2022-2 객체지향형 프로그래밍과 자료구조 Exam2A

2022. 10. 21.
담당교수 김영탁

2A. class Mtrx, class MtrxArray

2A.1 class Mtrx

```
/* Class_Mtrx.h */
. . . .  // 필요한 헤더파일 추가
class Mtrx {
        friend ostream& operator<<(ostream& fout, const Mtrx& m);
        friend istream& operator>>(istream& fin, Mtrx& m);
public:
        Mtrx();
        Mtrx(int num_row, int num_col);
        Mtrx(istream& fin);
        ~Mtrx(); // destructor
        int getN_row() { return n_row; }
        int getN_col() { return n_col; }
        string getName() { return name; };
        void setName(string nm) { name = nm;};
        Mtrx operator+(const Mtrx&);
        Mtrx operator+(double a);
        Mtrx operator-(const Mtrx&);
        Mtrx operator-(double a);
        Mtrx operator*(const Mtrx&);
        Mtrx operator*(double a);
        Mtrx operator~(); // transposed matrix
private:
        string name;
        int n_row;
        int n_col;
        double **dM;
};
```

- class Mtrx는 행렬 (matrix) 연산을 위한 클래스이며, 행렬의 덧셈, 뺄셈, 곱셈, 전치 (transpose) 연산과 입력 및 출력을 위한 연산을 연산자 오버로딩으로 제공한다.
- class Mtrx의 덧셈, 뺄셈, 곱셈 연산에 class Mtrx 인스탄스가 인수로 제공되면 두 행렬의 덧셈, 뺄셈, 곱셈 연산을 실행하여 생성된 결과 행렬을 반환한다.
- class Mtrx의 덧셈, 뺄셈, 곱셈 연산에 double 자료형 실수가 인수로 제공되면 행렬의 각 원소에 해당 실수 값의 덧셈, 뺄셈, 곱셈 연산을 실행하여 생성된 행렬을 반환한다.
- class Mtrx의 멤버함수들과 관련 함수들은 Class_Mtrx.h에 inline 함수로 구현 또는 Class_Mtrx.cpp 파일에 작성할 것.

2A.2 class MtrxArray

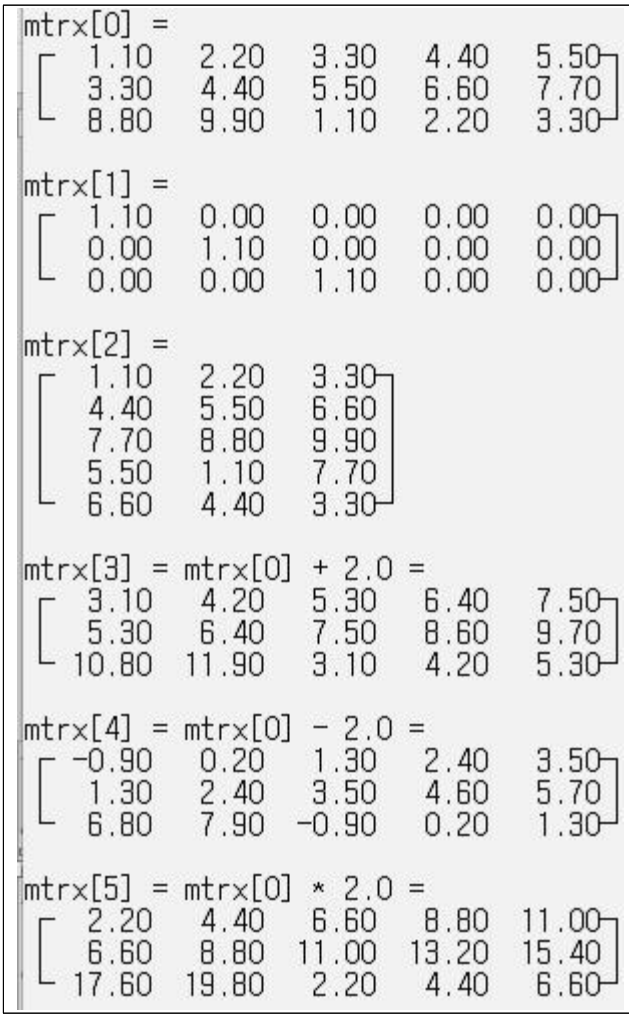
```
/* Class_MtrxArray.h */
. . . .  // 필요한 헤더파일 추가
class MtrxArray
{
public:
        MtrxArray(int array_size); // constructor
        ~MtrxArray();   // destructor
        Mtrx &operator[](int);
private:
        Mtrx *pMtrx;
        int mtrxArrySize;
        bool isValidIndex(int index);
};
```

- class MtrxArray는 class Mtrx의 인스탄스들을 배열로 구성하여 사용할 수 있도록 한다.
- class MtrxArray에는 배열에 포함된 원소(class Mtrx의 인스탄스)를 인덱스를 사용하여 접근할 수 있도록 인덱싱 연산자 ([])의 연산자 오버로딩을 기능을 제공한다.
- class MtrxArray의 멤버함수들과 관련 함수들은 Class_MtrxArray.h에 inline 함수로 구현 또는 Class_MtrxArray.cpp 파일에 작성할 것.

