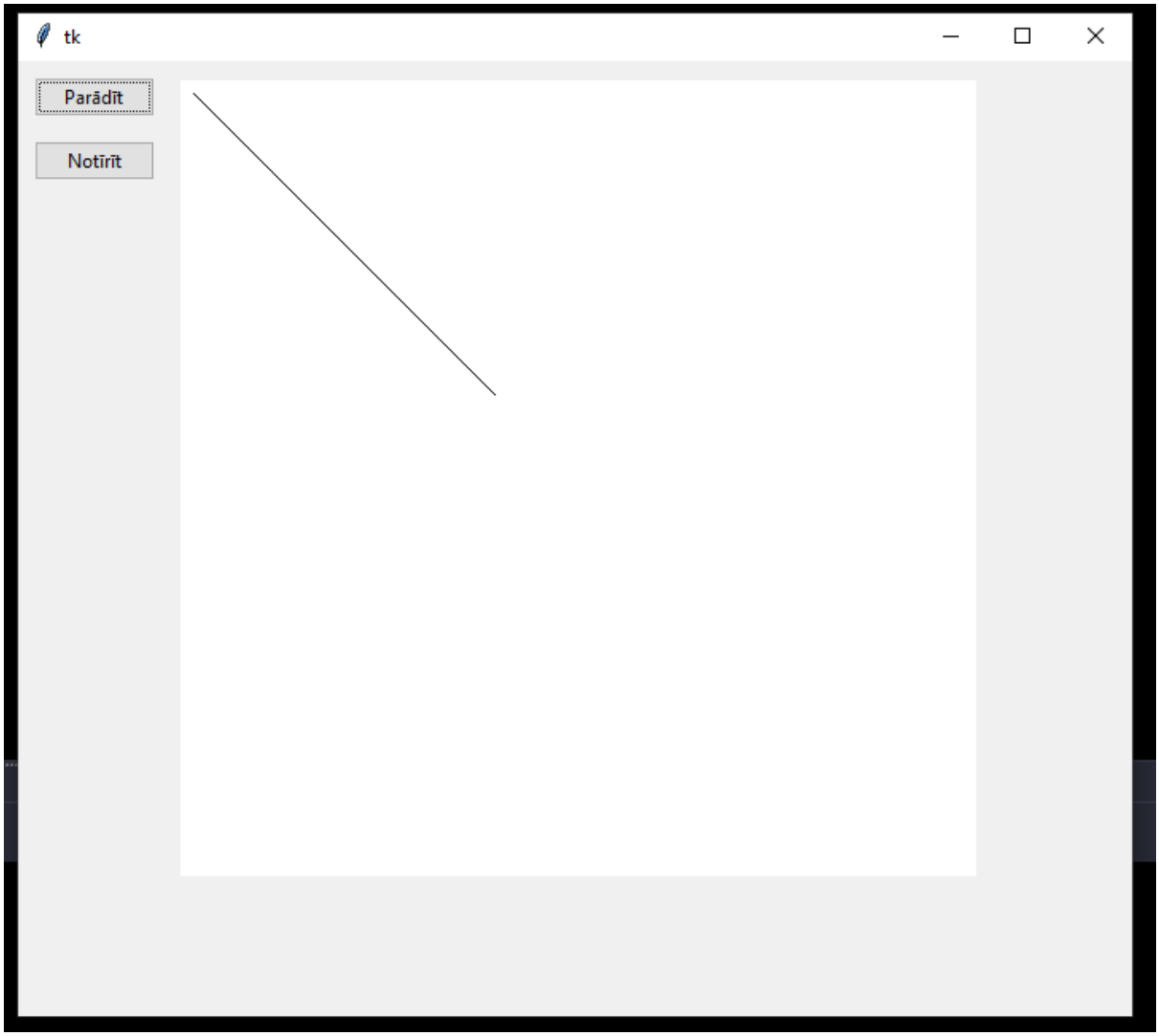
Kods:

```
# Programmas nosaukums: 3. uzd MPR8

# 3. uzdevums MPR8

# Uzdevuma formulējums: Novilkt taisnes nogriezni.

# Versija 1.0


import tkinter

from tkinter import ttk


def paradit():

    kanva.create_line(10,10,200,200)   # zīmēšanai


def notirit():

    kanva.delete("all") # notīrīsanai


logs = tkinter.Tk()

logs.geometry("700x600")
```

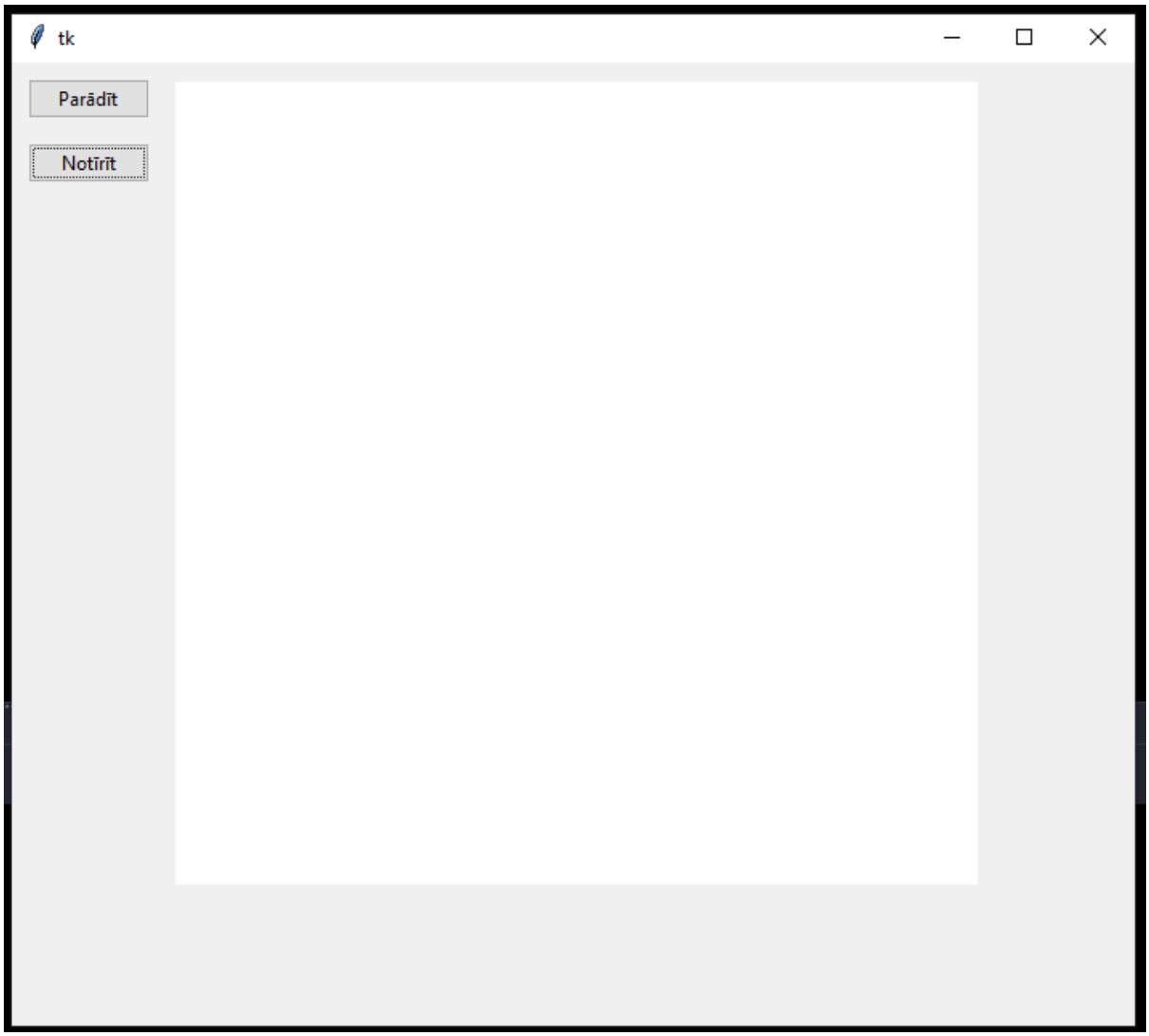
```
kanva = tkinter.Canvas(logs, bg="white", height=500, width=500)
kanva.place(x=100, y=10)
poga1= ttk.Button(logs, text="Parādīt", command=paradit)
poga1.place(x=10,y=10)
poga2 = ttk.Button(logs, text="Notīrīt", command=notirit)
poga2.place(x=10, y=50)

logs.mainloop()
```

Testa piemēri:

1) Parādīšana



2) Notīrīšanā



4. uzdevums

Uzzīmēt žogu visu vienā krāsā.

