

Zadanie 9, Języki Programowania, Czwartek 16.12.2014, zadanie domowe

1. Zadanie

Należy stworzyć klasę umożliwiającą wykonywanie operacji matematycznych na macierzach.

2. Funkcjonalność

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

3. Wykonanie

Klasa `Matrix` powinna mieć następujące pola prywatne:

```
int n,m; //wymiary macierzy, n - liczba wierszy, m - liczba kolumn
double** array; // Dwuwymiarowa tablica przechowująca wszystkie wartości
```

Konstruktory:

- Konstruktor z parametrami (domyślne wymiary $n = m = 2$)
- Konstruktor kopiujący i operator= (dla bezpieczeństwa)
- 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 rzeczywistymi z przedziału 1-9 całą macierz (użycie funkcji `rand()`, potrzebne biblioteki: `#include <stdlib.h>`, `#include <time.h>`, oraz ustawienie ziarna `srand(time(NULL))` – powinno być znane z C, jak nie to wyszukać w Google
- `Save(char* outputFile)` - zapisuje macierz do pliku (w main'ie zapisać do pliku podanego jako pierwszy parametr wywołania programu)
- `static Matrix* Read(char inputFile)` - wczytuje macierz NxM z pliku (w main'ie wczytać z pliku podanego jako drugi parametr wywołania programu), metoda statyczna – zwraca obiekt alokując mu pamięć wewnątrz metody
- `operator+` – działający tylko wtedy, kiedy dodawanie na macierzach jest wykonalne (dwie takie same macierze), w przeciwnym razie wypisać na ekranie komentarz o błędnym dodawaniu i zwrócić jedną z macierzy bez zmian
- `Operator()` - przyjmujący dwa argumenty i zwracający wartość przechowywaną w odpowiedniej komórce macierzy
- `Change(int i, int j)` Metoda umożliwiająca zamianę kolumn (i z j),

Dodatkowo (jeśli komuś się uda zrobić resztę):

- `operator*` - mnożenie dwóch macierzy (analogicznie jak dodawanie może być tylko wykonane gdy mnożone macierze spełniają warunek iloczynu macierzowego)

Uwaga! Dynamiczne tworzenie dwuwymiarowych tablic w C++: (tablica liczb całkowitych 5 x 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 i = 0; i < 5; i++)
    delete [] tab[i];
delete [] tab;
```