2A.3 class Mtrx와 class MtrxArray 인스탄스들을 활용하는 시험 프로그램

```
. . .  // include necessary header files and definitions
void main()
{
	ifstream fin;
	ofstream fout;

	fin.open("Matrix_data.txt");
	if (fin.fail())
	{
		cout << "Error in opening Matrix_data.txt !!" << endl;
		exit;
	}

	fout.open("output.txt");
	if (fout.fail())
	{
		cout << "Error in opening output.txt !!" << endl;
		exit;
	}

	MtrxArray mtrx(10);
	fin >> mtrx[0] >> mtrx[1] >> mtrx[2];
	mtrx[0].setName("mtrx[0]");
	mtrx[1].setName("mtrx[1]");
	mtrx[2].setName("mtrx[2]");
	cout << mtrx[0]; fout << mtrx[0];
	cout << mtrx[1]; fout << mtrx[1];
	cout << mtrx[2]; fout << mtrx[2];

	mtrx[3] = mtrx[0] + 2.0;
	mtrx[3].setName("mtrx[3] = mtrx[0] + 2.0");
	cout << mtrx[3]; fout << mtrx[3];
	mtrx[4] = mtrx[0] - 2.0;
	mtrx[4].setName("mtrx[4] = mtrx[0] - 2.0");
	cout << mtrx[4]; fout << mtrx[4];
	mtrx[5] = mtrx[0] * 2.0;
	mtrx[5].setName("mtrx[5] = mtrx[0] * 2.0");
	cout << mtrx[5]; fout << mtrx[5];

	mtrx[6] = mtrx[0] + mtrx[1];
	mtrx[6].setName("mtrx[6] = mtrx[0] + mtrx[1]");
	cout << mtrx[6]; fout << mtrx[6];
	mtrx[7] = mtrx[0] - mtrx[1];
	mtrx[7].setName("mtrx[7] = mtrx[0] - mtrx[1]");
	cout << mtrx[7]; fout << mtrx[7];
	mtrx[8] = mtrx[0] * mtrx[2];
	mtrx[8].setName("mtrx[8] = mtrx[0] * mtrx[2]");
	cout << mtrx[8]; fout << mtrx[8];
	mtrx[9] = ~mtrx[0];
	mtrx[9].setName("mtrx[9] = transposed(mtrx[0])");
	cout << mtrx[9]; fout << mtrx[9];

	fout.close();

} // end of main()
```

2A.4 실행 결과 (화면 및 파일 출력 결과 예시)



```
mtrx[0] =
┌  1.10   2.20   3.30   4.40   5.50┐
│  3.30   4.40   5.50   6.60   7.70│
└  8.80   9.90   1.10   2.20   3.30┘

mtrx[1] =
┌  1.10   0.00   0.00   0.00   0.00┐
│  0.00   1.10   0.00   0.00   0.00│
└  0.00   0.00   1.10   0.00   0.00┘

mtrx[2] =
┌  1.10   2.20   3.30┐
│  4.40   5.50   6.60│
│  7.70   8.80   9.90│
│  5.50   1.10   7.70│
└  6.60   4.40   3.30┘

mtrx[3] = mtrx[0] + 2.0 =
┌  3.10   4.20   5.30   6.40   7.50┐
│  5.30   6.40   7.50   8.60   9.70│
└ 10.80  11.90   3.10   4.20   5.30┘

mtrx[4] = mtrx[0] - 2.0 =
┌ -0.90   0.20   1.30   2.40   3.50┐
│  1.30   2.40   3.50   4.60   5.70│
└  6.80   7.90  -0.90   0.20   1.30┘

mtrx[5] = mtrx[0] * 2.0 =
┌  2.20   4.40   6.60   8.80  11.00┐
│  6.60   8.80  11.00  13.20  15.40│
└ 17.60  19.80   2.20   4.40   6.60┘

mtrx[6] = mtrx[0] + mtrx[1] =
┌  2.20   2.20   3.30   4.40   5.50┐
│  3.30   5.50   5.50   6.60   7.70│
└  8.80   9.90   2.20   2.20   3.30┘

mtrx[7] = mtrx[0] - mtrx[1] =
┌  0.00   2.20   3.30   4.40   5.50┐
│  3.30   3.30   5.50   6.60   7.70│
└  8.80   9.90   0.00   2.20   3.30┘

mtrx[8] = mtrx[0] * mtrx[2] =
┌ 96.80  72.60 102.85┐
│152.46 121.00 170.61│
└ 95.59 100.43 133.10┘

mtrx[9] = transposed(mtrx[0]) =
┌  1.10   3.30   8.80┐
│  2.20   4.40   9.90│
│  3.30   5.50   1.10│
│  4.40   6.60   2.20│
└  5.50   7.70   3.30┘
```

2A.5 결과물 제출

- 바탕화면에 Exam2 폴더를 생성 후 Exam2A_학번_이름 프로젝트를 생성
- 압축 파일 내에 포함사항 : 작성한 프로젝트, 실행결과 Capture(채점 시 정확한 실행 유무를 확인하기 위함)
- 실행 화면 캡쳐파일은 각 시험 섹션별로 프로젝트 폴더 내에 저장 후 시험 섹션별 폴더별로 압축
- 제출시 .vs 폴더는 삭제 후 문제별 폴더를 압축하여 제출