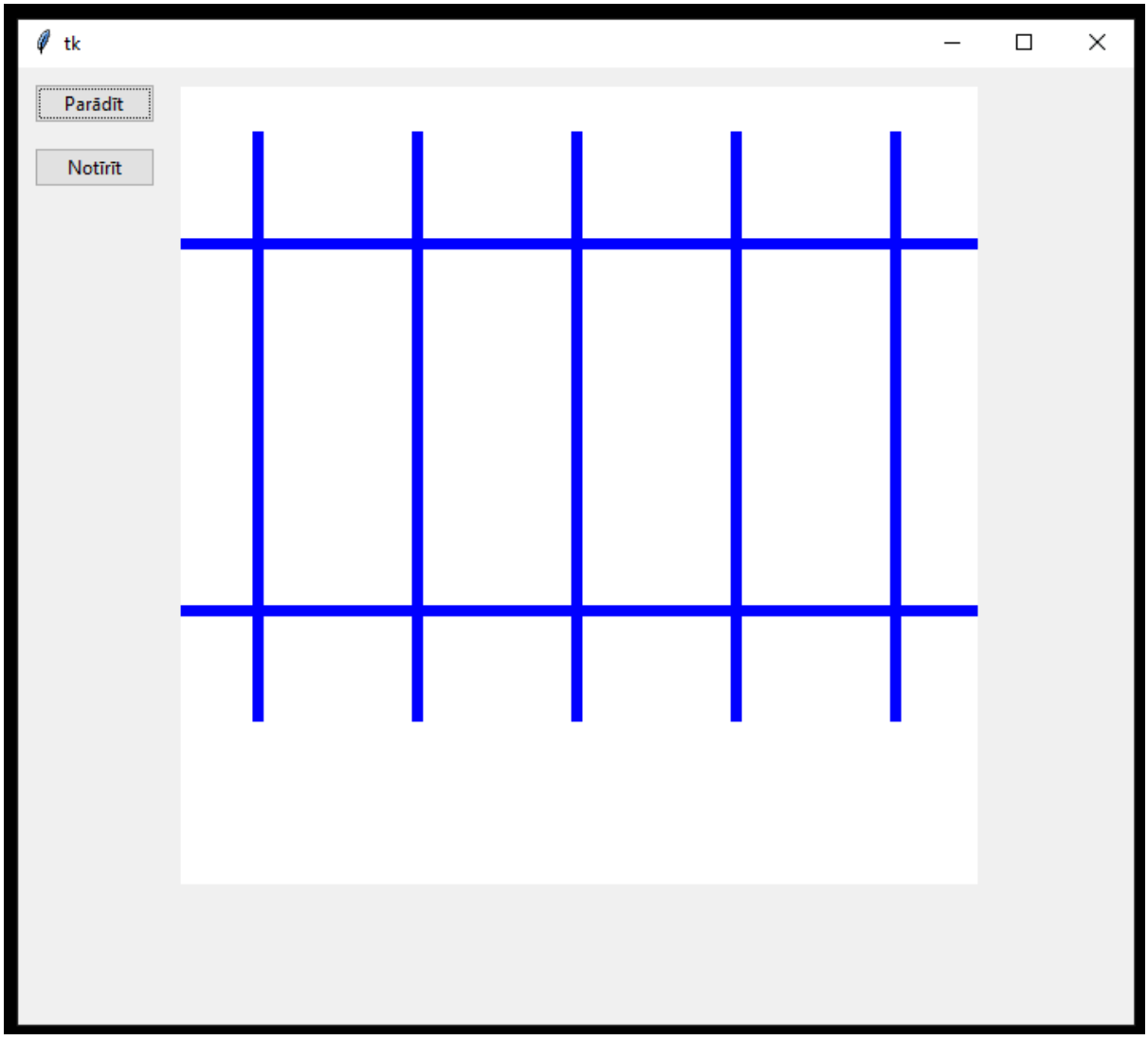
Kods:

```
# Programmas nosaukums: 4. uzd MPR8

# 4. uzdevums MPR8

# Uzdevuma formulējums: Uzzīmēt žogu visu vienā krāsā.

# Versija 1.0


import tkinter

from tkinter import ttk
```

```
def zogs():

    for x in range (50, 500, 100):
        kanva.create_line(x, 30, x, 400, width=7, fill="blue")

    for y in range (100, 400, 230):
        kanva.create_line(0, y, 700, y, width=7, fill="blue")

def notirit():
    kanva.delete("all") # notirisanai

logs = tkinter.Tk()
logs.geometry("700x600")

kanva = tkinter.Canvas(logs, bg="white", height=500, width=500)
kanva.place(x=100, y=10)
poga1= ttk.Button(logs, text="Parādīt", command=zogs)
poga1.place(x=10,y=10)
poga2 = ttk.Button(logs, text="Notīrīt", command=notirit)
poga2.place(x=10, y=50)

logs.mainloop()
```

Testa piemēri:

1) Parādīšana



4. uzdevums PU1

Uzzīmēt žogu ar latiņām katru savā krāsā.

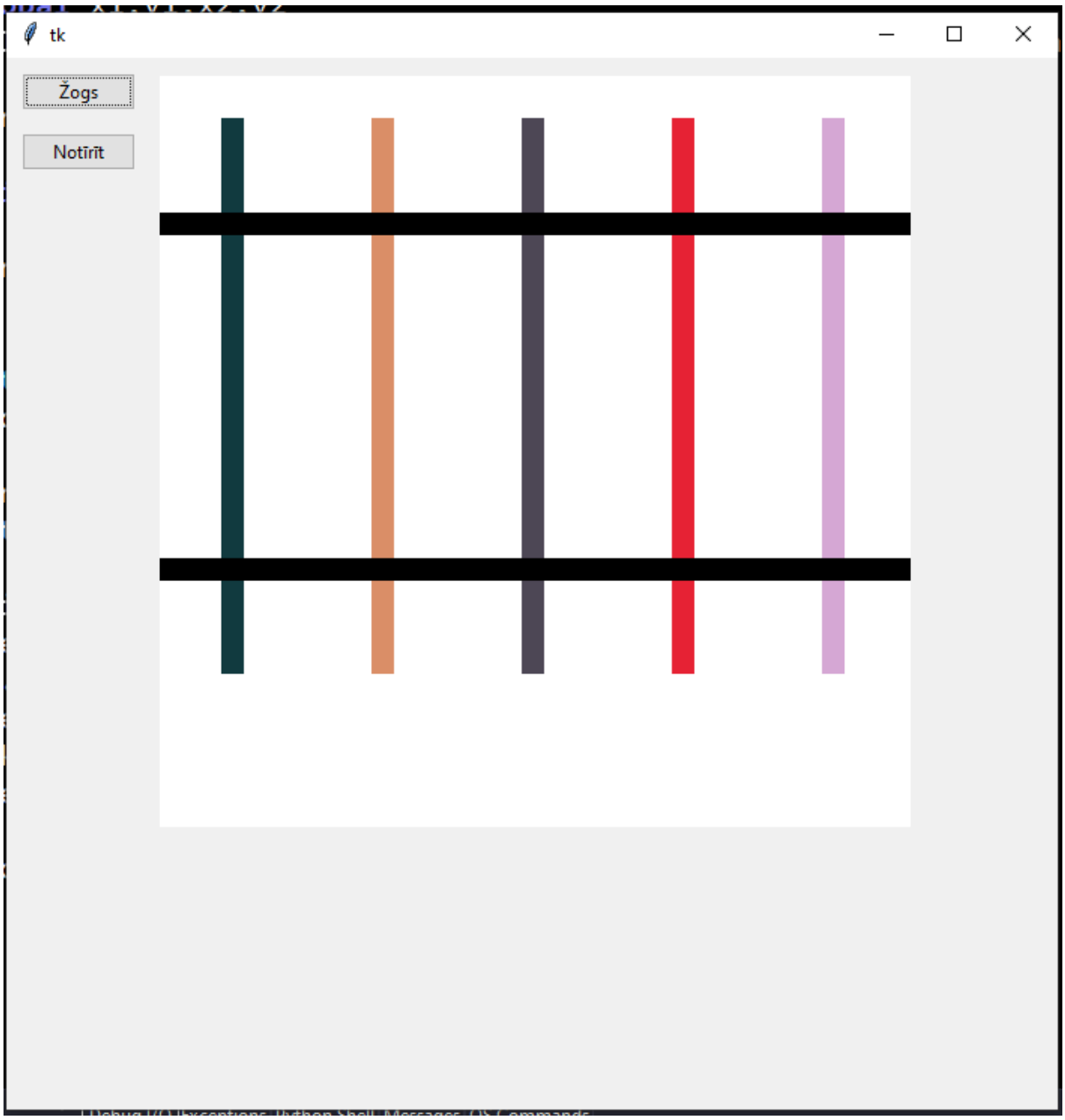
Kods:

```
# Programmas nosaukums: 4. uzd MPR8 PU

# 4. uzdevums MPR8 PU

# Uzdevuma formulējums: Uzzīmēt žogu ar latiņām katru savā krāsā.

# Versija 1.0


import tkinter

from tkinter import ttk
```

