

Lab 8.4.1 Operators: arithmetic operators

Objectives

Familiarize the student with:

- using operators in real programs;
- using exceptions in real programs.

Scenario

Write a class which holds matrices of different sizes and create methods to:

- add a value to a matrix (add a value to each individual cell of the matrix);
- add two matrices, and if their sizes are different, throw an exception;
- subtract a value from a matrix (subtract a value from each individual cell of the matrix);
- subtract two matrices, and if their sizes are different, throw an exception;
- multiply a value by a matrix (multiply a value by each individual cell of the matrix);
- multiply two matrices, and if the row count of the first matrix is different to the column count of the second matrix, throw an exception.

Add an overloaded operator to each of the methods above. In the main function, perform a few operations with operators only. Create two matrices, one with 1 in all cells, and another with 3 in all cells. Perform all operations, and print all indirect results. After testing this scenario, add a third matrix with a different size to the first one. Try to make your code throw all possible exceptions.

Example input

```
Matrix A:
1 1
1 1
Matrix B:
3 3
3 3
```

Example output

```
Matrix A + 1:
2 2
2 2
Matrix A * 2:
4 4
4 4
Matrix A - 1:
3 3
3 3
Matrix A + Matrix B:
6 6
6 6
Matrix A - Matrix B:
0 0
0 0
Matrix A * Matrix B:
18 18
18 18
```

Example input

```
Matrix A:
1 1
1 1
Matrix C:
3 3 3
3 3 3
```

Example output

```
Matrix A + Matrix C:  
One of matrix sizes is not valid  
Matrix A - Matrix C:  
One of matrix sizes is not valid  
Matrix A * Matrix C:  
One of matrix sizes is not valid
```