

## Семестр 3 (2019), Контрольная работа (вариант 1)

Требуется разработать класс `Matrix2`, реализующий работу с вещественными матрицами размера  $2 \times 2$ , а также программу, тестирующую методы этого класса.

### Описание класса

Конструктор по умолчанию создает нулевую матрицу.

Конструктор преобразования `Matrix2(double a)` создает диагональную матрицу. Параметр `a` задает значение элемента матрицы на диагонали

Конструктор `Matrix2(double a11, double a12, double a21, double a22)` задает матрицу вида

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}.$$

Должна быть реализована операция умножения `*` двух матриц в виде константного метода класса.

Должна быть реализована операция вызова функции с двумя целочисленными аргументами в виде константного метода класса. Значением выражения `m(i,j)` является элемент матрицы, стоящий на пересечении строки `i` и столбца `j`. Задание некорректного номера строки или столбца должно приводить к возбуждению исключения.

Должна быть реализована дружественная функция `print(const Matrix&)`, печатающая текстовое представление экземпляра класса в стандартный поток вывода.

### Требования к составу программы

Файл `matrix2.h` должен содержать описание класса.

Файл `matrix2.cpp` должен содержать реализацию методов класса.

Файл `main.cpp` должен содержать вызовы тестов.

Должен присутствовать файл `Makefile`. Программа должна собираться с помощью утилиты `make`.

### Требования к тестам

Должно быть реализовано четыре теста (на каждый конструктор и операцию `*`). Программа должна последовательно вызывать друг за другом все тесты. Каждый тест должен быть реализован в отдельной функции.

Пример теста для конструктора преобразования.

```
#include <assert.h>
...
void test2()
{
    printf("Test 2.\n");

    Matrix2 m(2.0);

    assert(m(1,1) == 2. &&
           m(1,2) == 0. &&
           m(2,1) == 0. &&
           m(2,2) == 2.);

    print(m);
}
```