**Основные этапы работы программы:**

1. **Чтение данных из файла:**
   * Программа считывает квадратную матрицу размера N x N из входного файла input.txt с использованием Scanner.
   * Данные записываются в двумерный массив A.
2. **Инициализация начальных значений:**
   * Вектор x инициализируется единицами (Arrays.fill(x, 1.0)).
   * Устанавливаются значения для eps (точность), maxIter (максимальное количество итераций), а также начальные значения для lambdaOld, lambdaCurrent и lambdaOlder.
3. **Начальная нормировка вектора:**
   * Вычисляется вектор Ax = A \* x с помощью метода multiply.
   * Нормировка вектора осуществляется через деление на евклидову норму (метод norm).
4. **Основной цикл итераций степенного метода:**
   * В каждой итерации вычисляется новое приближение собственного значения через скалярное произведение (dot(x, Ax)).
   * Вектор x обновляется нормированным значением Ax.
   * Проверяется условие сходимости (Math.abs(lambdaCurrent - lambdaOld) < eps).
   * В случае вырождения (normAx < 1e-30) цикл прерывается.
5. **Применение метода Эйткена для ускорения сходимости:**
   * Если выполнено более двух итераций, то вычисляется улучшенное приближение собственного значения методом Эйткена.
   * Это позволяет уменьшить погрешность вычислений.
6. **Запись результатов в выходной файл output.txt:**
   * Сохраняется количество итераций и информация о сходимости.
   * Выводятся значения, полученные степенным методом и методом Эйткена.
   * Записывается нормированный собственный вектор.

**Описание вспомогательных методов:**

1. **Метод multiply(double[][] A, double[] x):**
   * Реализует умножение матрицы A на вектор x.
   * Возвращает вектор-результат Ax.
2. **Метод dot(double[] a, double[] b):**
   * Вычисляет скалярное произведение двух векторов.
   * Суммирует произведения соответствующих элементов массивов a и b.
3. **Метод norm(double[] v):**
   * Вычисляет евклидову норму (длину) вектора.
   * Использует квадратный корень из суммы квадратов элементов вектора.

**Возможные ошибки и их обработка:**

* Ошибка при чтении входного файла (FileNotFoundException).
* Вырождение матрицы или вектора (Произошло вырождение на итерации).
* Ошибка записи в выходной файл (IOException).

Данная реализация наглядно демонстрирует применение методов численного анализа для задач линейной алгебры и позволяет использовать данный код в учебных или исследовательских проектах, связанных с вычислением собственных значений матриц.