#include <iostream>

#include <vector>

class Matrix {

private:

std::vector<std::vector<int>> data;

int rows;

int cols;

public:

Matrix(int r, int c) : rows(r), cols(c) {

data.resize(rows);

for (int i = 0; i < rows; i++) {

data[i].resize(cols);

}

}

Matrix(const std::vector<std::vector<int>>& d) : data(d), rows(d.size()), cols(d[0].size()) {}

Matrix operator+(const Matrix& other) const {

if (rows != other.rows || cols != other.cols) {

throw std::invalid\_argument("Matrices must have same dimensions");

}

Matrix result(rows, cols);

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

result.data[i][j] = data[i][j] + other.data[i][j];

}

}

return result;

}

Matrix operator-(const Matrix& other) const {

if (rows != other.rows || cols != other.cols) {

throw std::invalid\_argument("Matrices must have same dimensions");

}

Matrix result(rows, cols);

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

result.data[i][j] = data[i][j] - other.data[i][j];

}

}

return result;

}

Matrix operator\*(const Matrix& other) const {

if (cols != other.rows) {

throw std::invalid\_argument("Number of columns of first matrix must be equal to number of rows of second matrix");

}

Matrix result(rows, other.cols);

for (int i = 0; i < rows; i++) {

for (int j = 0; j < other.cols; j++) {

for (int k = 0; k < cols; k++) {

result.data[i][j] += data[i][k] \* other.data[k][j];

}

}

}

return result;

}

Matrix operator/(int divisor) const {

Matrix result(rows, cols);

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

result.data[i][j] = data[i][j] / divisor;

}

}

return result;

}

friend std::ostream& operator<<(std::ostream& os, const Matrix& m) {

for (int i = 0; i < m.rows; i++) {

for (int j = 0; j < m.cols; j++) {

os << m.data[i][j] << " ";

}

os << "\n";

}

return os; // Заполнение массива

}

};

#include <iostream>

#include "matrix.h"

using namespace std;

int main() {

Matrix a({ {1, 2}, {3, 4} });

Matrix b({ {5, 6}, {7, 8} });

Matrix c = a + b;// сложение матриц

cout << c;

Matrix d = a - b;// вычитание матриц

cout << d;

Matrix e = a \* b;//умножение матриц

cout << e;

Matrix f = a; // 2;деление матрицы на число 2

cout << f;

Matrix z = a \* 2; //умножение матрицы на число 2

cout << z;

return 0;

}