

```

#ifndef MY_MATRIX3
#define MY_MATRIX3

#include "MyVector3D.h"

class MyMatrix3
{
public:
    MyMatrix3();
    MyMatrix3(
        double a11, double a12, double a13,
        double a21, double a22, double a23,
        double a31, double a32, double a33);
    MyMatrix3(MyVector3D row1, MyVector3D row2, MyVector3D row3);
    ~MyMatrix3();
    std::string toString()const;
    MyMatrix3 operator +(const MyMatrix3 other) const;
    MyMatrix3 operator -(const MyMatrix3 other) const;
    MyMatrix3 operator *(const MyMatrix3 other) const;
    MyVector3D operator *(const MyVector3D vector)const;
    MyMatrix3 operator *(const double scale)const;

    MyMatrix3 transpose()const;
    double determinant() const;
    MyMatrix3 inverse() const;

    MyVector3D row(int row)const;
    MyVector3D column(int column) const;

    bool operator ==(const MyMatrix3 other)const;
    bool operator !=(const MyMatrix3 other)const;
    static MyMatrix3 rotationZ(double angleRadians);
    static MyMatrix3 rotationY(double angleRadians);
    static MyMatrix3 rotationX(double angleRadians);

    static MyMatrix3 translation(MyVector3D displacement); // 2d translation make
    sure z=1
    static MyMatrix3 scale(double scalingfactor);

private:
    double m11;
    double m12;
    double m13;
    double m21;
    double m22;
    double m23;
    double m31;
    double m32;
    double m33;
};

MyVector3D operator *(const MyVector3D vector, const MyMatrix3 matrix);

#endif // !MY_MATRIX3

```