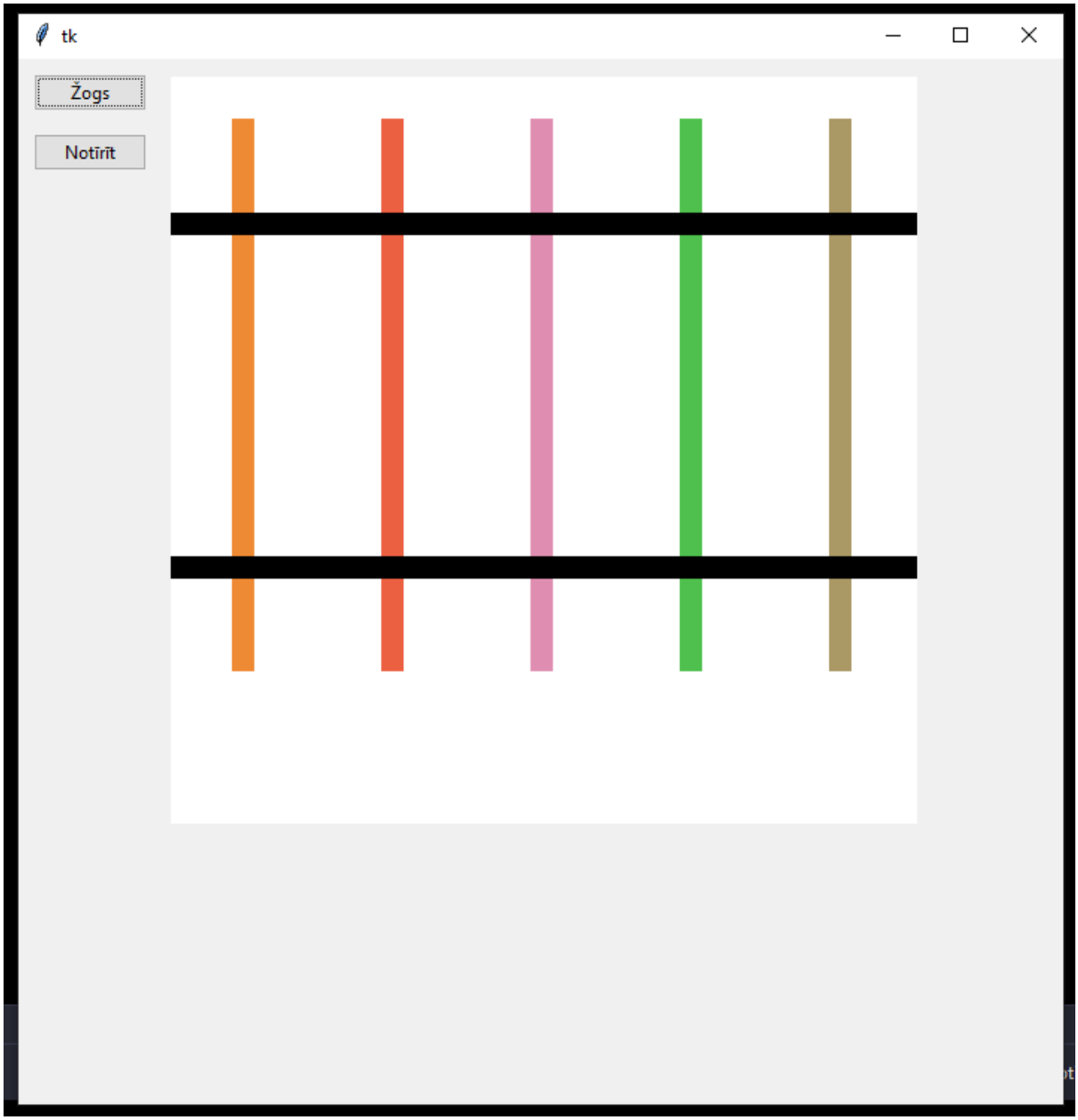
```
from random import randrange


def zogs():

    for x in range (50, 500, 100):
        global x1,y1,x2,y2
        color_c='#%02x%02x%02x' % (randrange(256), randrange(256), randrange(256)) # random rgb colour => HEX

        kanva.create_line(x, 30, x, 400, width=15, fill=color_c) # fill ar random color (vertikalas latinas)

    for y in range (100, 400, 230):

        kanva.create_line(0, y, 700, y, width=15, fill="black") # melnas horizontālas latiņas


def notirit():
    kanva.delete("all") # notirisanai

logs = tkinter.Tk()
logs.geometry("700x700")

kanva = tkinter.Canvas(logs, bg="white", height=500, width=500)
kanva.place(x=100, y=10)
poga1= ttk.Button(logs, text="Žogs", command=zogs)
poga1.place(x=10,y=10)
poga2 = ttk.Button(logs, text="Notīrīt", command=notirit)
poga2.place(x=10, y=50)

logs.mainloop()
```

Testa piemēri:

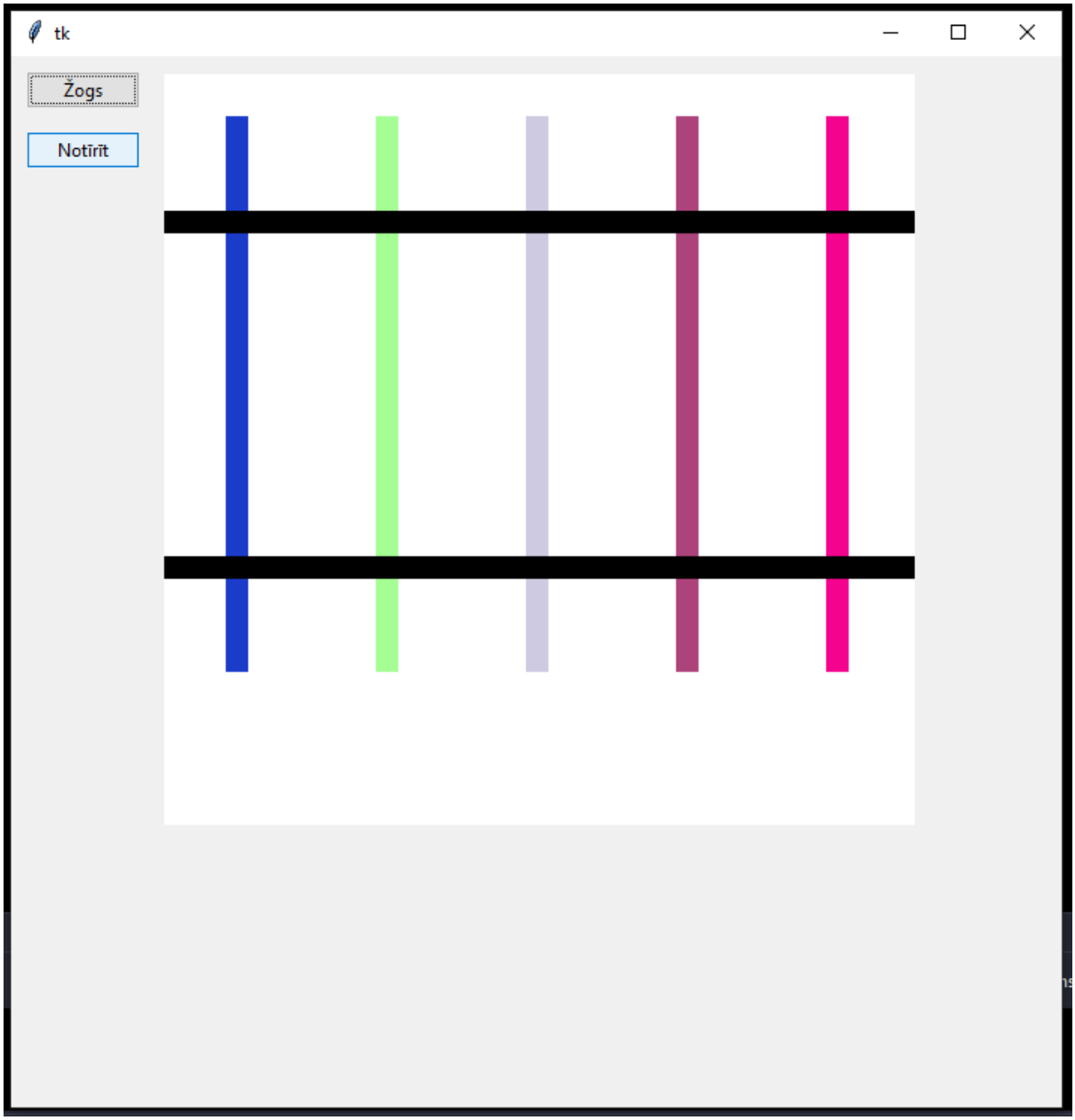
1)



2)



3)



5. uzdevums

Uzzīmēt žogu visu vienā krāsā.

Kods:

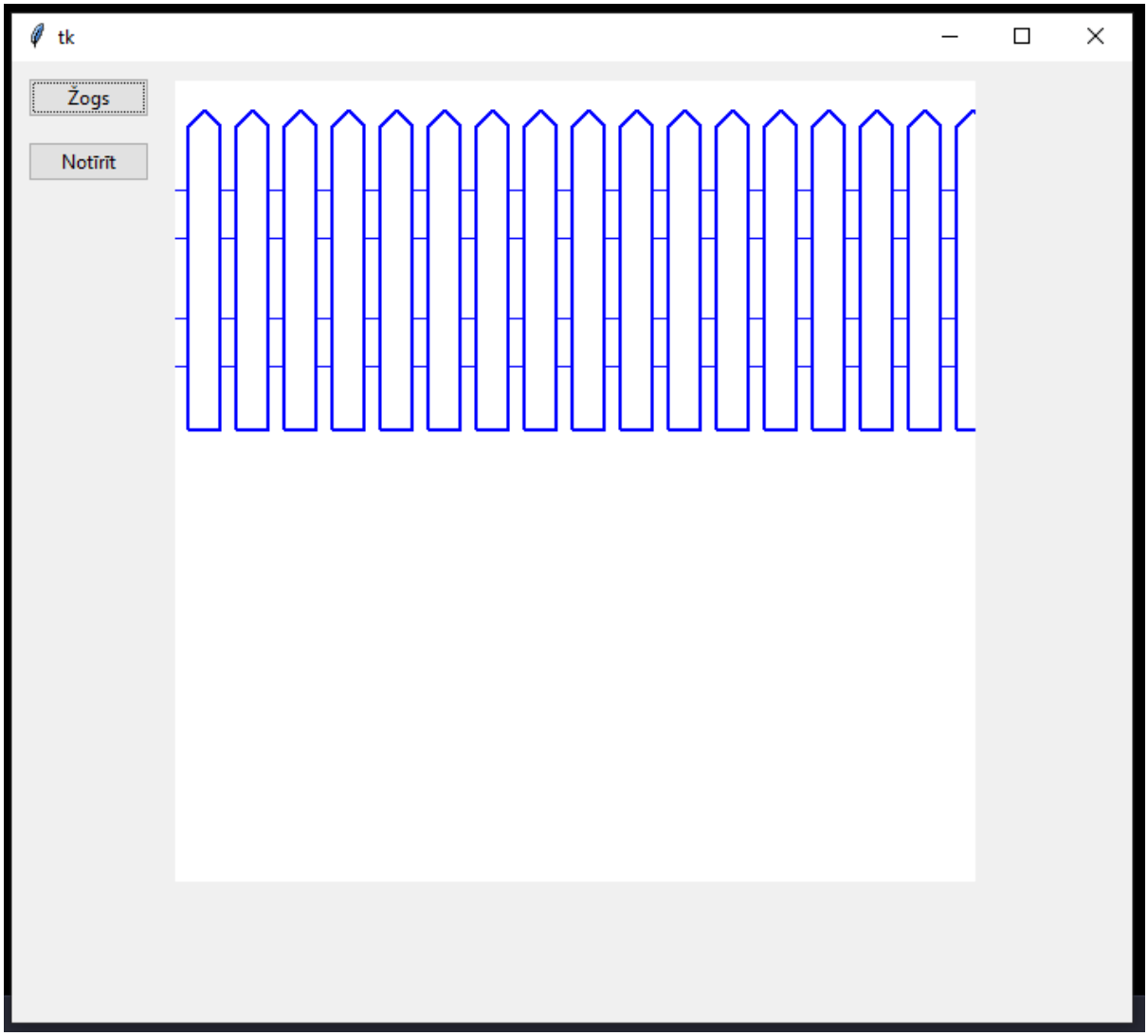
```
# Programmas nosaukums: 5. uzd MPR8

# 5. uzdevums MPR8

# Uzdevuma formulējums: Uzzīmēt žogu visu vienā krāsā.

# Versija 1.0
```

```
import tkinter
from tkinter import ttk

def zogs():

    x=20 # sakotnejas vertibas
    y=20

    for x in range (20, 550, 30):

        kanva.create_line(x,y, x-10, y+10, x-10, y+200, x+10, y+200, x+10, y+10, x, y, width=2, fill="blue")
# latiņas ar ciklu

# ------------------------------------------------------------

    kanva.create_line(10,70, 0, 70, width=1, fill="blue")
    kanva.create_line(10,100, 0, 100, width=1, fill="blue") # pirmas četras usiņas

    kanva.create_line(10,150, 0, 150, width=1, fill="blue")
    kanva.create_line(10,180, 0, 180, width=1, fill="blue")


    for x in range (1, 20, 1):
        kanva.create_line(10+30*x, 70, 0+30*x, 70, width=1, fill="blue")
        kanva.create_line(10+30*x, 100, 0+30*x, 100, width=1, fill="blue") # usiņas ar ciklu

        kanva.create_line(10+30*x, 150, 0+30*x, 150, width=1, fill="blue")
        kanva.create_line(10+30*x, 180, 0+30*x, 180, width=1, fill="blue")
```

```
def notirit():
    kanva.delete("all") # notirisanai


logs = tkinter.Tk()
logs.geometry("700x600")


kanva = tkinter.Canvas(logs, bg="white", height=500, width=500)
kanva.place(x=100, y=10)
poga1= ttk.Button(logs, text="Žogs", command=zogs)
poga1.place(x=10,y=10)
poga2 = ttk.Button(logs, text="Notīrīt", command=notirit)
poga2.place(x=10, y=50)


logs.mainloop()
```

Testa piemēri:

1) Parādīšana



6. uzdevums

Uzzīmēt rūtiņu lapu. Visi nogriežņi vienā krāsa.

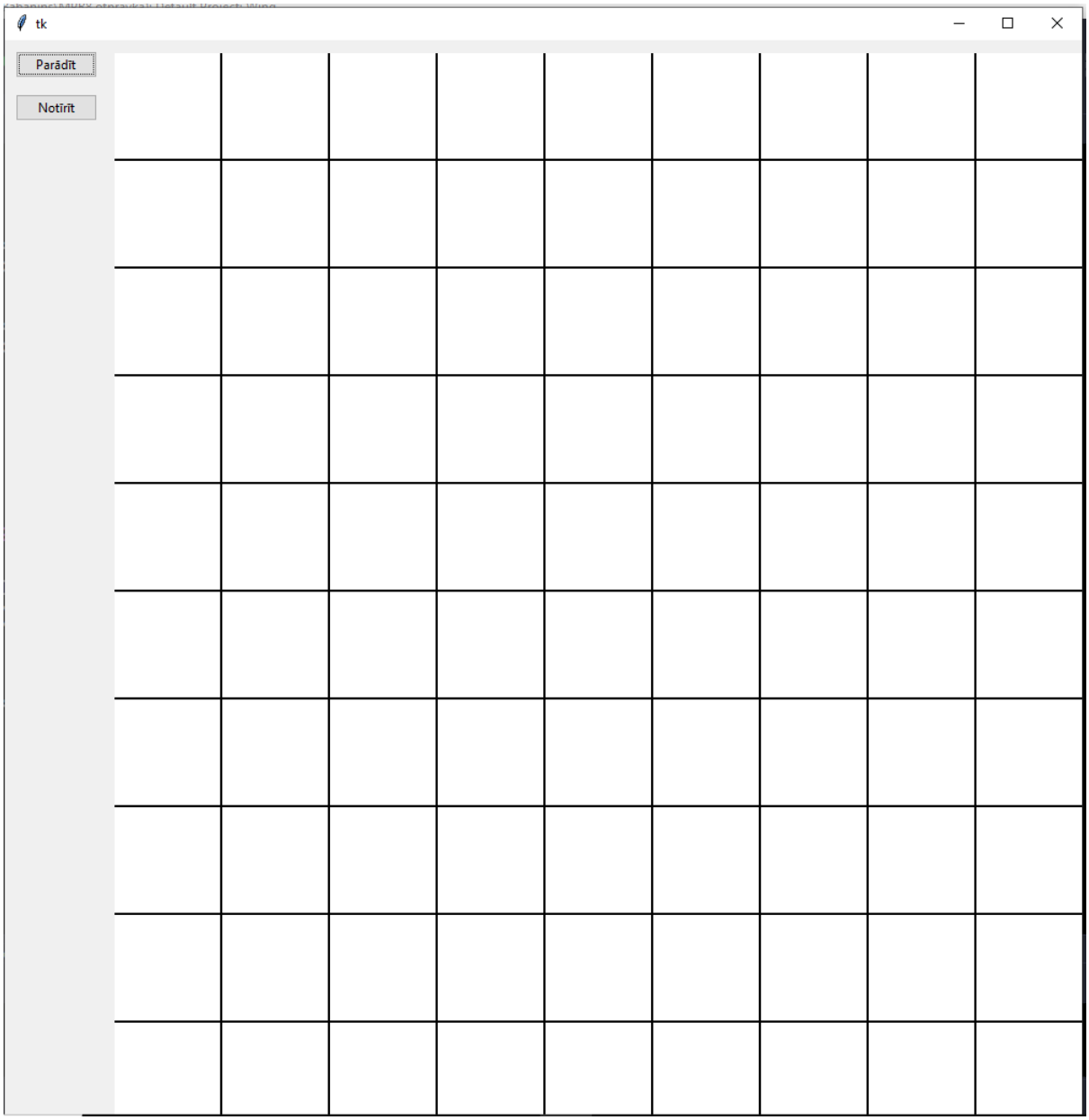
Kods:

```
# Programmas nosaukums: 6. uzd MPR8

# 6. uzdevums MPR8

# Uzdevuma formulējums: Uzzīmēt rūtiņu lapu. Visi nogriežņi vienā krāsā.

# Versija 1.0


import tkinter

from tkinter import ttk


def paradit():
```

```
    for x in range (100, 9000, 100):
        kanva.create_rectangle(x, 0, x, 1000, width=1, fill="blue")

    for y in range (100, 9000, 100):
        kanva.create_rectangle(0, y, 1000, y, width=1, fill="blue")



def notirit():
    kanva.delete("all") # notirisanai


logs = tkinter.Tk()
logs.geometry("1000x1000")


kanva = tkinter.Canvas(logs, bg="white", height=1000, width=1000)
kanva.place(x=100, y=10)
poga1= ttk.Button(logs, text="Parādīt", command=paradit)
poga1.place(x=10,y=10)
poga2 = ttk.Button(logs, text="Notīrīt", command=notirit)
poga2.place(x=10, y=50)


logs.mainloop()
```

Testa piemēri:

1) Parādīšana



7. uzdevums

Uzzīmēt saules starus.

