

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ  
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

**ОТЧЕТ**  
**О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**  
**«АНИМАЦИЯ СИСТЕМЫ»**  
**ПО ДИСЦИПЛИНЕ «ТЕОРЕТИЧЕСКАЯ МЕХАНИКА И**  
**ОСНОВЫ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ»**  
**ВАРИАНТ ЗАДАНИЯ №24**

Выполнил(а) студент группы М8О-208Б-20

**Попов Матвей Романович** \_\_\_\_\_  
подпись, дата

Проверил и принял

Доцент каф. 802, Чекина Е.А. \_\_\_\_\_  
подпись, дата

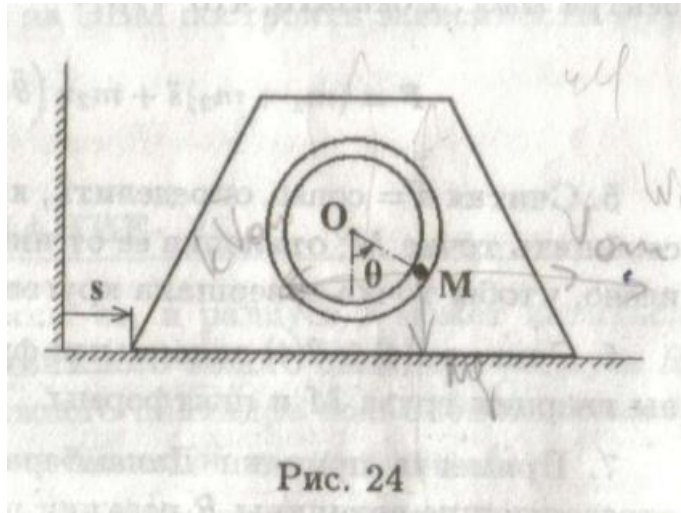
с оценкой \_\_\_\_\_

Москва, 2021

## Задание:

Реализовать анимацию движения механической системы используя язык программирования Python.

## Механическая система:



## Текст программы:

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation
import sympy as sp
import math

def Trapezoid(x0, y0):
    PX = [x0 - 10, x0 - (2/3) * 10, x0 + (2/3) * 10, x0 + 10, x0 - 10]
    PY = [y0 - 7.5, y0 + 10, y0 + 10, y0 - 7.5, y0 - 7.5]
    return PX, PY

t = sp.Symbol('t')
s = 4 * sp.cos(3 * t)
phi = 4 * sp.sin(t - 10)

Xspr = s * sp.cos(math.pi) + 0.8
Yspr = -s * sp.sin(math.pi) + 7.5

VmodSignPrism = sp.diff(s, t)
VxSpr = VmodSignPrism * sp.cos(math.pi)
VySpr = -VmodSignPrism * sp.sin(math.pi)

xA = Xspr - 5 * sp.sin(phi)
yA = Yspr + 5 * sp.cos(phi)

omega = sp.diff(phi, t)

VxA = VxSpr - omega * 5 * sp.cos(phi)
VyA = VySpr - omega * 5 * sp.sin(phi)

T = np.linspace(0, 20, 1000)
XSpr = np.zeros_like(T)
YSpr = np.zeros_like(T)
VXSpr = np.zeros_like(T)
VYSpr = np.zeros_like(T)
```

```

Phi = np.zeros_like(T)
XA = np.zeros_like(T)
YA = np.zeros_like(T)
VXA = np.zeros_like(T)
VYA = np.zeros_like(T)

for i in np.arange(len(T)):
    XSpr[i] = sp.Subs(Xspr, t, T[i])
    YSpr[i] = sp.Subs(Yspr, t, T[i])
    VXSpr[i] = sp.Subs(VxSpr, t, T[i])
    VYSpr[i] = sp.Subs(VySpr, t, T[i])
    Phi[i] = sp.Subs(phi, t, T[i])
    XA[i] = sp.Subs(xA, t, T[i])
    YA[i] = sp.Subs(yA, t, T[i])
    VXA[i] = sp.Subs(VxA, t, T[i])
    VYA[i] = sp.Subs(VyA, t, T[i])

fig = plt.figure(figsize = (17, 10))

ax1 = fig.add_subplot(121)
ax1.axis('equal')
ax1.set(xlim=[XSpr.min() - 20, XSpr.max() + 20], ylim=[YSpr.min() - 20, YSpr.max() + 20])
ax1.plot([XSpr.min() - 10, XSpr.max() + 10], [-(XSpr.min() - 10) * sp.tan(math.pi), -
(XSpr.max() + 10) * sp.tan(math.pi)], 'black')

PrX, PrY = Trapezoid(XSpr[0], YSpr[0])
Prism = ax1.plot(PrX, PrY, 'red')[0]

radius, = ax1.plot([XSpr[0], XA[0]], [YSpr[0], YA[0]], 'black')

varphi = np.linspace(0, 6.28, 20)
r = 0.2
Point = ax1.plot(XA[0] + r * np.cos(varphi), YA[0] + r * np.sin(varphi))[0]

T = np.linspace(0, 20, 1000)
VxP = np.zeros_like(T)
VyP = np.zeros_like(T)
VxC = np.zeros_like(T)
VyC = np.zeros_like(T)
l = np.zeros_like(T)

for i in np.arange(len(T)):
    VxP[i] = 0.14 * np.cos(0.01 * i)
    VyP[i] = -0.01 * np.sin(0.01 * i)
    VxC[i] = 0.04 * np.cos(0.01 * i)
    VyC[i] = 0
    l[i] = i

ax2 = fig.add_subplot(422)
ax2.plot(l, VxP)
ax2.set_ylabel('vx of point')

ax3 = fig.add_subplot(424)
ax3.plot(l, VyP)
ax3.set_ylabel('vy of point')

ax4 = fig.add_subplot(426)
ax4.plot(l, VxC)
ax4.set_ylabel('vx of center')

ax5 = fig.add_subplot(428)
ax5.plot(l, VyC)
ax5.set_ylabel('vy of center')

```

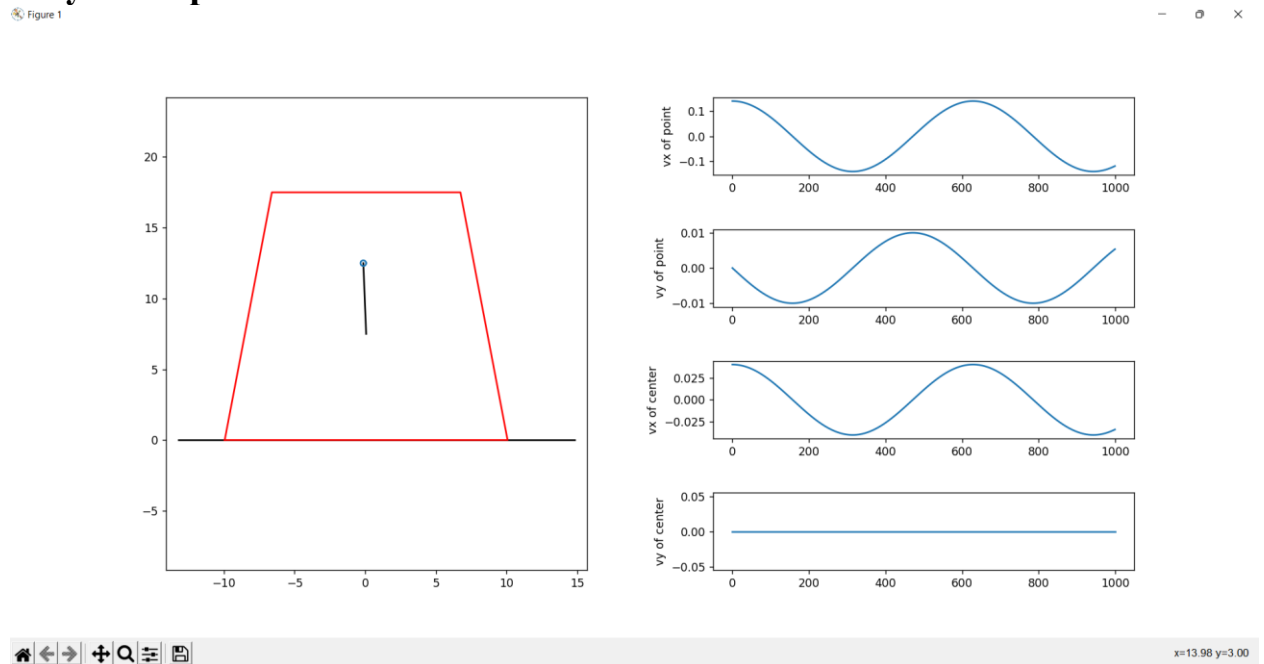
```
plt.subplots_adjust(wspace = 0.3, hspace = 0.7)

def anima(i):
    PrX, PrY = Trapezoid(XSpr[i], YSpr[i])
    Prism.set_data(PrX, PrY)
    radius.set_data([XSpr[i], XA[i]], [YSpr[i], YA[i]])
    Point.set_data(XA[i] + r * np.cos(varphi), YA[i] + r * np.sin(varphi))
    return Prism, radius, Point

anim = FuncAnimation(fig, anima, frames = 1000, interval = 0.01, blit = True)

plt.show()
```

## Результат работы:



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