**Задание 1.** **Диагональная матрица** — это квадратная матрица, все элементы которой, стоящие вне главной диагонали, равны нулю (читаем тут [Диагональная матрица — Википедия (wikipedia.org)](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%BC%D0%B0%D1%82%D1%80%D0%B8%D1%86%D0%B0)).

Необходимо создать и протестировать класс для представления диагональной матрицы, содержащей целые числа (int).

1. Объект класса хранит только элементы матрицы, расположенные на диагонали. Для этого используется **одномерный** массив.
2. Объект класса имеет свойство только для чтения **Size** – размер матрицы (например, для матрицы 5х5 свойство **Size** возвращает значение 5).
3. Класс располагает конструктором для создания матрицы. В конструктор передаётся список параметров (params) – это элементы матрицы, расположенные на диагонали. Если значение аргумента конструктора не корректно (например, равно null) – создаётся матрица нулевого размера (**Size**=0).
4. Для удобства работы класс предлагает индексатор **с двумя индексами i и j**. Если **i** не равно **j**, то индексатор возвращает значение 0 (при записи – ничего не происходит). Если значения индексов не корректны (выходят за границы: меньше нуля или больше или равны **Size**)– при записи ничего не происходит, а при чтении возвращается значение[[1]](#footnote-1) **0**.
5. Класс матрицы имеет экземплярный метод **Track()**, который возвращает сумму элементов матрицы, расположенных на главной диагонали.
6. Переопределите в классе матрицы методы Equals() и ToString(). Две матрицы считаются равными, если совпадают их размеры и соответствующие элементы на диагонали.
7. Создайте метод расширения для диагональной матрицы, выполняющий сложение двух диагональных матрицы. Результат работы метода – новая диагональная матрица. Если размеры матрицы не совпадают – меньшая матрица дополняется нулями.

**Задание 2.** Компания разрабатывает систему управления тренингами. Каждый тренинг состоит из набора лекций и практических занятий. Необходимо создать классы для представления следующих сущностей: ***тренинг***, ***лекция***, ***практическое занятие***.

Все указанные сущности обладают *текстовым описанием* (это произвольная строка, возможно пустая или равная null).

Лекция имеет *тему* (произвольная строка, возможно пустая или равная null), а практическое занятие – *ссылку на условие задания* (произвольная строка, возможно пустая или равная null) и *ссылку на решение* (произвольная строка, возможно пустая или равная null).

Тренинг должен хранить набор лекций и практических занятий во внутреннем массиве[[2]](#footnote-2) и иметь метод **Add()** для добавления лекции или практического занятия. Кроме этого, тренинг должен иметь метод **IsPractical()** – возвращает true, если тренинг содержит только практические занятия.

Реализуйте в классе тренинга экземплярный метод **Clone()**, выполняющий клонирование тренинга. Внимание: должно выполняться глубокое, а не поверхностное клонирование.

Протестируйте классы в консольном приложении.

1. В этой ситуации, конечно, нужно генерировать исключительную ситуацию. Однако на тренинге исключительные ситуации ещё не рассматривались. [↑](#footnote-ref-1)
2. Этот массив может содержать объекты лекций и практических занятий вперемешку. [↑](#footnote-ref-2)