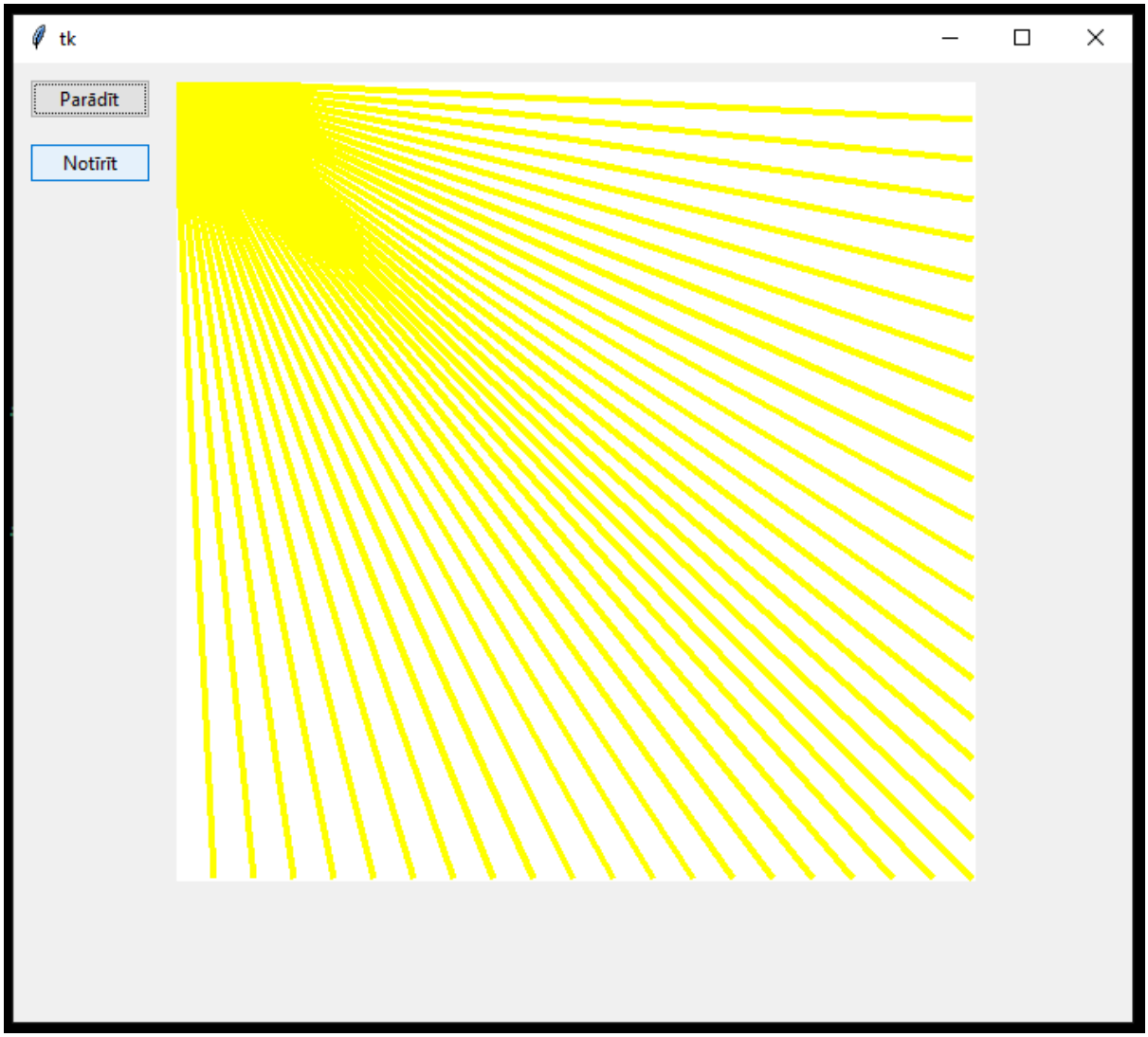
Kods:

```
# Programmas nosaukums: 7. uzd MPR8

# 7. uzdevums MPR8

# Uzdevuma formulējums: Uzzīmēt saules starus.

# Versija 1.0


import tkinter
```

```
from tkinter import ttk


def paradit():


    for x in range (0, 500, 25):
        kanva.create_line(0, 0, x, 500, width=4, fill="yellow") # no 0 līdz 45 grādiem


    for y in range (500, 0, -25):
        kanva.create_line(0, 0, 500, y, width=4, fill="yellow") # no 45 grādiem līdz 90 grāᴏdiem



def notirit():
    kanva.delete("all") # notirisanai


logs = tkinter.Tk()
logs.geometry("700x600")


kanva = tkinter.Canvas(logs, bg="white", height=500, width=500)
kanva.place(x=100, y=10)
poga1= ttk.Button(logs, text="Parādīt", command=paradit)
poga1.place(x=10,y=10)
poga2 = ttk.Button(logs, text="Notīrīt", command=notirit)
poga2.place(x=10, y=50)


logs.mainloop()
```

Testa piemēri:

1) Parādīšana



8. uzdevums

Uzzīmēt koncentriskus taisnstūrus

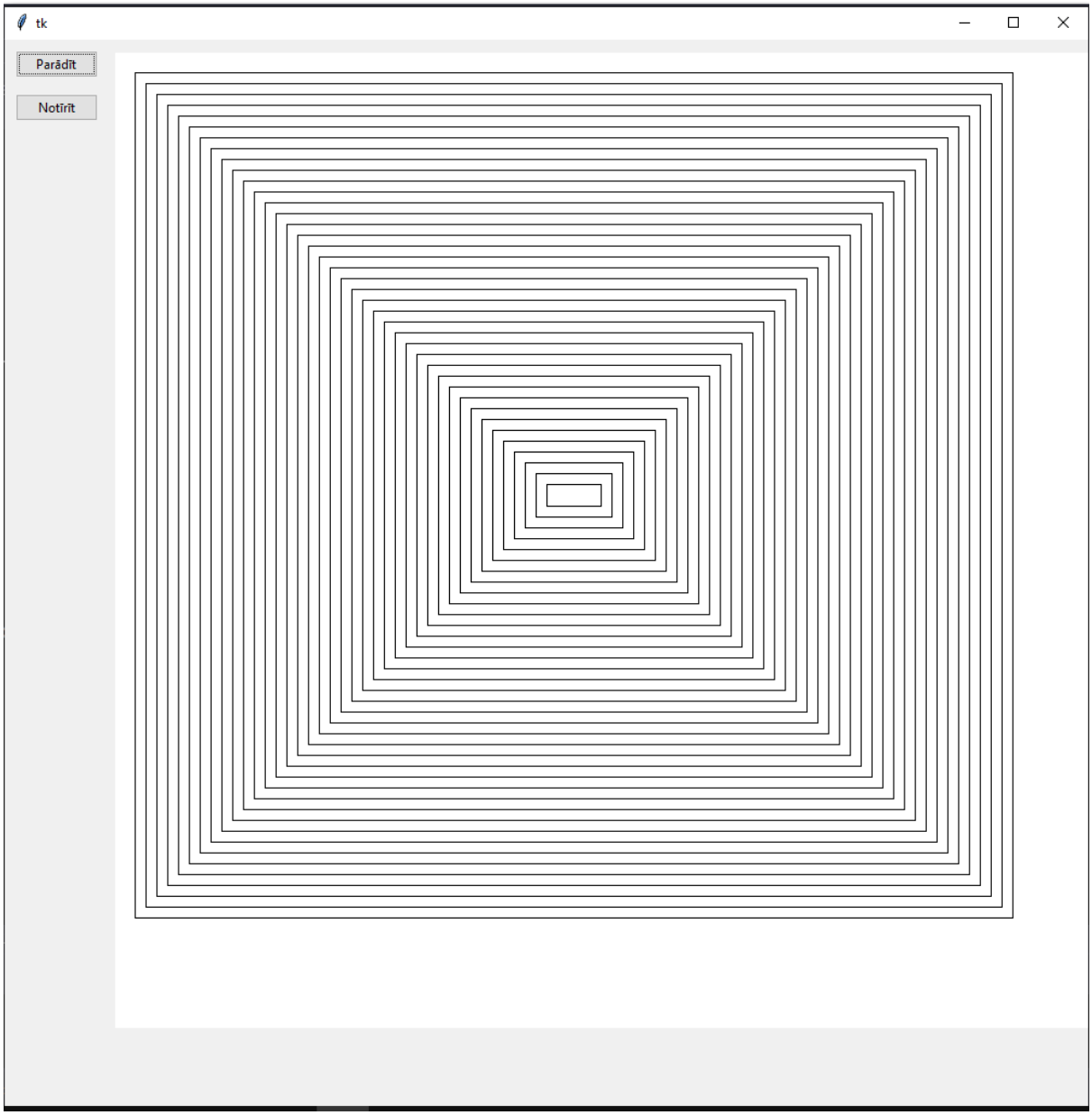
Kods:

```
# Programmas nosaukums: 8. uzd MPR8

# 8. uzdevums MPR8

# Uzdevuma formulējums: Uzzīmēt koncentriskus taisnstūrus

# Versija 1.0


import tkinter

from tkinter import ttk


x1 = 350
```

```
y1 = 300


x2 = 550
y2 = 400

def paradit():

    x1 = 400
    y1 = 400


    x2 = 450
    y2 = 420


    for a in range (0, 390, 10):
        kanva.create_rectangle(x1-a, y1 - a, x2 + a, y2 + a)

def notirit():
    kanva.delete("all") # notirisanai


logs = tkinter.Tk()
logs.geometry("1000x1000")


kanva = tkinter.Canvas(logs, bg="white", height=900, width=900)
kanva.place(x=100, y=10)
poga1= ttk.Button(logs, text="Parādīt", command=paradit)
poga1.place(x=10,y=10)
poga2 = ttk.Button(logs, text="Notīrīt", command=notirit)
poga2.place(x=10, y=50)


logs.mainloop()
```

Testa piemēri:

1) Parādīšana



9. uzdevums

Uzzīmēt koncentriskas riņķa līnijas.

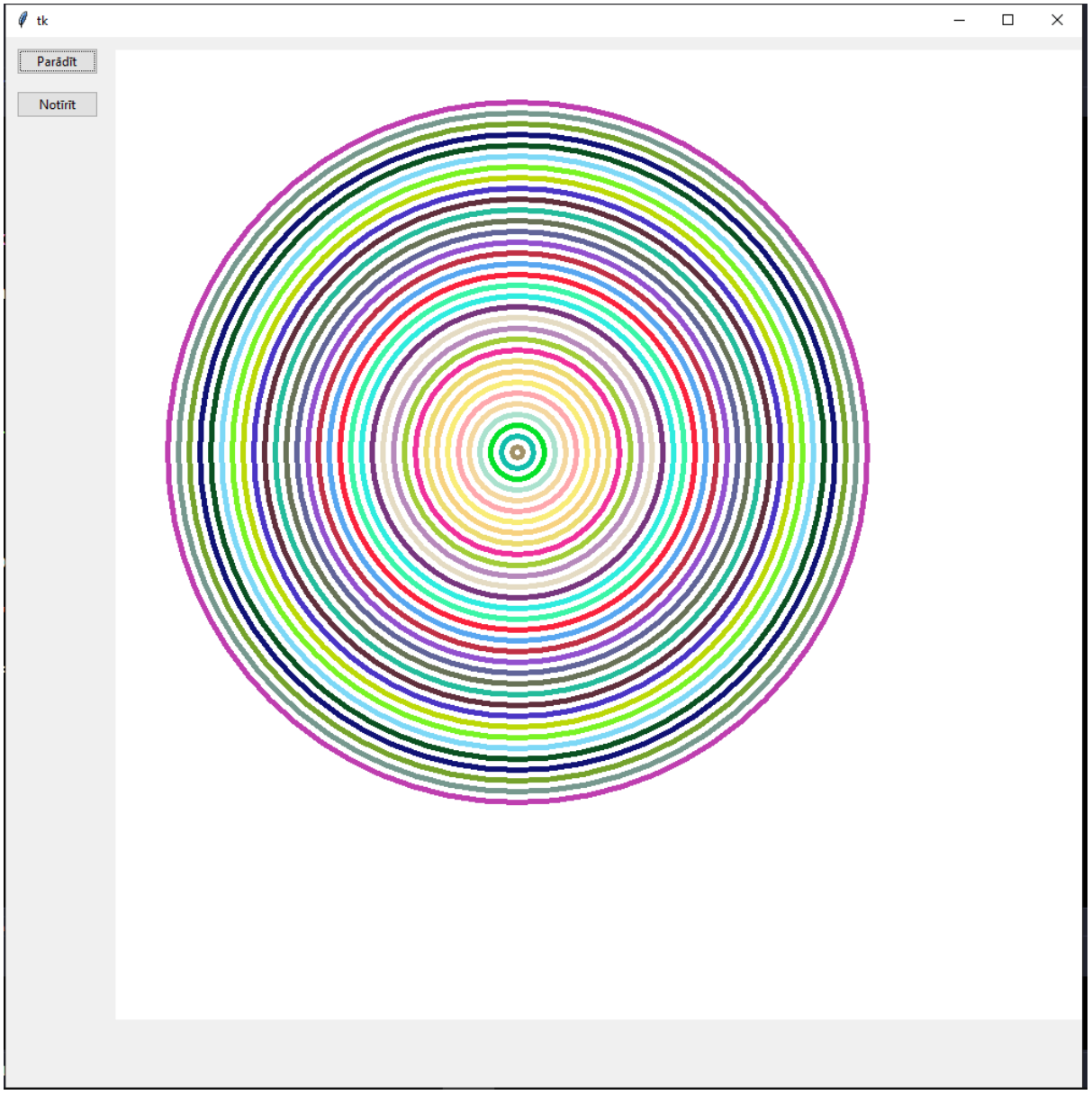
Kods:

```
# Programmas nosaukums: 9. uzd MPR8
# 9. uzdevums MPR8
# Uzdevuma formulējums: Uzzīmēt koncentriskas riņķa līnijas.
# Versija 1.0

import tkinter
from tkinter import ttk
from random import randrange

x1 = 350
y1 = 300

x2 = 550
y2 = 400

def paradit():

    x1 = 400
    y1 = 400

    x2 = 450
    y2 = 450

    for a in range (0, 600, 10):

        x1 = 50
```

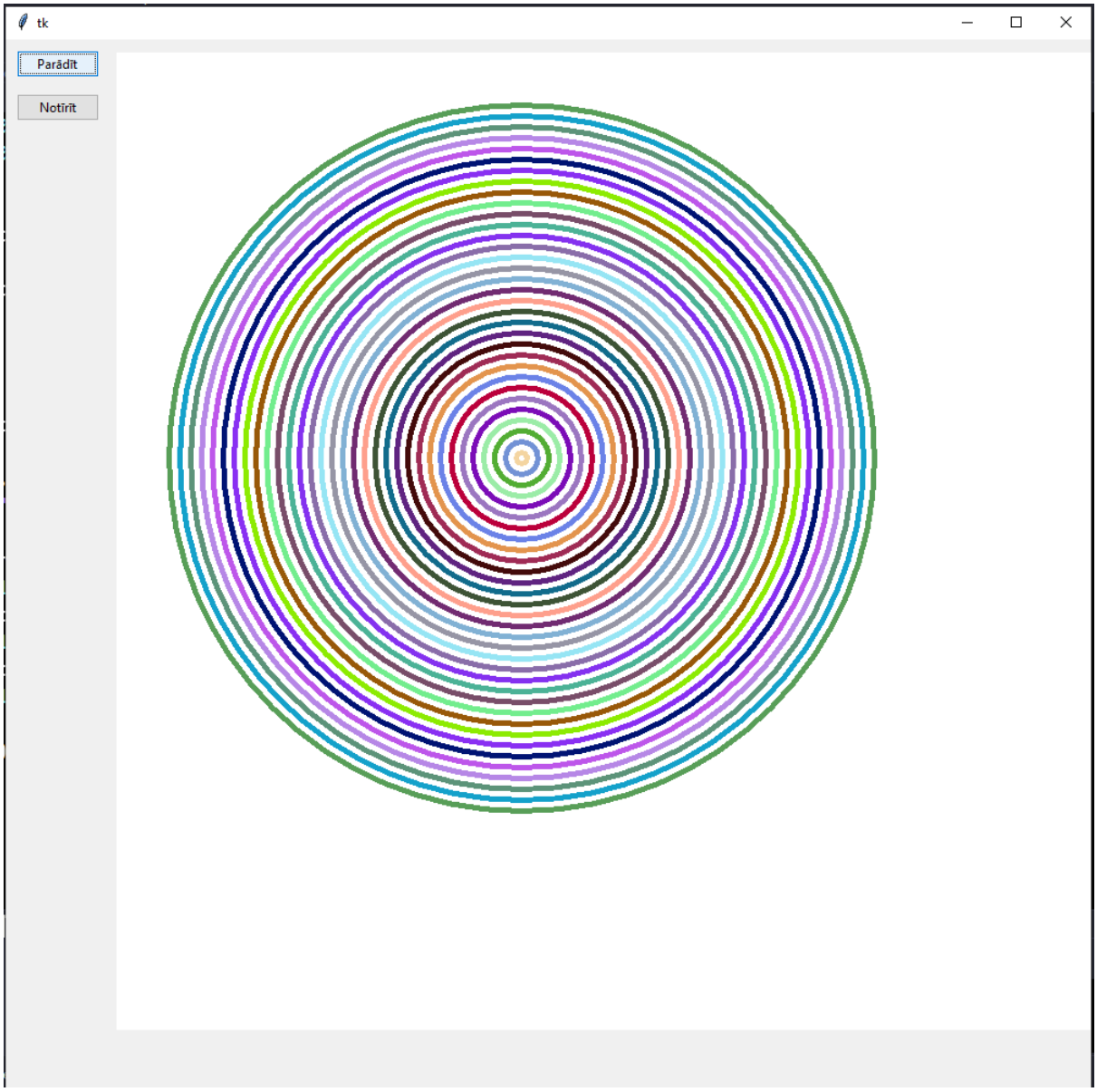
```
        y1 = 50


        x2 = 700
        y2 = 700



        color_c='#%02x%02x%02x' % (randrange(256), randrange(256), randrange(256))


        kanva.create_oval(x1+a, y1 + a, x2 - a, y2 - a,  outline=color_c, width="5")




def notirit():
    kanva.delete("all") # notirisanai


logs = tkinter.Tk()
logs.geometry("1000x1000")


kanva = tkinter.Canvas(logs, bg="white", height=900, width=900)
kanva.place(x=100, y=10)
poga1= ttk.Button(logs, text="Parādīt", command=paradit)
poga1.place(x=10,y=10)
poga2 = ttk.Button(logs, text="Notīrīt", command=notirit)
poga2.place(x=10, y=50)


logs.mainloop()
```

