CIS 3309 Section 2 Lab: Matrix Ops 2 Object Problem

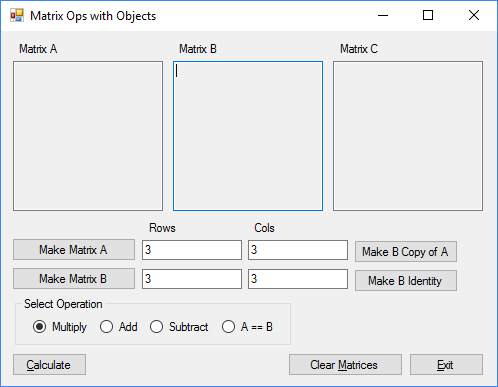
In this lab, you will convert your last lab project into an object-oriented solution. The first few steps are:

1. Open your Matrix solution in VS
2. Open another VS instance and create a solution named “Martix2”
3. Stretch the form in Matrix 2 to approximately the size of Matrix
4. Select all the items on the form in Matrix and copy and paste into Matrix2
5. Add the additional controls to the form as shown in the form below
6. Add a Matrix class to the Matrix2 solution and code as described below
7. Add the necessary code to the form class to run the app

The form below has two extra controls:

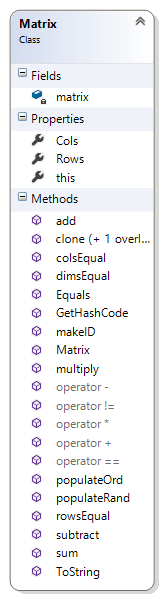
1. The “Make B Copy ofA” button
2. The “A == B” radio button

The form code is initially empty because copy-and-paste only copies controls and not the associated code. This is good because the code will be different than the last project.



After adding the Matrix class, you need to code it. Look at the Chap13 sample project in Course Documents to see some of the method and there implementation. The class diagrams below show the frmMain and Matrix classes from my solution. The executable included with this document shows how the program should run.





In the Matrix class:

matrix is a 2D array of double

Cols is a read-only property that returns the number of columns in the matrix

Rows is a read-only property that returns the number of rows in the matrix

this is an indexer for a matrix. Note that for a matrix you need this[int i, int j] for both dimensions.

Matrix is a constructor the instantiates the matrix of desired size.

The add, subtract and multiply methods perform their respective op on the A and B matrices, returning a C result matrix. Note that they are called as C = A.add(B);

The colsEqual, rowsEqual and dimsEqual methods compare the number of columns, rows and dimensions between two matrices for equality. Note bool b = A.dimsEqual(B); returns true if the matrices have the same dimensions.

Equals is an override method that compares two matrices for equality. They are equal if both are null, or both have the same numeric values in the same element positions.

GetHashCode is an override method that is implemented as:

// Overrides the GetHashCode method

public override int GetHashCode()

{

// Use sum for hash code

return sum().GetHashCode(); // Note that double overrides this too

}

sum calculates the sum of all elements in the matrix.

makeId makes an n x n identity matrix.

clone returns a copy of this matrix.

copy accepts a matrix and copied its elements to this.

populateRand populates this matrix with random doubles.

populateOrd populates this matrix with sequential doubles from 1.0 in the first element to d in the last, as traversed by a nested loop.

ToString

The operator methods overload the operators to work with matrices.

Look at the Chap13 solution and the included exe to see how these work.