ТЕХНИЧЕСКИЕ НАУКИ

УДК 004.42

Е. А. Аксёнов

МАТРИЧНЫЙ КАЛЬКУЛЯТОР НА ЯЗЫКЕ JAVA

Создание какого бы то ни было прикладного приложения требует от разработчика не только знания языка программирования, но и глубокого понимания темы, затрагиваемой этим приложением. В случае вычислительных программ (*как, например*, *калькулятора матриц*) *это понимание и умение его реализовать на языке программирования играет бóльшую роль, нежели знание самого языка. В разработанном приложении используются базовые языковые средства и принципы объектно-ориентированного программирования, но алгоритмы, которые реализованы этими простыми вещами, являются самыми интересными сложными моментами разработки. Они, как и другие моменты разработанного приложения, и будут рассмотрены в этой статье.*

Сначала кратко рассмотрим, что вообще такое матрицы. Исходя из определения, матрица – это математический объект, записываемый в виде прямоугольной таблицы элементов кольца или поля (например, целых, действительных или комплексных чисел), которая представляет собой совокупность строк и столбцов, на пересечении которых находятся её элементы [1]. На рисунке 1 показана стандартная запись матрицы.

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

Рисунок 1 – Общий вид матрицы

Проще говоря, это таблица размером $m \times n$, каждый элемент которой имеет индекс, состоящий из номера строки i и номера столбца j. Матрицы бывают разных видов, в зависимости от размера и содержимого.

Матрицы широко применяются в математике и физике для компактной записи и решения СЛАУ (систем линейных алгебраических уравнений) и систем дифференциальных уравнений. При этом количество строк матрицы соответствует количеству уравнений системы, а количество столбцов – количеству неизвестных величин. Матричный аппарат позволяет существенно упростить решение СЛАУ, сведя его к операциям над матрицами [2]. Умение работать с матрицами является актуальным навыком

для математиков и физиков, но, даже имея широкие знания и умения, некоторые операции над матрицами большого размера потребуют значительных сил и времени при вычислениях. Тут на помощь и придет данная программа.

В приложении доступны умножение и деление матрицы на число, возведение матрицы в степень, сложение и вычитание матриц, умножение матриц, нахождение обратной и транспонированной матрицы, нахождение определителя и типа матрицы. Знания о проводимых операциях при пользовании программой не потребуются, все ограничения, подсказки и другие указания будут сообщаться пользователю в ходе работы. Рассмотрим как проходит работа с приложением.

После «начального экрана», состоящего из названия программы и подсказок (рисунок 2), пользователь должен ввести свою матрицу. Корректность ввода этой и возможных других матриц контролируется приложением, оно знает все правила и ограничения для создания правильной матрицы в конкретной ситуации. Проверяется количество элементов в строке (оно задается после ввода первой строки и должно быть одинаковым для всех строк), количество строк (умножение матрицы на матрицу и их сложение возможны только для матриц определенных размеров) и сами элементы (они могут не являться числами). При некорректных строках программа не будет завершаться или прерываться, а выведет причину ошибочности строки и будет ожидать следующего ввода.

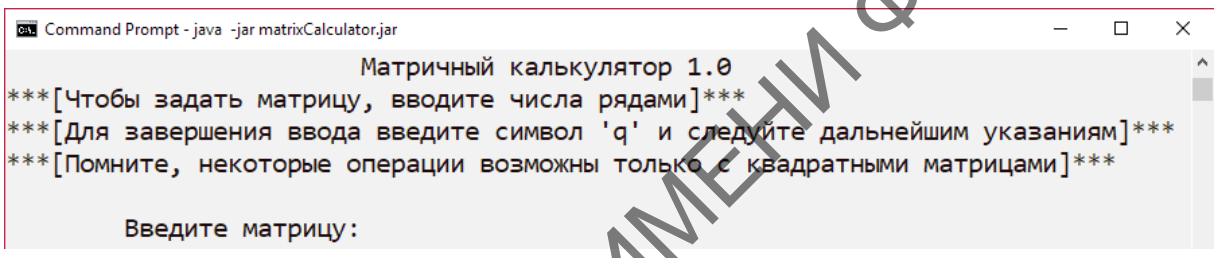

Рисунок 2 – Запуск приложения и ожидание ввода первой матрицы

После завершения ввода в программе сформируется объект класса «Matrix», над которым мы и будет выполнять вычисления. Он отобразится на экране вместе со списком категорий доступных операций (рисунок 3).

