

## Zadanie 9, Języki Programowania

### 1. Zadanie

Należy stworzyć klasę umożliwiającą pracę na macierzach.

### 2. Funkcjonalność

Klasa powinna umożliwiać zapisywanie i wczytywanie macierzy do/z pliku tekstowego, dodawanie i mnożenie macierzy, oraz zamienianie miejscami kolumn, jak również liczenie wyznaczników.

### 3. Wykonanie

Klasa `Matrix` powinna mieć następujące pola prywatne:

```
int n,m; //wymiary macierzy
```

```
double** array; // Dwuwymiarowa tablica przechowująca wszystkie wartości
```

Konstruktory:

- Konstruktor główny (domyślne wymiary  $m = n = 3$ )
- Konstruktor kopiujący (dlaczego jest niezbędny?)
- Destruktor

Oraz funkcje składowe (metody):

- `Change(int x, int y, double val)` – wpisującej wartość `val` do komórki `[x][y]` macierzy
- `Print` - wypisuje macierz na ekran
- `Random()` - wypełniająca losowo liczbami 1-9 całą macierz (**`#include <stdlib.h>`, `rand()`, `#include <time.h>`, `srand(time(NULL))` – raz na początku programu, ustawienie ziarna**)
- `Save(char*)` - zapisuje macierz do pliku (w mainie zapisać do pliku podanego jako pierwszy parametr wywołania programu)
- `Open(char*, int N, int M)` - wczytuje macierz  $N \times M$  z pliku (w mainie wczytać z pliku podanego jako drugi parametr wywołania programu)
- Przeciążone operatory: `+`, `*` (dodawania, mnożenia) Jeśli działania nie można wykonać, należy zwrócić macierz stojącą po lewej stronie operatora.
- Operator `()` - przyjmujący dwa argumenty i zwracający wartość przechowywaną w odpowiedniej komórce macierzy
- `Exchange(int i, int j)` Metoda umożliwiająca zamianę kolumn ( $i$  z  $j$ ),

### Dodatkowe:

- Metoda zwracająca wyznacznik dla macierzy  $2 \times 2$  lub  $3 \times 3$  (dla innych zwraca 0),
- Metoda zwracająca wyznacznik dla macierzy o dowolnych wymiarach

*Punkty będą przydzielane jedynie za kod przetestowany w funkcji main!*

### Dynamiczne tworzenie dwuwymiarowych tablic w C++: (tablica liczb całkowitych $5 \times 10$ )

```
int** tab = new int* [5];
```

```
for (int i = 0; i < 5; i++)
```

```
    tab[i] = new int [10];
```

W ten sposób stworzono tablicę dwuwymiarową którą statycznie zadeklarowalibyśmy jako:

```
int tab[5][10];
```

Na końcu należy również **zwolnić pamięć** dla dynamicznie zadeklarowanej tablicy:

```
for (int x = 0; x < 5; x++)
```

```
    delete [] tab[x];
```

```
delete [] tab;
```

**Przykładowy output:**

Macierz A:

1 0 0

0 0 4

1 2 2

Macierz B:

0 0 7

0 0 0

5 0 0

Operator+:

1 0 7

0 0 4

6 2 2

Operator\*:

0 0 7

20 0 0

10 0 7

Komorka (2,1) z operatora():4

Zamiana kolumn 0 z 1 dla A

0 1 0

0 0 4

2 1 2

Macierz A:

1 0 0 0

0 0 4 0

1 2 2 0

Macierz B:

0 0 7

0 0 0

5 0 0

0 0 0

Operator+:

1 0 0 0

0 0 4 0

1 2 2 0

Operator\*:

0 0 7 0

20 0 0 0

10 0 7 0

Komorka (2,1) z operatora():4

Zamiana kolumn 0 z 1 dla A

0 1 0 0

0 0 4 0

2 1 2 0