МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по учебной практике

по направлению «Генетические алгоритмы»

Студент гр. 2383 _____ Борисов И.П.

Преподаватель _____ Жангиров Т.Р.

Санкт-Петербург

2024

Цель работы

Доработать программу до полностью рабочего состояния.

Задание

Сделать GUI, где решается генетическим алгоритмом задача: Дано N матриц A с произвольными размерами (A_1: p_0 x p_1, A_2: p_1 x p_2, … , A_N: p_{N-1} x p_N). Необходимо определить порядок перемножения матриц, чтобы минимизировать количество операций умножения для вычисления результирующей матрицы B: p_0 x p_N. И в зависимости от выбора пользователя данные вводятся вручную, считываются с файла или случайно генерируются.

Выполнение работы.

Был изменён вывод хромосом – теперь выводится строчка с матрицами, которые умножаются по заданному в хромосоме порядке (Рисунок 1).

Была добавлена случайная генерация данных. Генерация матриц. Генерируется число матриц от 4 до 10, затем создаются случайно размеры матриц, величиной от 1 до 30, а первое число следующей матрицы всегда равно второму числу текущей матрицы. Созданные матрицы записываются в классе All_Matrix.

Генерация параметров. Генерируется случайное число: размер популяции от 6 до 30, количество поколений от 10 до 100, вероятность скрещивания от 0.6 до 0.9, вероятность мутации от 0.1 до 0.4. Параметры сохраняются в классе Parametres.

Была добавлена обработка исключений. Когда пользователь вводит некорректные данные вручную или через файл, открывается окно с сообщением об ошибке.

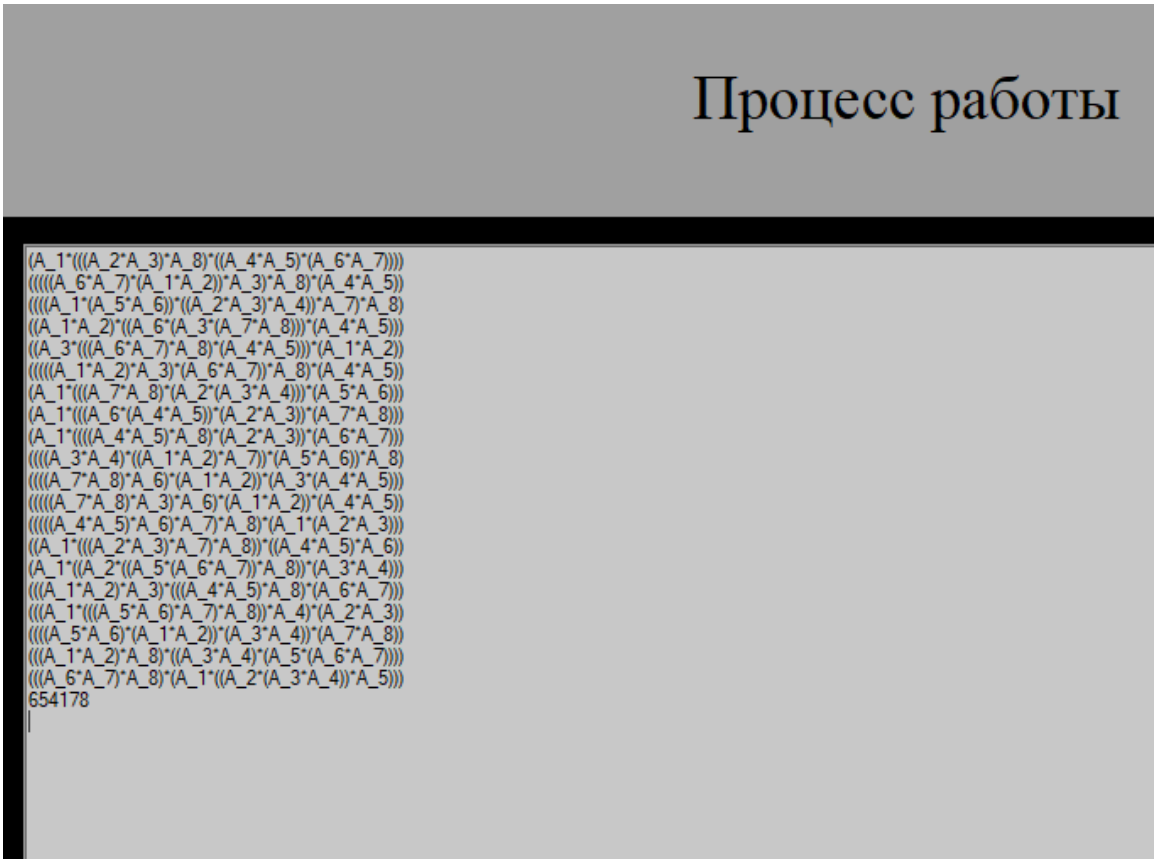
Процесс работы
(A_1*(((A_2*A_3)*A_8)*((A_4*A_5)*(A_6*A_7))))
(((((A_6*A_7)*(A_1*A_2))*A_3)*A_8)*(A_4*A_5))
((((A_1*(A_5*A_6))*((A_2*A_3)*A_4))*A_7)*A_8)
((A_1*A_2)*((A_6*(A_3*(A_7*A_8)))*(A_4*A_5)))
((A_3*(((A_6*A_7)*A_8)*(A_4*A_5)))*(A_1*A_2))
(((((A_1*A_2)*A_3)*(A_6*A_7))*A_8)*(A_4*A_5))
(A_1*(((A_7*A_8)*(A_2*(A_3*A_4)))*(A_5*A_6)))
(A_1*(((A_6*(A_4*A_5))*(A_2*A_3))*(A_7*A_8)))
(A_1*((((A_4*A_5)*A_8)*(A_2*A_3))*(A_6*A_7)))
((((A_3*A_4)*((A_1*A_2)*A_7))*(A_5*A_6))*A_8)
((((A_7*A_8)*A_6)*(A_1*A_2))*(A_3*(A_4*A_5)))
(((((A_7*A_8)*A_3)*A_6)*(A_1*A_2))*(A_4*A_5))
(((((A_4*A_5)*A_6)*A_7)*A_8)*(A_1*(A_2*A_3)))
((A_1*(((A_2*A_3)*A_7)*A_8))*((A_4*A_5)*A_6))
(A_1*((A_2*((A_5*(A_6*A_7))*A_8))*(A_3*A_4)))
(((A_1*A_2)*A_3)*(((A_4*A_5)*A_8)*(A_6*A_7)))
(((A_1*(((A_5*A_6)*A_7)*A_8))*A_4)*(A_2*A_3))
((((A_5*A_6)*(A_1*A_2))*(A_3*A_4))*(A_7*A_8))
(((A_1*A_2)*A_8)*((A_3*A_4)*(A_5*(A_6*A_7))))
(((A_6*A_7)*A_8)*(A_1*((A_2*(A_3*A_4))*A_5)))
654178

Рисунок 1.