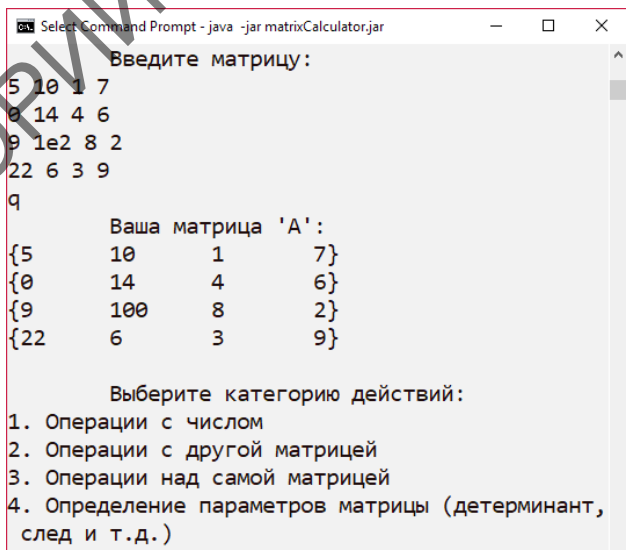

Рисунок 3 – Вывод получившейся матрицы и списка категорий операций

После выбора категории и одного из действий этой категории возможно потребуется ввести «аргумент» данной операции (число или другую матрицу).

Операции умножения и деления матрицы на число, сложения и вычитание матриц, транспонирование и нахождение типа довольно тривиальны для человека и легко реализуются в коде. А вот, например, алгоритм умножения матриц на матрицу в самом упрощенном виде гласит, что «нужно умножить каждую строку первой матрицы на каждый столбец второй матрицы». На деле для этого нам потребуется умножить каждый элемент каждой строки первой матрицы на каждый элемент каждой строки второй матрицы в определенном порядке много раз, провести ряд операций сложения, и, в результате, метод (блок кода в Java), ответственный за умножение, уже содержит в себе тройной вложенный цикл.

Возведение матрицы в степень означает, что нам нужно умножить матрицу на саму себя определенное количество раз. Для этого мы можем просто вызвать метод умножения $n - 1$ раз, где n – нужная для нас степень.

На рисунке 4 показан пример с умножением матрицы на матрицу, в котором, согласно свойству, количество строк у второй матрицы должно равняться количеству столбцов первой. Программа об этом знает и совершит умножение сразу после введения последней возможной строки, не ожидая появления специального символа завершения ввода. После получения результата можно продолжить или прекратить работу, выбрав один из предложенных вариантов.

