

Ограничение времени	1 секунда
Ограничение памяти	64.0 Мб
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Детали реализации

Требуется реализовать шаблонный класс `Matrix`, принимающий в качестве шаблонных параметров тип хранимых элементов, число строк матрицы и число столбцов. Так как шаблонные параметры - это константы времени компиляции, элементы матрицы можно хранить в двумерном C-style массиве (`std::array` в этом задании пока запрещен). Это поле-массив должно быть публичным для корректной работы агрегатной инициализации.

Для работы с матрицей предлагается реализовать следующие методы:

- `RowNumber()` и `ColumnNumber()`, возвращающие число строк и столбцов соответственно;
- Операция `()` ("круглые скобки") для доступа к элементам матрицы и возможности их изменения (подумайте, почему не `[]`);
- Метод `At`, аналогичный предыдущему пункту, но бросающий исключение `MatrixOutOfRangeException` в случае выхода за границы матрицы;
- Функцию `GetTransposed(matrix)`, возвращающую транспонированную версию переданной матрицы;
- Операции сложения, вычитания умножения матриц и их присваивающие версии. Обратите внимание, что эти операции применимы не к любым парам матриц!
- Операции умножения и деления на число (слева и справа), а также их присваивающие версии;
- Операции сравнения на равенство и неравенство двух матриц одинакового размера;
- Операции ввода из потока и вывода в поток.

Пример:

```
Matrix<int, 2, 3> a{{1, 2, 3}, {4, 5, 6}};
Matrix<int, 3, 2> b = GetTransposed(a);
// a + b      CE
a + a;        // Ok
// a * a      CE
a * b;        // Ok
// a *= b    // CE - в матрицу a нельзя присвоить матрицу другого размера
b *= Matrix<int, 2, 2>{1, 2, 3, 4}; // Ok

std::cout << a;
1 2 3
4 5 6
```