# Оператор дахин тодорхойлох лабораторийн ажил (Лаборатори №10)

М.Солонго

ХШУИС, МКУТ-ийн оюутан, 20b1num2104@stud.num.edu.mn

# 1. ОРШИЛ

Матрицан мэдээллийг боловсруулах matrix класс байгуулан private хандалтын түвшинтэй гишүүн өгөгдөл зарлан түүнд хандах public түвшинтэй getter setter функцийг тодорхойлж, “+,\*,-,=,++,--,+=,-=,\*=” операторуудыг дахин тодорхойлж матрицуудыг хооронд нь үржүүлж хасаж нэмэх зэрэг үйлдлүүдийг хийхийн тулд оператор хэрхэн тодорхойлдогийг судлан шаардлагатай онолын судалгааг кодын хамт хавсаргав.

# 2. ЗОРИЛГО

Оператор дахин тодорхойлж сурахын тулд дараах зорилтуудыг тавьж ажилласан. Үүнд:

1. Матрикс класс байгуулж түүний гишүүн өгөгдөл, гишүүн функцуудыг тодорхойлж бичих
2. Асуултын дагуу операторуудыг дахин тодорхойлох
3. Тайлбар бичих

# 3. ОНОЛЫН СУДАЛГАА

## 3.1 Оператор дахин тодорхойлох

Main(){

Int a,b=2,c=1;

a=b+c;

}

Уг үйлдлийг компайлер ойлгон үйлдлийг хийн хариу буцаах бол

Class Matrix{

…

}

Main(){

Matrix a,b,c;

a=b+c;

}

Үйлдлийг ойлгохгүй. Ийм учраас хэрэглэгч нэмэх операторыг дахин тодорхойлж болдог байна. Оператор дахин тодорхойлохдоо:

Буцаах\_утга **operator** дахин тодорхойлох оператор(параметр авах утга){

…

}

# 4. ХЭРЭГЖҮҮЛЭЛТ

## 4.1. Ямар операторуудыг дахин тодорхойлж болохгүй вэ?

Шууд хандалтын оператор (.)

Үйлчлэх хүрээний оператор (::)

Sizeof()

Нөхцөлт оператор (?:)

# 5. ДҮГНЭЛТ

Тухайн бичиж буй кодын онцлогоос хамааран операторыг янз бүрээр дахин тодорхойлж, ашиглаж, өөрт ашигтайгаар кодыг бичиж болно.

## 6. АШИГЛАСАН МАТЕРИАЛ

1. Объект хандлагат технологийн С++ програмчлал, Ж.Пүрэв, 2008, Улаанбаатар.
2. [C Programming Matrix Multiplication | C Program for Matrix Manipulation (educba.com)](https://www.educba.com/c-programming-matrix-multiplication/)

## 7. ХАВСРАЛТ

Matrix.cpp:

*#include* "Matrix.h"

*#include*<iostream>

using namespace std;

    Matrix::Matrix(float \*\_values, int \_n, int \_m) { *//параметртэй байгуулагч функц тодорхойлж байна*

        setM(\_m); *//классын m хувьсагчид утга дамжуулж байна*

        setN(\_n); *//классын n хувьсагчид утга дамжуулж байна*

*this*->values=new float\*[\_n]; *//хаяган хувьсагчид n хаяган санах ой  зарлаж өгч байна*

*for*(int i=0;i<\_n;i++){

            values[i]=new float[\_m]; *//нөөцөлсөн хаяган хүснэгтдээ дахин санах ой нөөцөлж байна*

        }

        int k=0;

*for*(int i=0;i<\_n;i++){

*for*(int j=0;j<\_m;j++){

*this*->values[i][j]=\_values[k++]; *//нөөцөлсөн 2 хэмжээст массивдаа параметрээс орж ирсэн хүснэгтийн элементийг хуулах*

            }

        }

    }

    Matrix::Matrix(int \_n, int \_m) { *//параметртэй байгуулагч функц тодорхойлж байна*

        setM(\_m);

        setN(\_n);

*this*->values=new float\*[\_n];

*for*(int i=0;i<\_n;i++){

            values[i]=new float[\_m];

        }

    }

    Matrix::~Matrix(){ *//устгагч функц тодорхойлолт*

*for*(int i=0; i<n; i++){

            delete[] values[i]; *//динамикаар нөөцөлсөн санах ойг устгаж байна*

        }

        delete[] values;

    }

    int Matrix::getM() { *//getM функцийг тодорхойлж байна int утга буцаана*

*return* m;

    }

    void Matrix::setM(int \_m) { *//SetM функцийг тодорхойлж байна*

*this*->m = \_m;

    }

    int Matrix::getN() { *//getN функцийг тодорхойлж байна int утга буцаана*

*return* n;

    }

    void Matrix::setN(int \_n) { *//setN функцийг тодорхойлж байна*

*this*->n = \_n;

    }

    void Matrix::setValues(float \_values,int \_n,int \_m){ *//SetValues функцийг тодорхойлж байна*

