

Zadanie 9, Języki Programowania, Wtorek, 20.12.2015

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 macierzy, oraz zamienianie miejscami kolumn.

3. Wykonanie

Klasa `Matrix` powinna mieć następujące pola prywatne:

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

Konstruktory:

- Konstruktor z parametrami (domyślne wymiary $n = m = 2$)
- Konstruktor kopiujący i operator= (dla bezpieczeństwa)
- Destraktor

Oraz funkcje składowe (metody):

- `Change(int x, int y, double val)` – wpisującej wartość `val` do komórki `[x][y]` macierzy
- `Print()` – wypisuje macierz na ekran **1 p.** (do tego miejsca)
- `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 **1 p.**
- `Save(char* outputFile)` - zapisuje macierz do pliku (w main'ie zapisać do pliku podanego jako pierwszy parametr wywołania programu `argv[1]`) **0.5 p.**
- `static Matrix* Read(char* inputFile)` - wczytuje macierz NxM z pliku (w main'ie wczytać z pliku podanego jako drugi parametr wywołania programu `argv[2]`), metoda statyczna – zwraca obiekt alokując mu pamięć wewnątrz metody **1 p.**
- 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 **1 p.**
- `Change(int i, int j)` Metoda umożliwiająca zamianę kolumn (i z j), metoda nazywa się tak samo jak wcześniejsza **0.5 p**

Uwaga! Dynamiczne tworzenie dwuwymiarowych tablic w C++: (tutaj 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;
```

Obsługa plików tekstowych – przykłady

Przykład zapisywania danych do pliku w języku C++:

```
#include <fstream> //nagłówek zawierający funkcje C++ do operacji na plikach
```

```
ofstream ofile;  
ofile.open("file.txt"); //"file.txt" jest tutaj typu char* i oznacza nazwę pliku  
wyjściowego  
ofile<<"tekst "<<zmienna<<endl; //przekierowanie danych do pliku jest analogiczne  
jak w przypadku wypisania na ekran za pomocą cout  
ofile.close();
```

Przykład wczytywania (w tym przypadku liczb) z pliku:

```
#include <fstream>
```

```
ifstream ifile;  
int val;  
ifile.open("file.txt");  
while(ifile>>val)  
{  
    cout<<"val: "<<val<<endl;  
}  
ifile.close();
```

Komentarz: `ifile>>val` – ten sposób pozwala na przypisanie do zmiennej `val` kolejnych liczb znajdujących się w pliku oddzielonych od siebie spacjami, tabulacjami lub enterami.