

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №6

дисциплина: Компьютерный практикум по моделированию

Студент: Журавлев Георгий Иванович

Группа: НФИбД 02-20

МОСКВА

2021__ г.

Цель работы:

1. Изучение Mathplotlib
2. Изучение tkinter
3. Написание игры

Список сокращений: -

Список обозначений: -

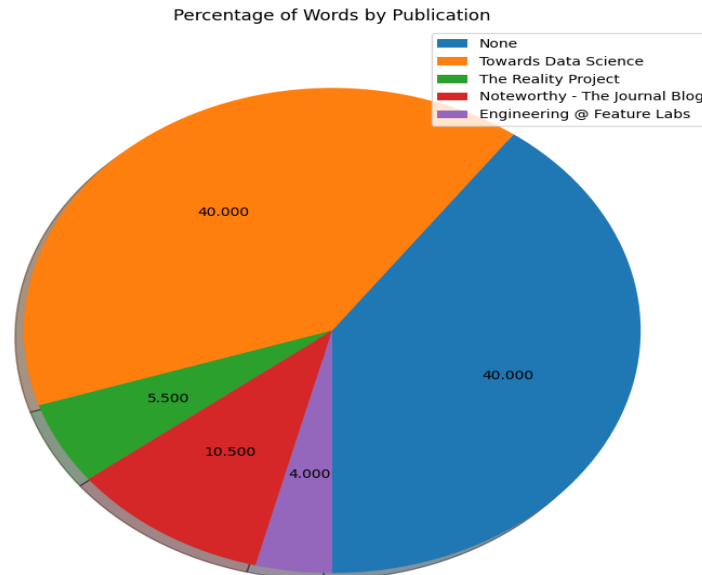
Выполнение работы

Задание 1.1.

```
1  import matplotlib.pyplot as plt
2
3  labels = ['None', 'Towards Data Science', 'The Reality Project',
4           'Noteworthy - The Journal Blog', 'Engineering @ Feature Labs']
5  sizes = [40, 40, 5.5, 10.5, 4]
6  # fig = plt.figure(figsize=(19, 10))
7  plt.title('Percentage of Words by Publication')
8
9  plt.pie(sizes, autopct='%.3f', shadow=True, startangle=-90, radius = 1)
10 plt.legend(loc='upper right', labels = labels)
11 # fig.savefig('pie.png')
12 plt.show()
13
```

1. Зададим переменные и параметры «labels,sizes»
2. Создадим круговую диаграмму(англ. Pie) и зададим параметры
3. Вывод на экран

пример работы программы:



