інструментів для створення складних додатків. Angular також має жорсткіші вимоги до структури проекту, що допомагає зберігати проект організованим та підтримувати кодову базу в чистоті.

React є одним з найпопулярніших фреймворків, який забезпечує гнучкість та ефективність при розробці веб-додатків. Він має багату екосистему та широку базу користувачів, що дає можливість розробникам швидко знайти рішення на будь-які проблеми. React також надає можливість використовувати компоненти з будь-яким іншим фреймворком або бібліотекою.

Отже, вибір між цими фреймворками залежить від різних факторів, і немає однозначної відповіді на питання про те, який краще.

Список літератури:

1. Штучний інтелект. Wikipedia: веб-сайт URL: http://surl.li/gwryz . (дата звернення 01.05.2023).

2. Штучний інтелект: етапи, загрози та стратегії / Офіційний сайт газети «Open». URL http://surl.li/gwrzf (дата звернення 02.05.2023).

3. Мартін Форд – професор Массачусетського технологічного університету: Пришестя роботів. Техніка і загроза майбутнього безробіття. Київ : Видавництво «Наш формат», 2016. 124 с.

4. Що може зробити зі світом штучний інтелект? Радіо-свобода: веб-сайт URL: https://www.radiosvoboda.org/a/details/28891073.html (дата звернення 02.05.2023).

УДК 004.2

Костенко Р.О., студент 2 курсу спеціальності 122 «Комп'ютерні науки»
Науковий керівник:
Потапова Н. А., к.е.н., доцент, доцент кафедри інформаційних технологій

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ РОЗВ'ЯЗУВАННЯ СИСТЕМ ЛІНІЙНИХ РІВНЯНЬ МЕТОДОМ ГАУСА

Донецький національний університет імені Василя Стуса, м. Вінниця

У сучасній математиці та науці процес розв'язання систем лінійних рівнянь використовується надзвичайно часто. Один з найпоширеніших методів розв'язання систем лінійних рівнянь - це метод Гауса, який був розроблений відомим математиком Карлом Фрідріхом Гаусом в 19 столітті. Метод Гауса полягає у послідовному виконанні елементарних операцій над рядками матриці коефіцієнтів системи рівнянь з метою приведення її до трикутної форми. Після приведення матриці до трикутної форми можна з легкістю визначити значення невідомих змінних шляхом використання методу зворотнього ходу. [1]

Метод Гауса є дуже ефективним та широко використовується у різних галузях науки та техніки, де потрібно вирішувати системи лінійних рівнянь. Він є одним з основних інструментів чисельного аналізу та алгебри. Наступний код є реалізацією методу Гауса для розв’язування систем лінійних рівнянь написаний об’єктно-орієнтованою мовою програмування Java. Метод має такі етапи:

1. Перевірка коректності розміру матриці. Матриця повинна мати розмірність n * (n+1), де n - кількість невідомих.

2. Приведення матриці до трикутної форми з використанням елементарних перетворень рядків. У циклі по k здійснюється вибір головного елемента maxRow - максимального за модулем елемента у стовпці k, а потім виконується перестановка рядків k та maxRow. Далі за допомогою коефіцієнта factor елементи, що знаходяться під діагоналлю, приводяться до нуля.

3. Обернений хід методу Гауса. У цьому етапі розв'язується система з нижньої до верхньої строчки з використанням знайдених попередньо значень x.

4. Перетворення одновимірного масиву x у двовимірний масив result, що повертається як результат роботи методу.

У реалізації використовується клас BigDecimal для округлення значень до двох знаків після коми. Це робить код більш точним і надійним для обробки чисел з плаваючою точкою. [2]

Лістинг програмного коду класу GaussMethod:

```
public class GaussMethod {
    public static double[][] solve(double[][] matrix) {
        int n = matrix.length;
        int m = matrix[0].length;
        if (n != m - 1) {
            throw new IllegalArgumentException("Матриця повинна бути розмірності
n x (n+1)");
        }
        for (int k = 0; k < n; k++) {
            int maxRow = k;
            for (int i = k + 1; i < n; i++) {
                if (Math.abs(matrix[i][k]) > Math.abs(matrix[maxRow][k])) {
                    maxRow = i;
                }
            }
            double[] temp = matrix[k];
            matrix[k] = matrix[maxRow];
            matrix[maxRow] = temp;
            for (int i = k + 1; i < n; i++) {
                double factor = matrix[i][k] / matrix[k][k];
                for (int j = k + 1; j < m; j++) {
                    matrix[i][j] -= factor * matrix[k][j];
                }
                matrix[i][k] = 0.0;
            }
        }
        double[] x = new double[n];
        for (int i = n - 1; i >= 0; i--) {
            double sum = 0.0;
            for (int j = i + 1; j < n; j++) {
                sum += matrix[i][j] * x[j];
            }
            x[i] = (matrix[i][m-1] - sum) / matrix[i][i];
        }
```

```
        double[][] result = new double[n][1];
        for (int i = 0; i < n; i++) {
            BigDecimal bd = new BigDecimal(x[i]).setScale(2,
RoundingMode.HALF_UP);
            result[i][0] = bd.doubleValue();
        }
        return result;
    }
}
```

Для демонстрації роботи програми зрівняємо результати роботи з ручним пошуком розв’язання в програмі Excel, отже:

	x1	x2	x3	b
	2	-3	4	15
	3	1	-1	4
	4	-1	2	5
1 ітерація	x1	x2	x3	b
	1	-1,5	2	7,5
	0	5,5	-7	-18,5
	0	5	-6	-25
2 ітерація	x1	x2	x3	b
	1	-1,5	2	7,5
	0	1	-1,27273	-3,36364
	0	0	0,363636	-8,18182
3 ітерація	x1	x2	x3	b
	1	-1,5	2	7,5
	0	1	-1,27273	-3,36364
	0	0	1	-22,5
		x1	4,5	
		x2	-32	
		x3	-22,5	

Рис. 1.1. Результат роботи Excel

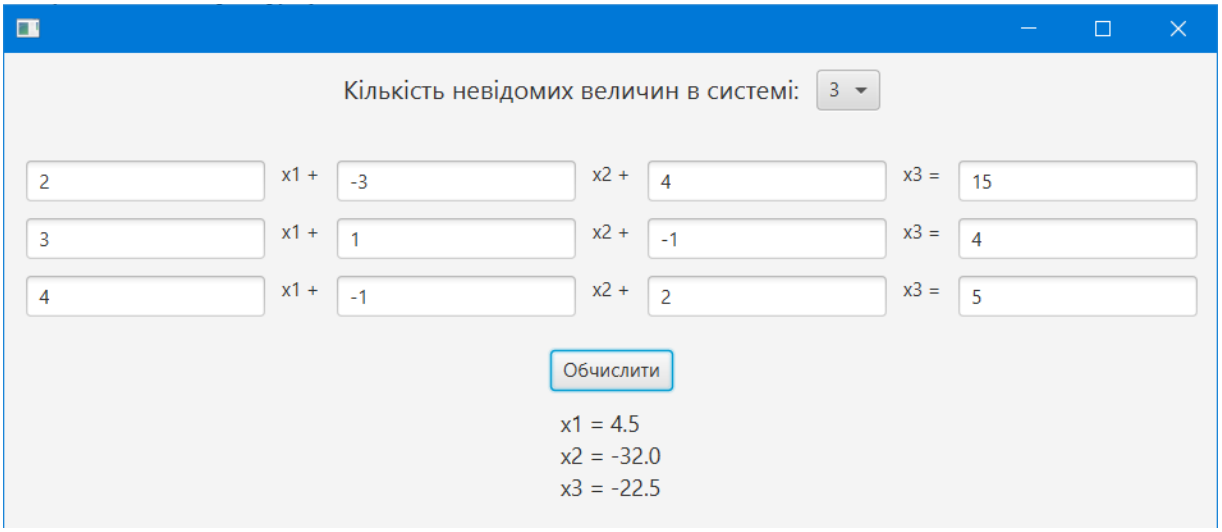

Рис. 1.2. Результат роботи програми

Отже, метод Гауса дозволяє швидко та ефективно розв'язувати системи лінійних рівнянь за допомогою елементарних перетворень рядків, що не змінюють розв'язку системи. Розв'язуючи систему за допомогою цього методу, ми зводимо її до трикутної форми, яка спрощує розв'язування. Можем побачити, що

результат роботи програми та Excel аналогічно, отже програма працює коректно та може облегшити розв’язання такого роду задач. [3]

Список літератури

1. *Волонтир Л.О., Зелінська О.В., Потапова Н.А., Чіков І.А. Чисельні методи. Навчальний посібник. Вінниця: ВНАУ. 2020. 322 с.*
2. *Jaan K., “Numerical methods in engineering with Matlab”, 2005. 435 с.*
3. *JavaFX Official Documentation. URL: https://fxdocs.github.io/docs/html5/*

УДК 004.6

Лаптєва М. А.., студентка 1 курсу
спеціальності 122 «Комп’ютерні науки»
Науковий керівник:
Гончар В. М., асистент
кафедри інформаційних технологій

АЛГОРИТМИ ЗНАХОДЖЕННЯ КІЛЬКОСТІ ШЛЯХІВ МІЖ ВЕРШИНАМИ В ГРАФАХ

Донецький національний університет імені Василя Стуса, м. Вінниця

Існує декілька алгоритмів для знаходження кількості шляхів між вершинами графа. Вибір алгоритму залежить від конкретних характеристик графа і бажаної ефективності обчислень. Нижче наведено три найпоширеніші алгоритми, які використовуються для цієї мети:

1. **Пошук в глибину (DFS):**

Пошук у глибину (DFS) - це алгоритм, який використовується для дослідження або обходу графа або деревовидної структури даних. Він починається з заданої вихідної вершини і систематично досліджує якомога далі вздовж кожної гілки, перш ніж повернутися назад. DFS можна використовувати для різних завдань, таких як пошук зв'язаних компонентів, виявлення циклів, топологічне сортування і, як згадувалося раніше, знаходження кількості шляхів між вершинами.

DFS використовує рекурсивний підхід для обходу графа. Алгоритм підтримує стек для відстеження відвіданих вершин та їхніх сусідів, які ще не були досліджені. При зворотному обході алгоритм витягує останню вершину зі стеку і продовжує досліджувати її сусідів, що залишилися невідвіданими.

DFS часто реалізується за допомогою рекурсивної функції, що робить код лаконічним та інтуїтивно зрозумілим.

Під час виконання DFS алгоритм може виконувати різні дії на різних етапах, такі як ініціалізація обходу, обробка вершини або завершення обходу. Модифікуючи реалізацію, ви можете адаптувати DFS до конкретних вимог.