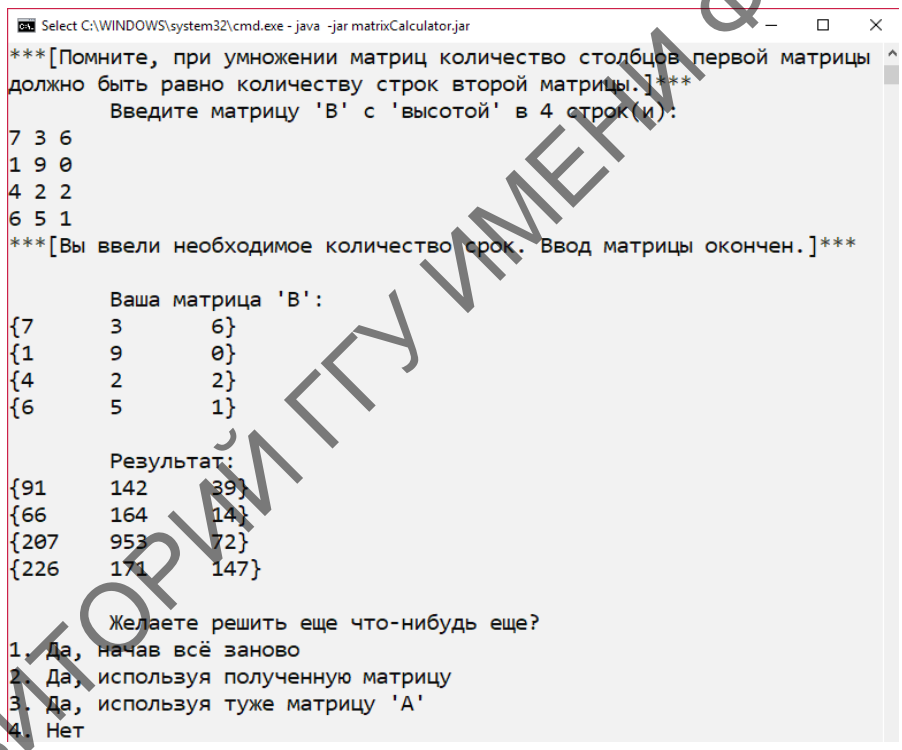

Рисунок 4 – Вывод результата и списка дальнейших доступных действий

Нахождение обратной матрицы и вычисление определителя – это самые трудоемкие операции, особенно для матриц четвертого порядка и выше. Чтобы найти обратную матрицу «4 на 4» методом алгебраических дополнений вам придется найти 16 миноров, а это 16 умноженных на число определителей «3 на 3» или 144 определителя «2 на 2», а это 464 математических операции умножения, деления, сложения, смены знака. Или 17 910 операций для матрицы «6 на 6» или 126 490 910 операций для матрицы «10 на 10» (количество было посчитано этой программой).

Существуют способы ведения расчетов типа метода Гаусса (для нахождения обратной матрицы), правила треугольников (для нахождения определителя «3 на 3») и т. д., несколько облегчающие эти расчеты. Первой мыслью было их и реализовать в

программе. Как оказалось, то, что создано для человека не всегда лучшим образом подходит для компьютерных вычислений. В методе Гаусса нужно с помощью элементарных преобразований привести матрицу к единичному виду, параллельно выполняя те же действия с другой, единичной матрицей. Многие операции и перестановки будут сразу очевидны человеку, примерно знающему, что нужно делать, но для программы нужен или громоздкий алгоритм, пытающийся повторять логику человека или вычисления «напролом», приводящие к нарастающей погрешности (если не реализовать в Java функционал дробей). Метод треугольников вообще подходит только лишь для одного вида матриц. В результате с помощью рекурсии (прием в программировании, когда метод вызывает сам себя) был реализован метод алгебраических дополнений. Он заключается в следующей формуле:

$$A^{-1} = \frac{1}{|A|} \cdot A^T,$$

где A^{-1} – обратная матрица, $|A|$ – определитель матрицы A, A^T – транспонированная матрица алгебраических дополнений соответствующих элементов матрицы А.

Матрица алгебраических дополнений – это матица миноров, но с измененными знаками определенных элементов. Т.е. при нахождении обратной матрицы нам понадобится вычислить некоторое количество определителей. А как найти определитель матрицы n на n? Основной задачей в этом будет нахождение n числа определителей миноров размером m на m, где $m = n - 1$. Как найти определитель матрицы m на m? Нужно будет найти m определителей миноров размером k на k, где $k = m - 1$. И так далее пока не дойдем до размера «2 на 2». Тут мы и получаем на цикл вызовов методов. Для вычисления минора находится определитель, а для вычисления определителя находятся миноры. Полученная матрица превращается в матрицу алгебраических дополнений, транспонируется и затем делится на определитель. Транспонирование и деление на число, как одни из отдельно доступных операций, вынесены в отдельные методы.

Литература

1 Wikipedia [Электронный ресурс] / Матрица (математика) – Википедия. – Режим доступа: https://ru.wikipedia.org/wiki/Матрица_(математика). – Дата доступа: 30.04.2018.

2 Onlinemschool [Электронный ресурс] / Матрицы: определения и основные понятия. – Режим доступа: http://ru.onlinemschool.com/math/library/matrix. – Дата доступа: 30.04.2018.

УДК 519.95

Н. А. Алёшин, Г. Л. Карасёва

НЕВЫРОЖДЕННОСТЬ РЕШЕНИЯ СПЕЦИАЛЬНОЙ ЗАДАЧИ ОПТИМАЛЬНОГО УПРАВЛЕНИЯ

Исследуется специальная задача оптимального управления с фазовыми ограничениями. Введено понятие структуры и определяющих элементов. Получена система нелинейных уравнений доводки. Исследованы достаточные условия, при выполнении которых система уравнений доводки имеет единственное решение, и это решение системы может быть построено методом Ньютона. Сформулирована лемма о невырожденности решения.

В классе кусочно-непрерывных функций рассмотрим задачу оптимального управления с фазовыми ограничениями: