

Задача 4. Матричный калькулятор

Требуется написать калькулятор, осуществляющий операции над матрицами (см. раздел Логика работы).

Входные данные программа должна считывать из текстового файла. Результат должен быть записан в другой текстовый файл в том же формате. Для работы с текстовыми файлами требуется использовать кодировку UTF-8.

Запуск

Программа должна запускаться из командой строки и принимать следующие параметры:

- Параметр 1 - имя файла со входными данными
- Параметр 2 - имя файла с выходными данными

Пример использования (часть с `java -jar ...` здесь опущена):

- `input.txt output.txt`

Логика работы

Программа должна осуществлять одну из двух операций (два независимых варианта):

- Вариант 1 - умножение двух матриц
- Вариант 2 - обращение матрицы

Входные и выходные данные

Текстовое представление каждой из матриц должно следовать следующему формату:

```
Строка 0:<кол-во строк m, int> <кол-во столбцов n, int>
Строка 1:<значение элемента (1,1), float> <значение элемен
та (1,2), float> ... <значение элемента (1,n), float>
...
Строка m:<значение элемента (m,1), float> <значение элемен
та (m,2), float> ... <значение элемента (m,n), float>
Строка m+1:<пустая строка>
```

В качестве десятичного разделителя требуется использовать точку. Кроме того, в случае варианта 2 во входном файле данные матриц будут следовать друг за другом.

Пример текстового представления матрицы:

```
2 3
2.0 0.0 -1.0
3.12 42.0 1.5
```

Рекомендации по реализации программы

В случае успешного выполнения программа должна вывести в консоль сообщение об успехе и завершиться. В случае каких-либо ошибок программа должна вывести сообщение с деталями ошибки и завершиться с ненулевым статус кодом.

Желательно спроектировать иерархию классов в программе с прицелом на разделение ответственности между модулями, повторное использование кода и расширяемость. Вариант разбиения программы на классы:

- Класс Main - считывает параметры программы, проверяет их, читает входной файл и путем манипуляций с объектами других классов выполняет работу программы.
- Класс “десериализатор” - отвечает за преобразование текста (`String` или массива `String[]`) в объект класса “матрица”.
- Класс “сериализатор” - отвечает за преобразование объекта класса “матрица” в текст (`String`).
- Класс “матрица” - содержит метод, отвечающий за операцию калькулятора и возвращающий результат (матрицу с результатом).
- Классы для специфичных для программы исключений - используются для обработки нестандартных ситуаций. При этом стоит построить иерархию исключений с одним общим родительским классом.