

Требования к оформлению кода

Интерфейс класса разместить в файле `Matrix.h`. Реализацию методов разместить в файле `Matrix.cpp`.

Полученные файлы поместить в tar-архив и загрузить в систему SmartLMS ([edu.hse.ru](https://edu.hse.ru/mod/assign/view.php?id=683534)) — элемент контроля «СКБ222 - Домашняя работа №4» (<https://edu.hse.ru/mod/assign/view.php?id=683534>).

Ключи компиляции `-std=c++98 -Wall -Werror -pedantic -lm`.

Ограничения по памяти и времени — стандартные.

Задание (базовая часть [△] 8 баллов + дополнительная [◇] 2 балла)

Доработать класс `Matrix` из домашней работы № 3 реализовав:

- △ оператор копирующего присваивания, выполняющий глубокое копирование;
- △ возможность выполнения операций $a+b$, $a-b$ где `Matrix a, b`;
- △ возможность выполнения операции $a*b$, $d*b$, $a*d$, где `Matrix a, b` и `double d`;
- △ возможность выполнения операции $a*b$, где `double a` и `Matrix b`;
- △ возможность выполнения операций `o<<a, i>>a` где `Matrix a`, `std::ostream o`, `std::istream i` (формат данных: число строк и столбцов матрицы через пробел, затем, с новой строки, в каждой строке, элементы матрицы построчно разделенные пробелом);
- ◇ метод получения верхнетреугольной матрицы (приведения матрицы к ступенчатому виду);
- ◇ метод проверки что матрица является треугольной;

В случае возникновения ошибок запускать исключение, производное от `std::exception`.

В случае выполнения дополнительной части определить константу препроцессора `MATRIX_HAS_GAUSS` в файле `Matrix.h`:

```
#ifndef Matrix_h
#define Matrix_h 20221202L

/* ... */

class Matrix {

    /* .. */

public:
    Matrix();
    Matrix(const Matrix&);
    Matrix(size_t rows, size_t cols);
    ~Matrix();

    Matrix& operator=(const Matrix&);

    /* ... */

    double& at(size_t r, size_t c) ;
    const double& at(size_t r, size_t c) const ;

    /* ... */

    void Transpose();

    /* ... */

#ifdef MATRIX_HAS_GAUSS
    Matrix Upper() const;
    bool is_triangular() const;
#endif /*MATRIX_HAS_GAUSS*/

};

#endif /*Matrix_h*/
```