Testa piemēri:

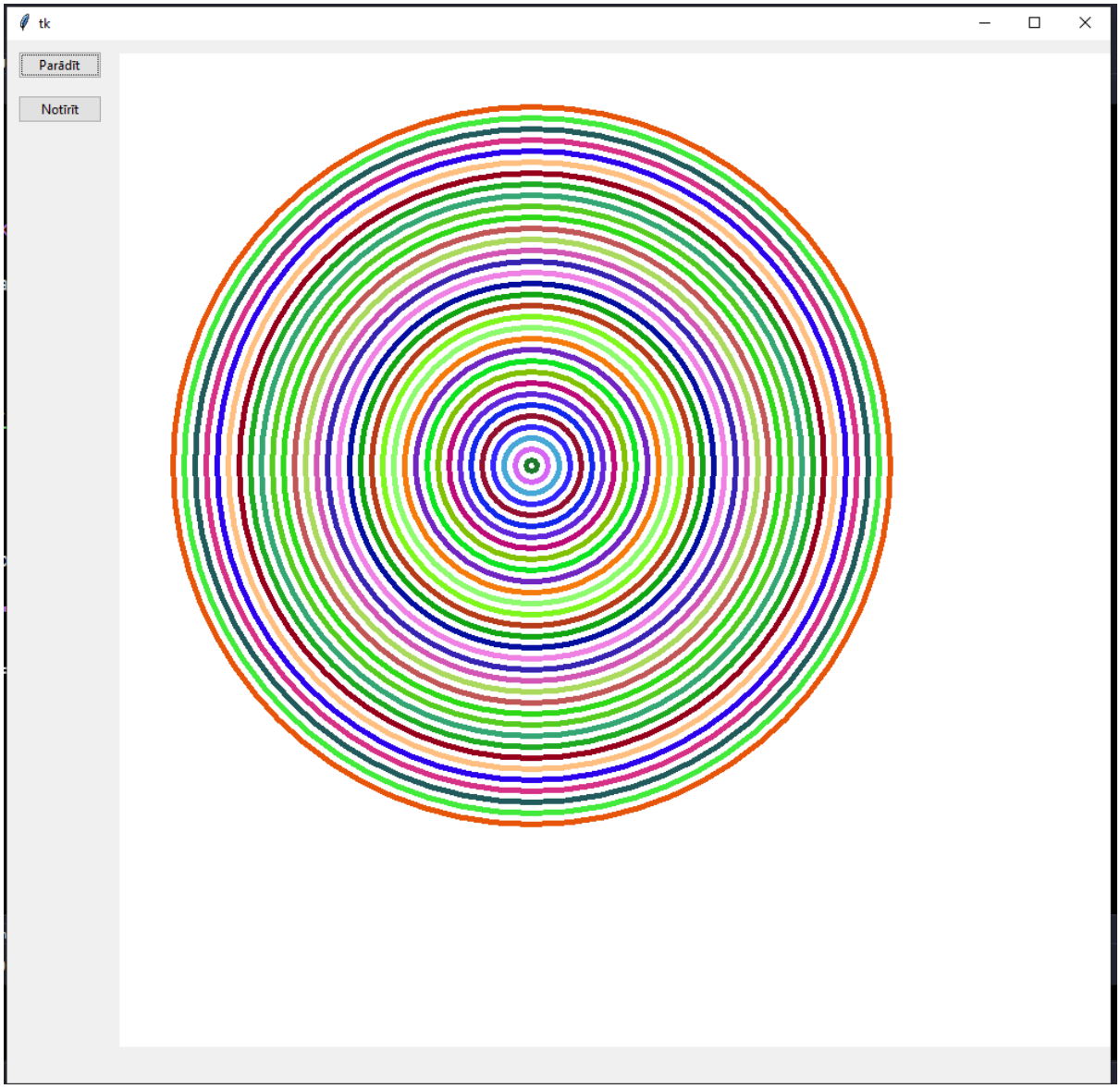
1) Parādīšana



2)



3)



9. uzdevums PU3

Uzzīmēt mērķi.

Kods:

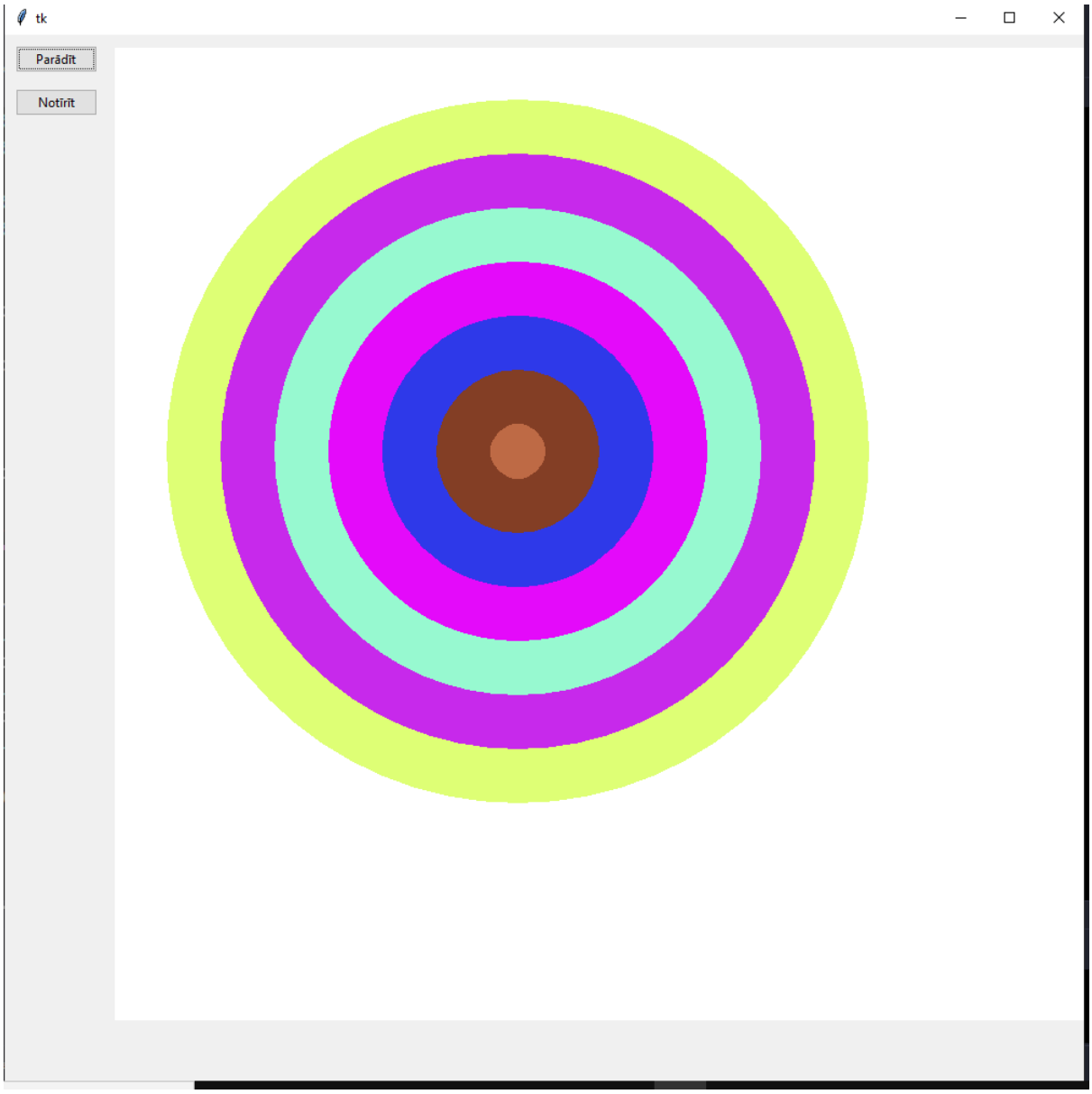
```
# Programmas nosaukums: 9. uzd MPR8 PU3

# 9. uzdevums MPR8 PU3

# Uzdevuma formulējums: Uzzīmēt mērķi.

# Versija 1.0
```

```
import tkinter
from tkinter import ttk
from random import randrange


def paradit():

    for a in range (0, 400, 50):

        x1 = 50
        y1 = 50

        x2 = 700
        y2 = 700


        color_c='#%02x%02x%02x' % (randrange(256), randrange(256), randrange(256))

        kanva.create_oval(x1+a, y1 + a, x2 - a, y2 - a, fill=color_c, outline="")


def notirit():
    kanva.delete("all") # notirisanai

logs = tkinter.Tk()
logs.geometry("1000x1000")

kanva = tkinter.Canvas(logs, bg="white", height=900, width=900)
kanva.place(x=100, y=10)
poga1= ttk.Button(logs, text="Parādīt", command=paradit)
poga1.place(x=10,y=10)
```