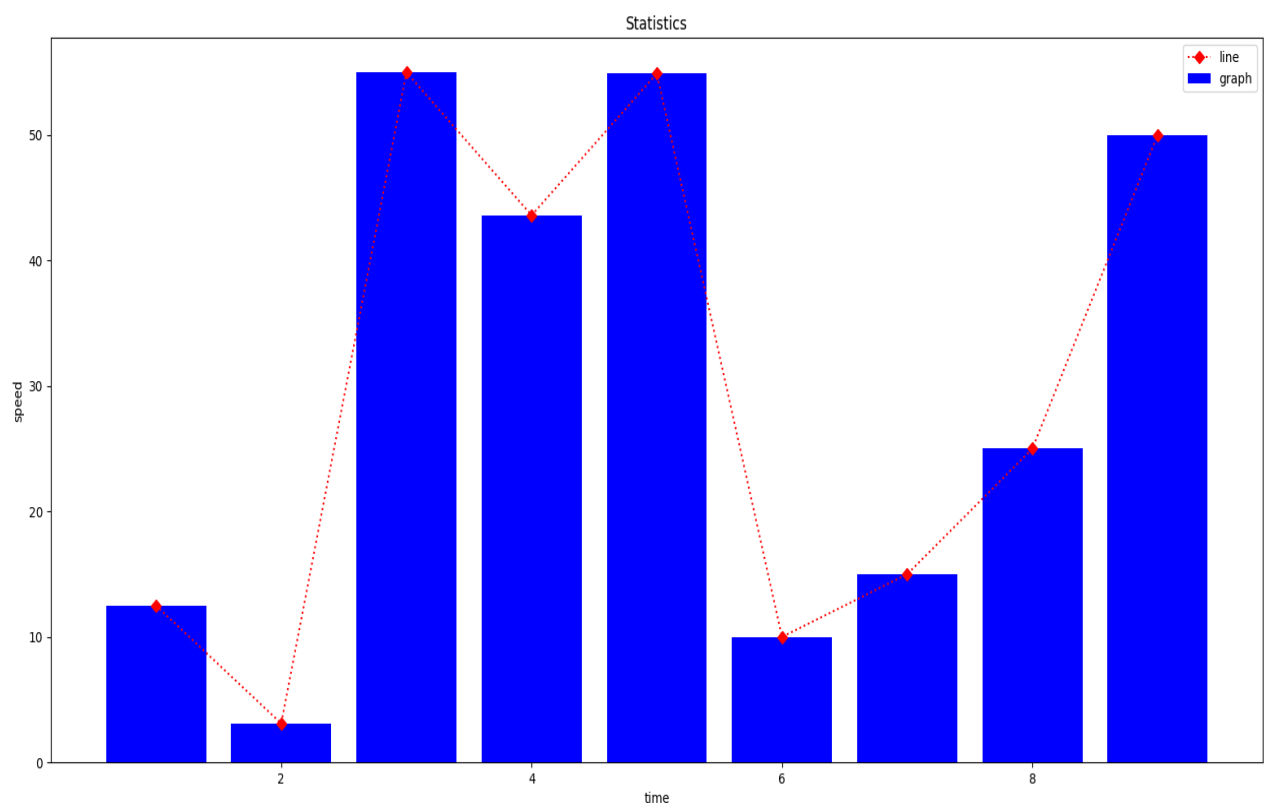
Задание 1.2.

```
1
2 import matplotlib.pyplot as plt
3 x = [1, 2, 3, 4, 5, 6, 7, 8, 9]
4 y = [12.5, 3.1, 55, 43.6, 54.9, 10, 15, 25, 50]
5 plt.plot(x, y, label='f(x)', linestyle='dotted', color='r', marker='D',
6 linewidth='1.5')
7 plt.bar(x, y, label='Statistics', color='blue')
8 plt.xlabel('time')
9 plt.ylabel('speed')
10 plt.title('Statistics')
11
12 plt.legend(loc='upper right', labels=['line', 'graph'])
13 # fig.savefig('pie.png')
14 plt.show()
```

1. Зададим переменные и параметры «x,y»
2. Создадим линию и зададим параметры(англ. Plot).

3. Создадим диаграмму и зададим параметры(англ. bar).

пример работы программы:



Задание 2.

```
1  import tkinter
2  import random
3  WIDTH = 1920
4  HEIGHT = 1080
5  BG_COLOR = 'white'
6  M_RADIUS = 30
7  M_COLOR = 'green'
8  INIT_DX = 3
9  INIT_DY = 3
10 S_BALL = 10
11 MAX_RADIUS = 40
12 MIN_RADIUS = 20
13 BAD_COLOR = 'red'
14 COLORS = ['blue', 'gold', 'green', 'black', BAD_COLOR]
15 NUM_OF_BALLS = 4
16
17 class Balls():
18     def __init__(self, x, y, r, color, dx=0, dy=0):
19         self.x = x
20         self.y = y
21         self.r = r
22         self.color = color
23         self.dx = dx
24         self.dy = dy
25
26     def draw(self):
27         canvas.create_oval(self.x-self.r, self.y-self.r,
28                             self.x+self.r, self.y+self.r,
29                             fill=self.color, outline=BG_COLOR
30                             if self.color != BAD_COLOR else 'black')
31
```

```
32     def hide(self):
33         canvas.create_oval(self.x-self.r, self.y-self.r,
34                             self.x+self.r, self.y+self.r,
35                             fill=BG_COLOR, outline=BG_COLOR)
36
37     def is_collision(self, ball):
38         a = abs(self.x+self.dx-ball.x)
39         b = abs(self.y+self.dy-ball.y)
40         return (a*a + b*b)**0.5 <= self.r + ball.r
41
42     def move(self):
43         if (self.x+self.r+self.dx>=WIDTH) or (self.x-self.r+self.dx<=0):
44             self.dx = -self.dx
45         if (self.y+self.r+self.dy>=HEIGHT) or (self.y-self.r+self.dy<=0):
46             self.dy = -self.dy
47         for ball in balls:
48             if self.is_collision(ball):
49                 if ball.color != BAD_COLOR:
50                     ball.hide()
51                     balls.remove(ball)
52                     self.dx = -self.dx
53                     self.dy = -self.dy
54                 else:
55                     self.dx = self.dy = 0
56         self.hide()
57         self.x += self.dx
58         self.y += self.dy
59         self.draw()
```

```
61 def mouse_click(event):
62     global main_ball
63     if event.num == 1:
64         if 'main_ball' not in globals():
65             main_ball = Balls(event.x, event.y, M_RADIUS, M_COLOR, INIT_DX, INIT_DY)
66             main_ball.draw()
67         else:
68             if main_ball.dy*main_ball.dx>0:
69                 main_ball.dy = -main_ball.dy
70             else:
71                 main_ball.dx = -main_ball.dx
72     elif event.num == 3:
73         if main_ball.dy*main_ball.dx>0:
74             main_ball.dx = -main_ball.dx
75         else:
76             main_ball.dy = -main_ball.dy
77
78 def create_list_of_balls(number):
79     lst = []
80     while len(lst)< number:
81         next_ball = Balls(random.choice(range(MAX_RADIUS, WIDTH-MAX_RADIUS)),
82                             random.choice(range(MIN_RADIUS, HEIGHT-MIN_RADIUS)),
83                             random.choice(range(MIN_RADIUS, MAX_RADIUS)),
84                             random.choice(COLORS))
85         is_collision = False
86         for ball in lst:
87             if next_ball.is_collision(ball):
88                 is_collision = True
89                 break
90         if not is_collision:
91             lst.append(next_ball)
```

```

95     def count_bad_balls(list_of_balls):
96         result = 0
97         for ball in list_of_balls:
98             if ball.color == BAD_COLOR:
99                 result += 1
100         return result
101
102     def main():
103         if 'main_ball' in globals():
104             main_ball.move()
105             if len(balls)-num_of_bad_balls == 0:
106                 canvas.create_text(WIDTH/2, HEIGHT/2, text = 'You won!',
107                                     font = 'Arial 20', fill = 'green')
108                 main_ball.dx = main_ball.dy = 0
109             elif main_ball.dx*main_ball.dy == 0:
110                 canvas.create_text(WIDTH/2, HEIGHT/2, text = 'Loose!',
111                                     font = 'Arial 20', fill = 'red')
112         root.after(S_BALL,main)
113
114
115     root = tkinter.Tk()
116     root.title('Magic balls')
117     canvas = tkinter.Canvas(root,width = WIDTH, height = HEIGHT, bg = BG_COLOR)
118     canvas.pack()
119     canvas.bind('<Button-1>',mouse_click)
120     canvas.bind('<Button-3>',mouse_click, '+')
121     balls = create_list_of_balls(NUM_OF_BALLS)
122     num_of_bad_balls = count_bad_balls(balls)
123     main()
124
125     root.mainloop()

```

пример работы программы:



You won!



Loose!



Заключение.

Благодаря этой лабораторной работе, я узнал: как работать с диаграммами в Python;

Познакомился с модулем `tkinter`; Написал игру с шариками.