*this*->values[\_n][\_m]=\_values; *//\_n \_m дэх индексэд \_values тоог оруулна*

    }

    float Matrix::getValues(int i,int j){ *//getValues функцийг тодорхойлж байна*

*return* *this*->values[i][j]; *//i j индекс дээр хадгалагдаж буй утгыг буцаана*

    }

    Matrix Matrix::operator+(Matrix b){ *//+ оператор дахин тодорхойлж байна матрикс буцаана*

        Matrix c(n,m); *//а б матрицыг нэмэхийн тулд с матриц зарлаж байна*

        Matrix tmp(1,1); *//1 1 хэмжээтэй матриц зарлаж байна*

*if*(*this*->n==b.getN()&&*this*->m==b.getM()){ *//нэмэх гэж буй 2 матриц-н мөр багана тэнцүү биш бол нэмэхэд асуудал гарах учир нөхцөлөөр шалгана*

*for*(int i=0;i<*this*->getN();i++){

*for*(int j=0;j<*this*->getM();j++){

                        c.values[i][j]=*this*->values[i][j]+b.values[i][j];*//а матриц-н 2 хэмжээстэй хүснэгтийн элемент бүр дээр б матриц-н элементийг нэмж байна*

                    }

                }

*return* c;

        }*else*{

*return* tmp; *//аль нэг мөр багана тэнцүү биш бол 1 1 хэмжээтэй матриц буцаана*

        }

    }

    Matrix Matrix::operator\*(Matrix b){ *//\* операторыг дахин тодорхойлж байна*

        Matrix c(n,n); *//а б матрицыг үржихийн тулд с матриц зарлаж байна*

        Matrix tmp(1,1); *//1 1 хэмжээтэй матриц зарлаж байна*

        int s=0;

*if*(*this*->n == b.m && *this*->m == b.n){

*for*(int i=0;i<*this*->n;i++){

*for*(int j=0;j<*this*->n;j++){

*for*(int k=0;k<*this*->m;k++){

                        s=s+values[i][k]\*values[k][j]; *//мөрийн дагуу элемент баганын дагуу элэментүүдийг үржүүлж S хувьсагчид хадгална*

                    }

                    c.values[i][j]=s; *//хадгалсан тоогоо харгалзах байрлалд оруулж хадгална*

                    s=0; *//S-г тэглэнэ*

                }

            }

*return* c; *//с матрицыг буцаана*

        }*else*{

            cout<<"error"<<endl; *//үржүүлж болохгүй учир error гэж дэлгэцэнд хэвлэж харуулна*

*return* tmp; *//1 1 хэмжээтэй матрицыг буцаана*

        }

    }

    Matrix Matrix::operator-(Matrix b){ *//- операторыг дахин тодорхойлж байна*

        Matrix c(n,m);

        Matrix tmp; *//1 1 хэмжээтэй матриц зарлаж байна*

*if*(n==b.getN()&&m==b.getM()){ *//нэмэх гэж буй 2 матриц-н мөр багана тэнцүү биш бол нэмэхэд асуудал гарах учир нөхцөлөөр шалгана*

*for*(int i=0;i<*this*->getN();i++){

*for*(int j=0;j<*this*->getM();j++){

                        c.values[i][j]=*this*->values[i][j]-b.values[i][j];*//а матрицын бүх харгалзах элементээс б матрицын элементийг хасаж байна*

                    }

                }

*return* c; *//с матрицыг буцаана*

        }*else*{

*return* tmp; *//хасаж буй 2 матриц адил мөр баганатай биш байвал 1 1 хэмжээтэй матрицыг буцаана*

        }

    }

    void Matrix::operator=(Matrix b){ *//= операторыг дахин тодорхойлж байна*

        Matrix tmp(1, 1); *//1 1 хэмжээтэй матриц зарлаж байна*

*if*(*this*->n==b.getN()&&*this*->m==b.getM()){

*for*(int i=0;i<*this*->n;i++){

*for*(int j=0;j<*this*->m;j++){

*this*->values[i][j]=b.values[i][j]; *//матрицын бүх элементэд б матрицын элементийг оногдуулж байна*

                }

            }

        }*else*{

            cout<<"error '='";

        }

    }

    void Matrix::operator++(void){ *//++ операторыг дахин тодорхойлж байна*

*for*(int i=0;i<*this*->n;i++){

*for*(int j=0;j<*this*->m;j++){

*this*->values[i][j]=*this*->values[i][j]+1; *//матрицын бүх элементийг 1-ээр хорогдуулж байна*

            }

        }

    }

    void Matrix::operator--(void){ *//-- операторыг дахин тодорхойлж байна*

*for*(int i=0;i<*this*->n;i++){

*for*(int j=0;j<*this*->m;j++){

*this*->values[i][j]=*this*->values[i][j]-1; *//матрицын бүх элементийг 1-ээр хорогдуулж байна*

            }

        }

    }

    void Matrix::operator+=(int a){ *//+= операторыг дахин тодорхойлж байна*

*for*(int i=0;i<*this*->n;i++){

*for*(int j=0;j<*this*->m;j++){

*this*->values[i][j]+=a; *//матрицын бүх элементийг a хувьсагчаар нэмэгдүүлж байна*

