

# Хоёр хэмжээст матрицуудын хооронд дахин тодорхойлсон оператор ашиглан үйлдэл хийх (Лаборатори №10)

Ү.Мэндсайхан

МУИС, ХШУИС, Компьютерын ухаан, mendsaikhan@gurvanshiidel.mn

## 1. ОРШИЛ

Энэхүү лабораторийнн ажилаар оператор дахин тодорхойлох талаар судлаж матрицийг илэрхийлэх класс зарлаж өгсөн. Мөн  $+$ ,  $-$ ,  $*$ ,  $=$ ,  $++$ ,  $--$ ,  $+=$ ,  $*=$  гэсэн операторуудийг матриц дээр үйлдэл хийж болохоор классд дахин тодорхойлж өгсөн.

## 2. ЗОРИЛГО

Оператор дахин тодорхойлох талаар судлаж  $+$ ,  $-$ ,  $*$ ,  $=$ ,  $++$ ,  $--$ ,  $+=$ ,  $-=$ ,  $*=$  гэх мэт операторуудыг дахин тодорхойлно. Үүний тулд дараах зорилтуудыг тавьж ажилласан.

2.1 Оператор дахин тодорхойлох талаар судлах

2.2 Хос болон ганц операндтай операторыг дахин тодорхойлох талаар судлах

2.3 Матрицын мөр болон баганыг солих оператор дахин тодорхойлох.

## 3. ОНОЛЫН СУДАЛГАА

### 3.1 Оператор дахин тодорхойлох

Оператор дахин тодорхойлох гэдэг нь C++ хэлэнд байдаг default операторуудаас гадна классийн түвшинд объектоос объектод үйлдэл хийж болохуйц операторыг дахин тодорхойлох гэсэн үг юм. Оператор дахин тодорхойлохдоо C++ хэлэнд байдаг операторуудаас дахин тодорхойлж өгөх ёстой байдаг. Мөн оператор дахин тодорхойлохдоо ганц операндтай оператор болон хос операндтай оператор дахин тодорхойлох гэсэн 2 арга байдаг. Хоёр объектын гишүүн өгөгдлүүд болох хоёр тэмдэгт мөрийг залгахдаа  $+$  операторыг дахин тодорхойлох боломжтой юм. Учир нь хоёр тэмдэгтийг шууд  $+$  оператораар залгах боломжгүй байдаг. Мөн объект нь хэд хэдэн төрлийн гишүүн өгөгдөл хадгалж байж болох учраас операторуудыг дахин тодорхойлж өгөх зүйтэй юм.

## 4. ХЭРЭГЖҮҮЛЭЛТ

### 1. Матриц set гишүүн функц

```
void Matrix::setValues(float **a) {
    int i, j;
    for (i = 0; i < mor; i++) {
        for (j = 0; j < bagana; j++) {
            _values[i][j] = a[i][j];
        }
    }
}
```

### 2. Матриц get гишүүн функц

```
float **Matrix::getValues() {
    return _values;
}
```

### 3. Матрицийг матриц дээр нэмэх + операторийг тодорхойлж өгсөн

```
Matrix Matrix::operator+(Matrix mat) {
    if (bagana == mat.getRow() && mor == mat.getColumn()) {
        Matrix new_Matrix(mor, bagana);
        int i, j;
        for (i = 0; i < mor; i++) {
            for (j = 0; j < bagana; j++) {
                new_Matrix._values[i][j] = _values[i][j] + mat._values[i][j];
            }
        }
        return new_Matrix;
    } else {
        cout << "2 matriltiin hemjee tohirsongui" << endl;
        return *this;
    }
}
```

### 4. Матрицийг матрицаас хасах - операторийг тодорхойлж өгсөн

```
Matrix Matrix::operator-(Matrix mat) {
    if (bagana == mat.getRow() && mor == mat.getColumn()) {
        Matrix new_Matrix(mor, bagana);
        int i, j;
        for (i = 0; i < mor; i++) {
            for (j = 0; j < bagana; j++) {
                new_Matrix._values[i][j] = _values[i][j] - mat._values[i][j];
            }
        }
        return new_Matrix;
    } else {
        cout << "2 matriltiin hemjee tohirsongui" << endl;
        return *this;
    }
}
```