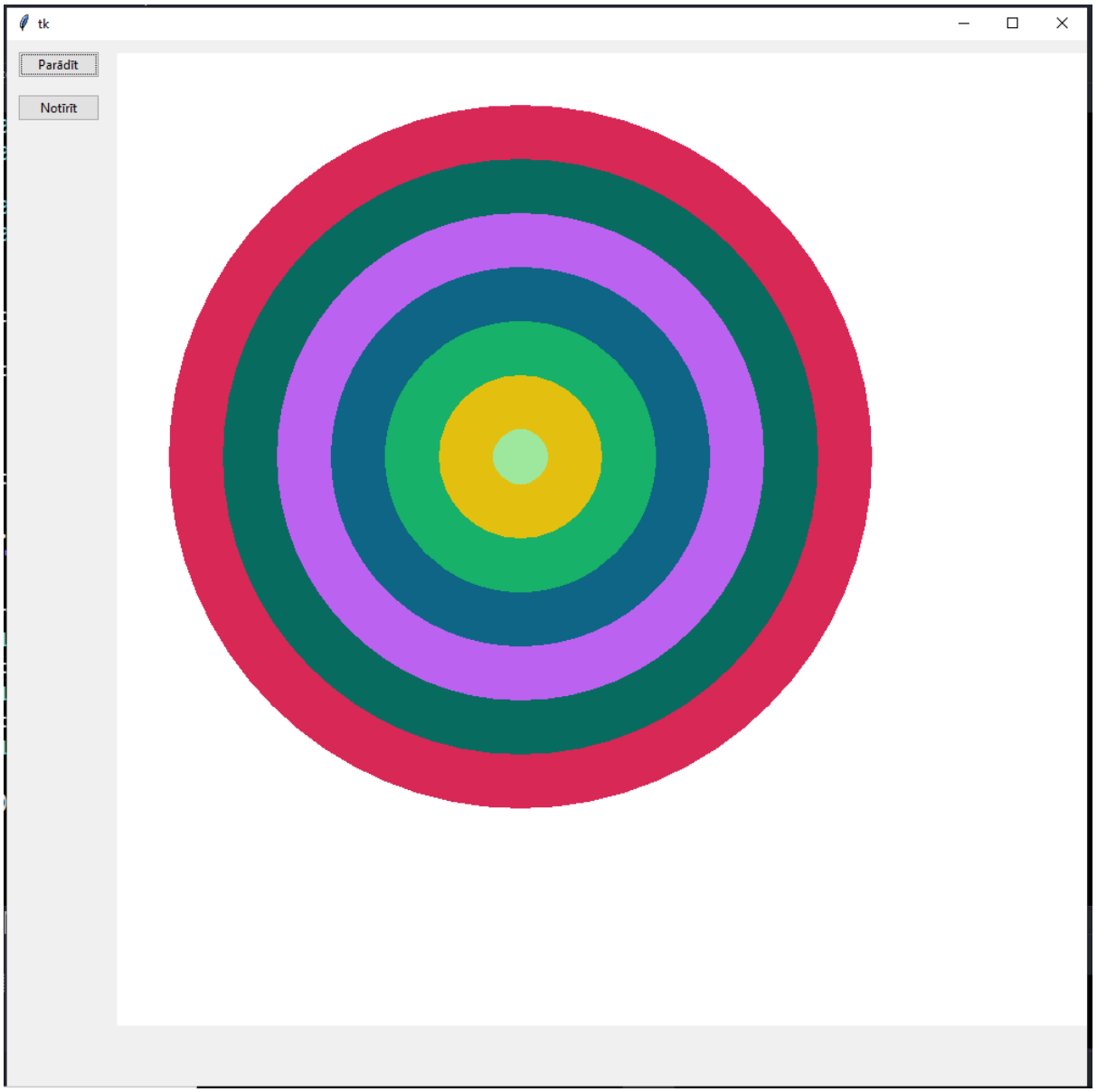
```
poga2 = ttk.Button(logs, text="Notīrīt", command=notirit)
poga2.place(x=10, y=50)

logs.mainloop()
```

Testa piemēri:

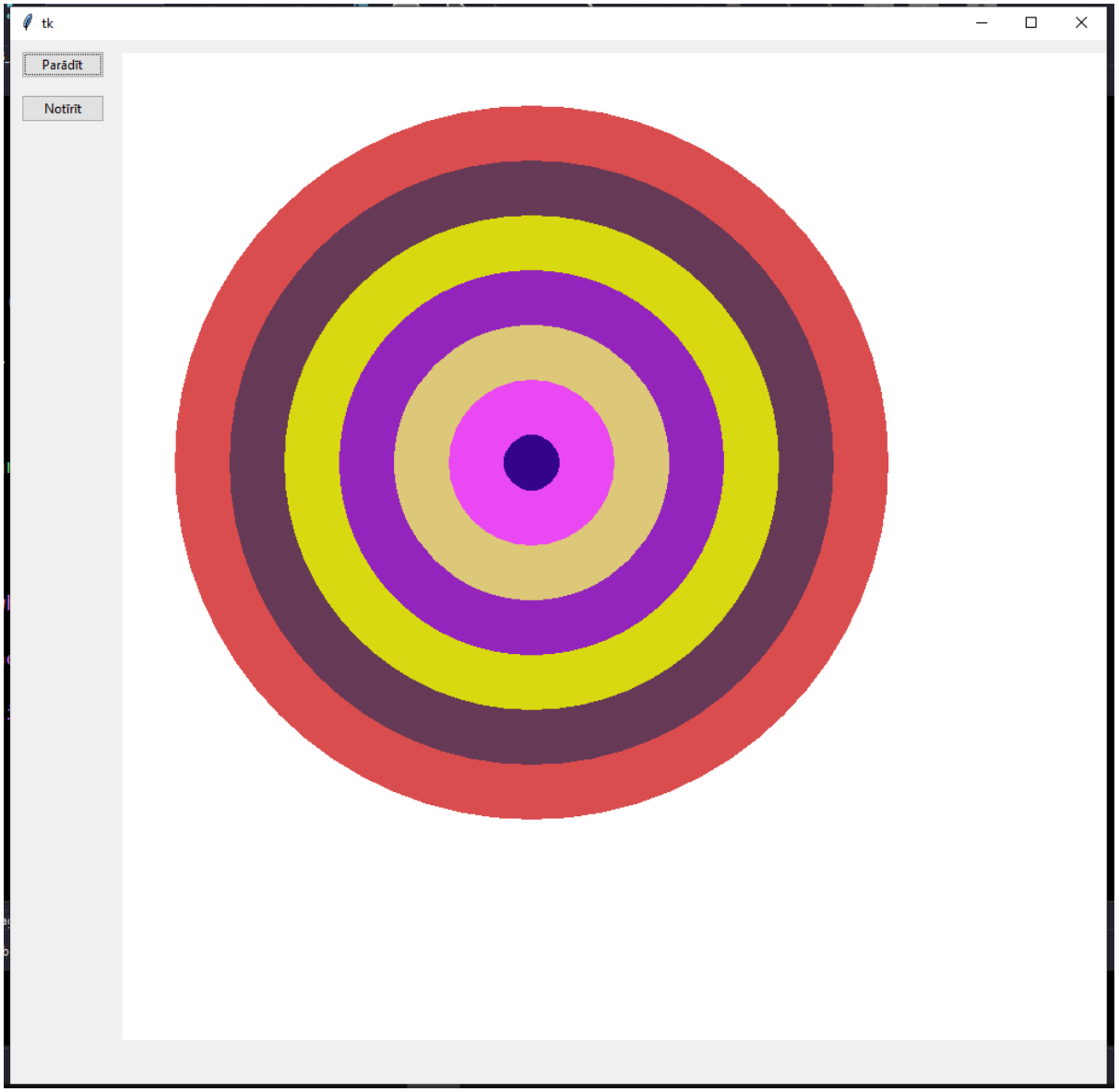
1)



2)



3)


tk
Parādīt
Notīrīt