

## Наследование и интерфейсы (повторение)

1. Реализовать интерфейс `IMatrix` (квадратная матрица вещественных чисел), содержащий методы:

- получить элемент с заданными индексами,
- изменить элемент с заданными индексами,
- вычислить определитель матрицы (лучше методом Гаусса).

3. Напишите класс `Matrix` (квадратная матрица произвольного вида). Размерность матрицы  $N$  задается при создании объекта и в дальнейшем не меняется. Матрицу хранить в виде одномерного массива длины  $N \times N$ . В классе сделайте конструктор по размерности, добавьте методы `equals` и `hashCode`.

4. Напишите производный класс `DiagMatrix` (диагональная матрица). Размерность матрицы определяется при ее создании и в дальнейшем не меняется. Создайте конструктор по размерности матрицы и конструктор по набору элементов на диагонали. Метод изменения элемента при попытке записать ненулевое значение вне диагонали должен выбрасывать исключение.

5. Напишите класс `UpTriangleMatrix` (верхнетреугольная матрица). Размерность матрицы также задается при ее создании и далее не меняется.

6. Добавить в класс `Matrix` поле для хранения вычисленного значения определителя (кэш для значения определителя) и флаг, который означает, что это значение действительно на данный момент. Методы, изменяющие матрицу, должны сбрасывать этот флаг. При установленном флаге метод вычисления определителя должен просто возвращать сохраненное значение без пересчета.

7. \*Напишите компаратор для матриц, который сравнивает определители матриц.

8. \*Напишите класс `MatrixService` со статическим методом `arrangeMatrices`, который получает на вход массив матриц произвольного вида и сортирует этот массив по неубыванию определителей матриц (найдите в классе `Arrays` подходящий метод сортировки).