

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 № 13**

Выполнил(а) студент группы М8О-208Б-22

Былькова Кристина Алексеевна \_\_\_\_\_  
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.

Текст программы:

```
import matplotlib.pyplot as p
from matplotlib.animation import FuncAnimation
import numpy as n

# Определяем сетку времени
T = n.linspace(0, 10, 100)

Phi = n.sin(0.5*T)*n.pi
Psi = n.sin(0.55*T)*n.pi

# Создаем окно для анимации:
fgr = p.figure()
plt = fgr.add_subplot(1,1,1)
plt.axis('equal')

# Задаем начальные геометрические характеристики
r1 = 3
m1 = 0
r3 = 1
r2 = 1
m2 = 0

plt.plot([- (2*r1), (2*r1)], [0,0], '--')
plt.plot([0,0], [- (2*r1), (2*r1)], '--')

# Малая неподвижная окружность
Alp = n.linspace(0, 2*n.pi, 100)
Xsh = r3 * n.sin(Alp)
Ysh = r3 * n.cos(Alp)
Shtift = plt.plot(Xsh,Ysh)[0]

Xc1 = (r1-r3) * n.sin(Phi[0])
Yc1 = (r1-r3) * -n.cos(Phi[0])
# Большая подвижная окружность
Xa1 = r1 * n.sin(Alp)
Ya1 = r1 * n.cos(Alp)
A1 = plt.plot(Xa1+Xc1,Ya1+Yc1)[0]

Xc2 = Xc1+(r1-r2) * n.sin(Psi[0])
Yc2 = Yc1+(r1-r2) * -n.cos(Psi[0])
# Малая подвижная окружность
Xa2 = r2 * n.sin(Alp)
Ya2 = r2 * n.cos(Alp)
A2 = plt.plot(Xa2+Xc2,Ya2+Yc2)[0]

# Функция зменения кадров
def run(i):
    Xc1 = (r1-r3) * n.sin(Phi[i])
    Yc1 = (r1-r3) * -n.cos(Phi[i])
```

```
Xc2 = Xc1+(r1-r2) * n.sin(Psi[i])
Yc2 = Yc1+(r1-r2) * -n.cos(Psi[i])
A1.set_data(Xa1+Xc1,Ya1+Yc1)
A2.set_data(Xa2+Xc2,Ya2+Yc2)

# Запуск анимации
anim = FuncAnimation(fgr, run, frames = len(T), interval = 1)
fgr.show()
```

Результат работы программы:

