Лабораторная работа №3

Определение требований к программному обеспечению. Разработка Технического задания

Цель работы: приобретение практических навыков по построению ТЗ

Задание к лабораторной работе:

Разработать проектный документ Техническое задание (ТЗ) на разработку программного продукта согласно теме индивидуального задания (Приложение 1).

Ход работы:

Задание

2. Анимация алгебpаических опеpаций.

Написать пpогpамму, наглядно показывающую пpоцесс:

- умножения, сложения , вычитания двух матpиц;

- составления обратной и транспонированной матриц.

Все тpи матpицы показать на экpане и с вpеменной задеpжкой показать пpоцесс заполнения элементов pезультиpующей матpицы.

import numpy as np

import matplotlib.pyplot as plt

from matplotlib.animation import FuncAnimation

def matrix\_addition\_animation(matrix\_a, matrix\_b):

    result\_matrix = np.zeros\_like(matrix\_a)

    fig, ax = plt.subplots()

    def update(frame):

        ax.clear()

        ax.axis('off')

        ax.set\_title("Matrix Addition")

        ax.text(0.1, 0.8, f"Frame: {frame}", fontsize=12)

        ax.text(0.1, 0.6, "Matrix A:", fontsize=12)

        ax.text(0.1, 0.5, str(matrix\_a), fontsize=12)

        ax.text(0.1, 0.3, "Matrix B:", fontsize=12)

        ax.text(0.1, 0.2, str(matrix\_b), fontsize=12)

        alpha = frame / frames  # Процент выполнения

        result\_matrix = matrix\_a \* (1 - alpha) + matrix\_b \* alpha

        ax.text(0.1, 0.0, f"Result Matrix (Interpolated):", fontsize=12)

        ax.text(0.1, -0.1, str(result\_matrix), fontsize=12)

    frames = 100  # Количество кадров

    ani = FuncAnimation(fig, update, frames=frames, interval=50, repeat=False)

    plt.show()

# Пример использования

matrix\_a = np.array([[1, 2], [3, 4]])

matrix\_b = np.array([[5, 6], [7, 8]])

matrix\_addition\_animation(matrix\_a, matrix\_b)