            }

        }

    }

    void Matrix::operator-=(int a){ *//-= операторыг дахин тодорхойлж байна*

*for*(int i=0;i<*this*->n;i++){

*for*(int j=0;j<*this*->m;j++){

*this*->values[i][j]-=a; *//матрицын бүх элементийг a хувьсагчаар хорогдуулж байна*

            }

        }

    }

    void Matrix::operator\*=(int a){ *//\*= операторыг дахин тодорхойлж байна*

*for*(int i=0;i<*this*->n;i++){

*for*(int j=0;j<*this*->m;j++){

*this*->values[i][j]\*=a; *//матрицын бүх элементийг a хувьсагчаар үржүүлж байна*

            }

        }

    }

    void Matrix::showData(void){ *//мэдээлэл дэлгэцэлж харуулах функц тодорхойлж байна*

*if*(*this*->n==1 && *this*->m==1){ *//матриц 1 1 хэмжээтэй байвал error гэж хэвлэж харуулна*

            cout<<"error to display"<<endl;

        }*else*{

*for*(int i=0;i<*this*->n;i++){

*for*(int j=0;j<*this*->m;j++){

                    cout<<*this*->getValues(i,j)<<" ";

                }

                cout<<endl;

            }

        }

    }

Matrix.h:

*#ifndef* \_\_Matrix\_\_

*#define* \_\_Matrix\_\_

*#include*<iostream>

using namespace std;

class Matrix {

    private:

    int m,n;

    float \*\*values;

    public:

    Matrix(int m=1, int n=1);

    Matrix(float \*\_values, int \_n, int \_m);

    ~Matrix();

    int getM();

    int getN();

    void setM(int \_m);

    void setN(int \_n);

    void setValues(float \_values,int \_n,int \_m);

    float getValues(int i,int j);

    void showData(void);

    Matrix operator+(Matrix b);

    Matrix operator\*(Matrix b);

    Matrix operator-(Matrix b);

    void operator=(Matrix b);

    void operator++(void);

    void operator--(void);

    void operator+=(int a);

    void operator-=(int a);

    void operator\*=(int a);

};

*#endif*

Main.cpp:

*#include*<iostream>

*#include* "Matrix.h"

using namespace std;

main(){

    int rows,col; *//мөр багана хадгалах хувьсагч зарлаж байна*

    cout<<"rows: ";

    cin>>rows; *//гараас мөр хувьсагчид утга авна*

    cout<<"columns: ";

    cin>>col; *//гараас багана хуьсагчид утга авна*

    int n=rows\*col;

    float values[n]; *//матрицын элементүүдийг хадгалах массив зарлаж байна*

    cout<<"a matrix: ";

*for*(int i=0;i<n;i++){

        cin>>values[i]; *//массивт гараас утга авч байна*

    }

    Matrix a(values,rows,col); *//а матриц класс зарлан массив болон мөр баганын хэмжээг дамжуулж өгнө*

    float values1[n];

    cout<<"b matrix: ";

*for*(int i=0;i<n;i++){

        cin>>values1[i];

    }

    Matrix b(values1,rows,col); *//б матриц класс зарлан массив болон мөр баганын хэмжээг дамжуулж өгнө*

    cout<<"----------------------"<<endl;

    cout<<"a+b: "<<endl;

    Matrix c=a+b; *//с матриц зарлан а+б хариуг хэвлэж харуулна*

    c.showData();

    cout<<"----------------------"<<endl;

    cout<<"a-b: "<<endl;

    Matrix d=a-b; *//d матриц зарлан а-б хариуг хэвлэж харуулна*

    d.showData();

    cout<<"----------------------"<<endl;

    cout<<"a\*b: "<<endl;

    Matrix e=a\*b; *//e матриц зарлан а\*б хариуг хэвлэж харуулна*

    e.showData();

    cout<<"----------------------"<<endl;

    cout<<"a=b: "<<endl;

    a=b; *//а матриц-д б матриц-г оноох*

    a.showData();

    cout<<"----------------------"<<endl;

    cout<<"a++: "<<endl;

    ++a; *//а матриц-г 1-ээр нэмэгдүүлж байна*

    a.showData();

    cout<<"----------------------"<<endl;

    cout<<"a--: "<<endl;

    --a; *//а матриц-г 1-ээр хорогдуулж байна*

    a.showData();

    cout<<"----------------------"<<endl;

    cout<<"a+=2: "<<endl;

    a+=2; *//а матриц-г 2-оор нэмэгдүүлж байна*

    a.showData();

    cout<<"----------------------"<<endl;

    cout<<"a-=3: "<<endl;

    a-=3; *//а матриц-г 3-аар хорогдуулж байна*

    a.showData();

    cout<<"----------------------"<<endl;

    cout<<"a\*=2: "<<endl;

    a\*=2; *//а матриц-г 2 дахин нэмэгдүүлж байна*

    a.showData();

}