|  |  |
| --- | --- |
| **лого для документов 2022** | ***Федеральное агентство по рыболовству***  ***Федеральное государственное бюджетное образовательное***  ***учреждение высшего образования***  ***«Астраханский государственный технический университет»***  **Система менеджмента качества в области образования, воспитания, науки и инноваций**  **ООО «ДКС РУС» по международному стандарту ISO 9001:2015** |

ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И КОММУНИКАЦИЙ

**КАФЕДРА АВТОМАТИЗИРОВАННЫХ СИСТЕМ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ**

**ЗАЩИТА ИНФОРМАЦИИ**

(*наименование дисциплины*)

**ОТЧЕТ**

о выполнении заданий к лабораторной работе №2

*(название лабораторной работы)*

|  |  |
| --- | --- |
|  | Выполнила:  студент гр. ДИНРБ-41 |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Кузургалиев Р.А. |
|  | «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_2024 г. |
|  | Максимальное количеству баллов \_\_\_\_\_  ЗАЩИЩЕНО:  Получено баллов \_\_\_\_\_\_ |
|  | Преподаватель: \_\_\_\_\_\_\_Сахипова Ж.Ш. |
|  | «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2024 г. |

**Астрахань – 2024**

**Лабораторная работа №2:**

**ЗАДАНИЕ К ЛАБОРАТОРНОЙ РАБОТЕ №2**

Код программы на языке Python:

import matplotlib.pyplot as plt  
import numpy as np  
  
def X(t, A, B, C, a, b, c):  
 return A \* np.sin(a\*t) + B \* np.cos(b \* t) + t\*C\* np.cos(np.cos(c \* t))  
print('Введите A:')  
A = float(input())  
print('Введите B:')  
B = float(input())  
print('Введите C:')  
C = float(input())  
print('Введите a:')  
a = float(input())  
while a <= 0:  
 print('Введите a:')  
 a = float(input())  
  
print('Введите b:')  
b = float(input())  
while b <= 0:  
 print('Введите b:')  
 b = float(input())  
print('Введите c:')  
c = float(input())  
while c <= 0:  
 print('Введите c:')  
 c = float(input())  
print('Введите криптографический ключ k:')  
K = int(input())  
print('Введите количество интервалов N:')  
N = int(input())  
print('Введите начало исследуемого интервала t1:')  
t1 = int(input())  
print('Введите начало исследуемого интервала t2:')  
t2 = int(input())  
# Генерация значений t  
t\_values = np.linspace(t1, t2, 1000)  
# Вычисление значений функции  
y\_at\_x = X(t\_values, A, B, C, a, b, c)  
# Построение графика  
plt.title('График функции X(t)')  
plt.xlabel('t')  
plt.ylabel('X(t)')  
x\_coords = []  
y\_coords = []  
ti = t1  
for i in range(N+1):  
 x\_coords.append(ti)  
 y\_coords.append(X(ti, A, B, C, a, b, c))  
 plt.plot([ti, ti], [0, X(ti, A, B, C, a, b, c)], color='red', linestyle='-') # Вторичные данные  
 ti = ti + abs(t2-t1)/N  
# Построение линии от point1 до point2  
plt.plot(t\_values, y\_at\_x, color='blue', linestyle='-') # Основные данные  
plt.plot(x\_coords, y\_coords, color='orange', linestyle='-') # Вторичные данные  
plt.axhline(0, color='black',linewidth=0.5, ls='--')  
plt.axvline(0, color='black',linewidth=0.5, ls='--')  
plt.grid()  
plt.show()  
x\_pairs = []  
y\_pairs = []  
for i in range(N):  
 x\_pairs.append([x\_coords[i], x\_coords[i+1]])  
 y\_pairs.append([y\_coords[i], y\_coords[i+1]])  
rng = np.random.RandomState(K)  
indices = rng.permutation(len(x\_pairs))  
x\_pairs = np.array(x\_pairs)  
y\_pairs = np.array(y\_pairs)  
shuffle\_x = x\_pairs  
shuffle\_y = y\_pairs[indices]  
#print('x: ',shuffle\_x, ' y: ' , shuffle\_y)  
plt.title('График функции Y(t)')  
plt.xlabel('t')  
plt.ylabel('Y(t)')  
for i in range(N):  
 plt.plot(shuffle\_x[i], shuffle\_y[i], color='green', linestyle='-')  
for i in range(N-1):  
 if shuffle\_y[i+1, 0] > shuffle\_y[i, 1]:  
 pair = [0, shuffle\_y[i+1, 0]]  
 else:  
 pair = [0, shuffle\_y[i, 1]]  
 plt.plot([shuffle\_x[i+1, 0], shuffle\_x[i+1, 0]], pair, color='red', linestyle='-')  
plt.plot([shuffle\_x[0, 0], shuffle\_x[0, 0]], [0, shuffle\_y[0,0]], color='red', linestyle='-')  
plt.plot([shuffle\_x[N-1, 1], shuffle\_x[N-1, 1]], [0, shuffle\_y[N-1,1]], color='red', linestyle='-')  
plt.grid()  
plt.show()  
  
original\_indices = np.argsort(indices) # Получаем индексы, которые сортируют перемешанный массив  
restored\_array\_x = shuffle\_x  
restored\_array\_y = shuffle\_y[original\_indices]  
  
for i in range(N):  
 plt.plot(restored\_array\_x[i], restored\_array\_y[i], color='blue', linestyle='-')  
for i in range(N-1):  
 if restored\_array\_y[i+1, 0] > restored\_array\_y[i, 1]:  
 pair = [0, restored\_array\_y[i+1, 0]]  
 else:  
 pair = [0, restored\_array\_y[i, 1]]  
 plt.plot([restored\_array\_x[i+1, 0], restored\_array\_x[i+1, 0]], pair, color='red', linestyle='-')  
plt.plot([restored\_array\_x[0, 0], restored\_array\_x[0, 0]], [0, restored\_array\_y[0,0]], color='red', linestyle='-')  
plt.plot([restored\_array\_x[N-1, 1], restored\_array\_x[N-1, 1]], [0, restored\_array\_y[N-1,1]], color='red', linestyle='-')  
#print('x: ',restored\_array\_x, ' y: ' , restored\_array\_y)  
plt.grid()  
plt.show()