5. Матриц дээр бүхэл тоо нэмэх + операторийг тодорхойлж өгсөн

```
Matrix Matrix::operator+(float a) {
    Matrix new_Matrix(mor, bagana);
    int i, j;
    for (i = 0; i < mor; i++) {
        for (j = 0; j < bagana; j++) {
            new_Matrix._values[i][j] = _values[i][j] + a;
        }
    }
    return new_Matrix;
}
```

6. Матрицийг матрицаар үржүүлэг \* операторийг тодорхойлж өгсөн

```
Matrix Matrix::operator*(Matrix mat) {
    if (bagana == mat.getRow()) {
        int i, j, mSize, s, nilber, bagana;
        mSize = mor * bagana;
        Matrix new_Matrix(mor, mat.getColumn());
        for (i = 0; i < mor; i++) {
            for (bagana = 0; bagana < mat.getColumn(); bagana++) {
                nilber = 0;
                for (j = 0; j < bagana; j++) {
                    s = _values[i][j] * mat._values[j][i];
                    nilber = nilber + s;
                }
                new_Matrix._values[i][bagana] = nilber;
            }
        }
        return new_Matrix;
    } else {
        cout << "\n2 matritsiin hemjee tohirsongui:" << endl;
        return *this;
    }
}
```

7. Матрицийг матрицад оноох = операторийг тодорхойлж өгсөн

```
void Matrix::operator=(Matrix mat) {
    if (bagana == mat.getColumn() && mor == mat.getRow()) {
        for (int i = 0; i < mor; i++) {
            for (int j = 0; j < bagana; j++) {
                _values[i][j] = mat._values[i][j];
            }
        }
    } else {
        cout << "2 matritsiin hemjee tohirsongui" << endl;
    }
}
```

8. Нэг операндтай ++ операторийг дахин тодорхойлж өгсөн

```
//Матрицыг нэгээр нэмэгдүүлэх
void Matrix::operator++() {
    for (int i = 0; i < mor; i++) {
        for (int j = 0; j < bagana; j++) {
            _values[i][j]++;
        }
    }
}
```

9. Нэг операндтай -- операторийг дахин тодорхойлж өгсөн

```
//Матрицыг нэгээр хорогдуулах
void Matrix::operator--() {
    for (int i = 0; i < mor; i++) {
        for (int j = 0; j < bagana; j++) {
            _values[i][j]--;
        }
    }
}
```

10. Матрицийг матрицаар нэмэгдүүлэх += операторийг дахин тодорхойлж өгсөн

```
void Matrix::operator+=(float a) {
    for (int i = 0; i < mor; i++) {
        for (int j = 0; j < bagana; j++) {
            _values[i][j] += a;
        }
    }
}
```

11. Матрицийг матрицаар хорогдуулах -= операторийг дахин тодорхойлж өгсөн

```
void Matrix::operator-=(float a) {
    for (int i = 0; i < mor; i++)
        for (int j = 0; j < bagana; j++)
            _values[i][j] -= a;
}
```

12. Матрицийг элементүүдийг а тоогоор үржүүлэх \*= операторийг дахин тодорхойлж өгсөн

```
void Matrix::operator*=(float a) {
    for (int i = 0; i < mor; i++)
        for (int j = 0; j < bagana; j++)
            _values[i][j] *= a;
}
```

13. Матрицийн баганыг мөр лүү, мөрийг багана луу хөрвүүлэх гишүүн функц тодорхойлж өгсөн

```
Matrix Matrix::operator~() {
    Matrix copyMatrix(bagana, mor); //Шинэ объект үүсгэх
    for (int i = 0; i < mor; i++) {
        for (int j = 0; j < bagana; j++) {
            copyMatrix._values[j][i] = _values[i][j]; //Матриц хөрвүүлэх
        }
    }
    return copyMatrix; //Хөрвүүлсэн матрицаа буцаах
}
```

## 5. ДҮГНЭЛТ

Операторыг дахин тодорхойлж өгснөөр бид объектын түвшинд олон боломж бий болох юм. Жишээ нь классын гишүүн өгөгдөл бүр дээр объект дуудан ажиллах боломжтой. Бүхэл тоон дээр бүхэл тоо нэмж, тэмдэгт мөрийг тэмдэгт мөрд нэмж залгах гэх мэт үйлдлүүдийг зэрэг хийх боломж нээгдэнэ.

## 6. АШИГЛАСАН МАТЕРИАЛ

1. Объект хандлагат технологийн C++ програмчлал, Ж.Пүрэв, 2008, Улаанбаатар

## 7. ХАВСРАЛТ

Github линк, [Lab 10](#)