ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

**ОТЧЕТ**

**О ВЫПЛОНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**

**«АНИМАЦИЯ СИСТЕМЫ»**

**ПО ДИСЦИПЛИНЕ «ТЕОРЕТИЧЕСКАЯ МЕХАНИКА И ОСНОВЫ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ»**

**ВАРИАНТ ЗАДАНИЯ № 17**

Выполнил(а) студент группы М8О-207Б-22

Диёров Давронхон Умид угли\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись, дата

Проверил и принял

Зав. каф. 802, Бардин Б.С.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

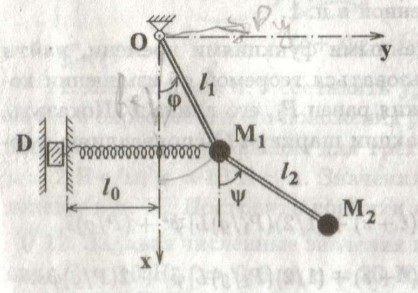
подпись, дата

с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва, 2023

**Задание:** построить анимацию движения системы с помощью Python.

**Механическая система:**



**Текст программы:**

import numpy as np

import matplotlib.pyplot as plt

from matplotlib.animation import FuncAnimation

import math

fig = plt.figure(figsize=[15, 7])

ax = fig.add\_subplot(1, 1, 1)

ax.set(xlim=[-1, 18], ylim=[-4, 10])

Steps = 500

t\_fin = 20

t = np.linspace(0, t\_fin, Steps)

phi = 0.6\*np.sin(1\*t)

psi = 0.4\*np.sin(1.5\*t)

l1 = 3

l2 = 4

X\_A = 2.5

Y\_A = 6

X\_B = X\_A+l1\*np.sin(phi)

Y\_B = Y\_A-l1\*np.cos(phi)

X\_C = X\_B+l2\*np.sin(psi)

Y\_C = Y\_B-l2\*np.cos(psi)

BoxX = 1

BoxY = 2

X\_Box = np.array([-2\*BoxX, 0, 0, -2\*BoxX, -2\*BoxX])

Y\_Box = np.array([Y\_B[0]/3, Y\_B[0]/3, -Y\_B[0]/3, -Y\_B[0]/3, Y\_B[0]/3])

X\_Triangle = np.array([2, 3, 2.5, 2])

Y\_Triangle = np.array([6.5, 6.5, 6.05, 6.5])

X\_Ground = [-2, -2, 12]

Y\_Ground = [8, -3, -3]

X\_Ground1 = np.array([0, 0])

Y\_Ground1 = np.array([0, 6])

K = 17

Sh = 0.4

b = 1/(K-2)

X\_Spr = np.zeros(K)

Y\_Spr = np.zeros(K)

X\_Spr[0] = 0

Y\_Spr[0] = 0

X\_Spr[K-1] = 1

Y\_Spr[K-1] = 0

for i in range(K-2):

X\_Spr[i+1] = b\*((i+1) - 1/2)

Y\_Spr[i+1] = Sh\*(-1)\*\*i

L\_SprX = X\_B

L\_SprY = Y\_B

Drawed\_Spring = ax.plot(X\_Spr\*L\_SprX[0], (Y\_Spr+Y\_A)+L\_SprY[0]-l1)[0]

Drawed\_Box = ax.plot(X\_Box, Y\_Box, linewidth=3)[0]

Drawed\_Triangle = ax.plot(X\_Triangle, Y\_Triangle)[0]

Drawed\_AB = ax.plot([X\_A, X\_B[0]], [Y\_A, Y\_B[0]], linewidth=2)[0]

Point\_A = ax.plot(X\_A, Y\_A, marker='o', markersize=5)[0]

Drawed\_BC = ax.plot([X\_B[0], X\_C[0]], [Y\_B[0], Y\_C[0]], linewidth=2)[0]

Point\_B = ax.plot(X\_B[0], Y\_B[0], marker='o', markersize=15)[0]

Point\_C = ax.plot(X\_C[0], Y\_C[0], marker='o', markersize=15)[0]

ax.plot(X\_Ground, Y\_Ground, color='Black', linewidth=2)

ax.plot(X\_Ground1, Y\_Ground1, color='Black', linewidth=2)

def anima(i):

Drawed\_AB.set\_data([X\_A, X\_B[i]], [Y\_A, Y\_B[i]])

Drawed\_BC.set\_data([X\_B[i], X\_C[i]], [Y\_B[i], Y\_C[i]])

Drawed\_Spring.set\_data(X\_Spr\*L\_SprX[i], (Y\_Spr+L\_SprY[i]+Y\_A)-l1-l1)

Drawed\_Box.set\_data(X\_Box, Y\_Box+Y\_B[i])

Point\_B.set\_data(X\_B[i], Y\_B[i])

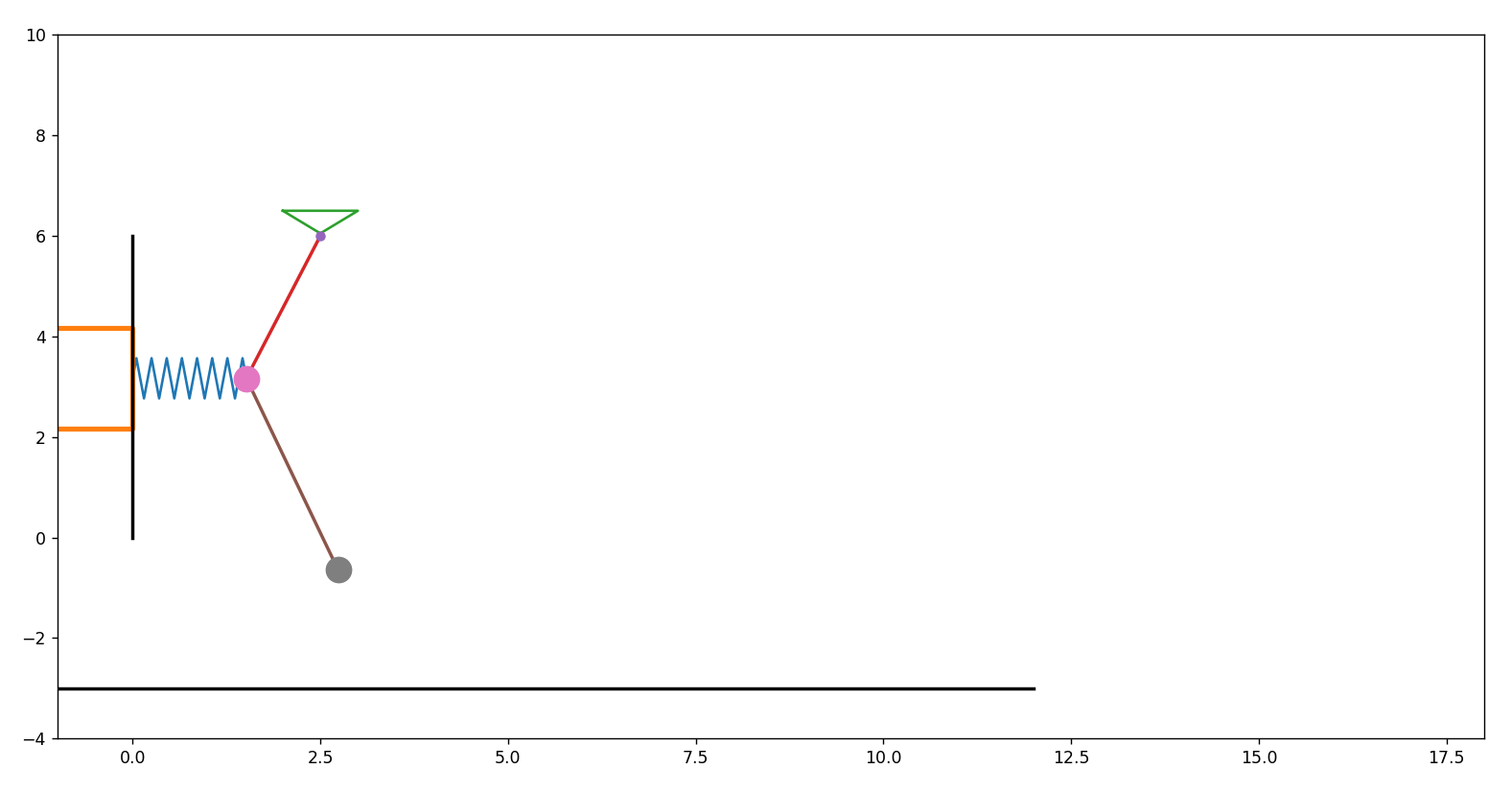
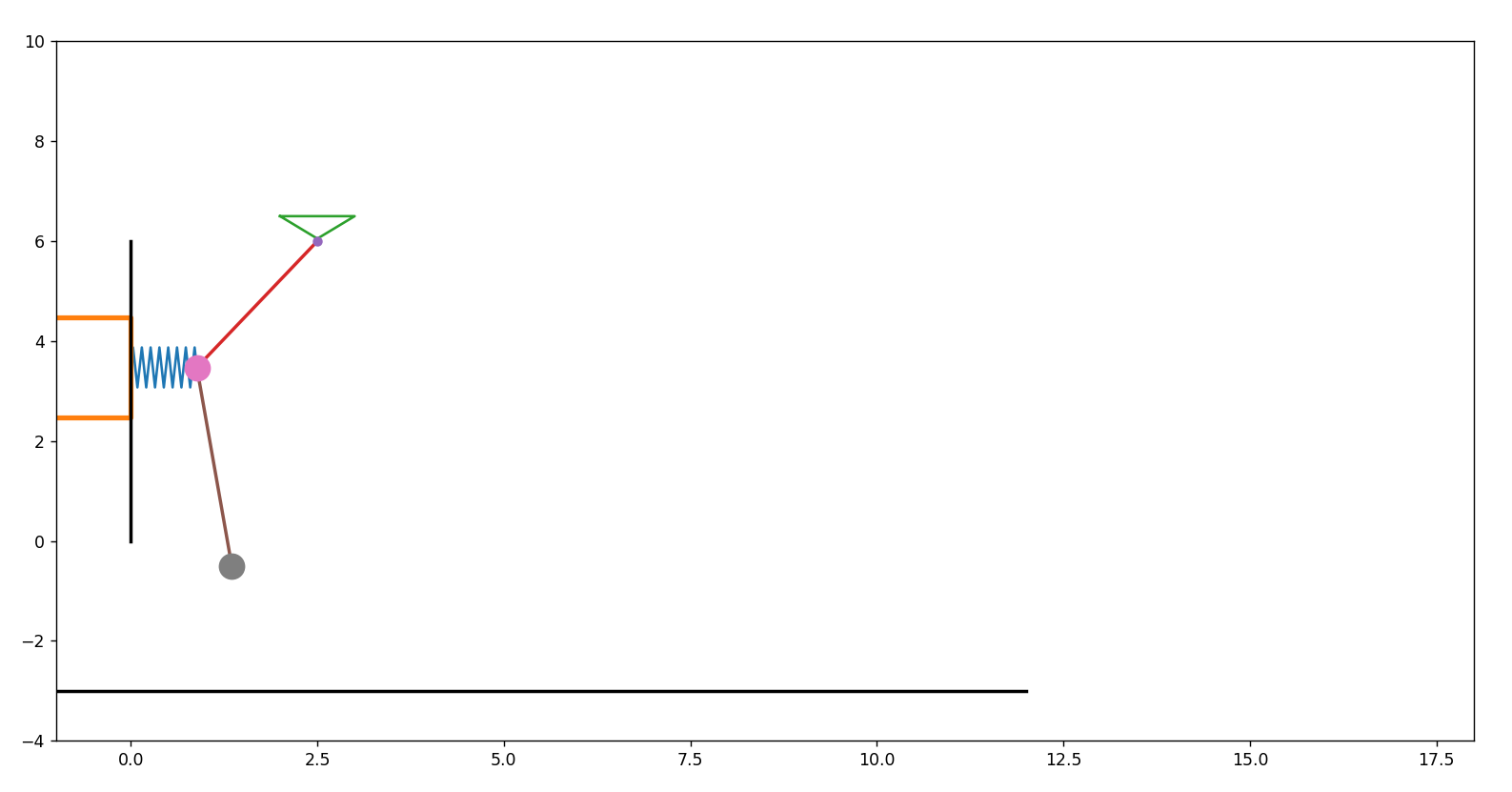
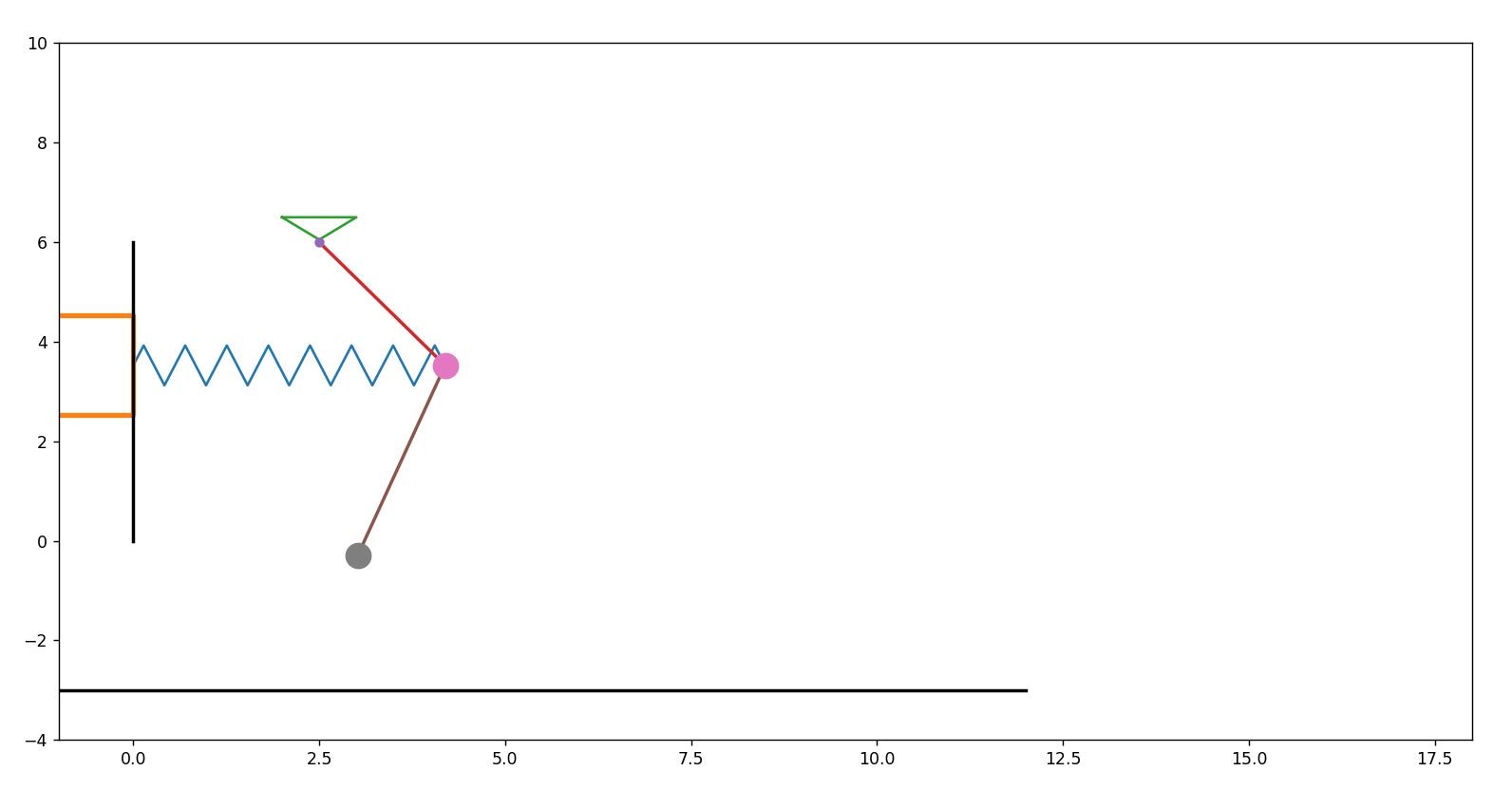
Point\_C.set\_data(X\_C[i], Y\_C[i])

return [Drawed\_AB, Point\_B, Drawed\_Spring, Drawed\_BC]

anim = FuncAnimation(fig, anima, frames=len(t), interval=40, repeat=False)

plt.show()

**Результат работы:**

****

**Вывод:** проделав лабораторную работу, приобрёл практические навыки в анимации системы.