

LAB 3

Zdefiniować klasę **macierz** do reprezentowania statycznych tablic kwadratowych o maksymalnym rozmiarze **NxN**

```
class macierz {
    static const int N=10;    //więcej o polach static będzie na w4
    double    t[N][N]={0};
    int        n;              //bieżący wymiar macierzy

public:
    // zdefiniuj tylko niezbędne elementy dla klasy
    // (wykorzystane w funkcji main)
};
```

Należy utworzyć projekt składający się z 3 plików: **macierz.hpp**, **macierz.cpp**, **progrmain.cpp**

Plik **macierz.hpp** powinien zawierać tylko definicję klasy (ewentualnie nagłówki poza klasowych funkcji), plik **macierz.cpp** powinien zawierać implementację metod, funkcji i operatorów, funkcji zaprzyjaźnionych, itp.

Plik **main.cpp** otrzymujesz *gotowy*. Podziel samodzielnie funkcję **main()** na mniejsze fragmenty i sukcesywnie rozwijaj kod.

Uwagi do etapów zadania

ETAP 1

Zdefiniuj odpowiednie składowe lub funkcje zaprzyjaźnione dla klasy **macierz**.

Przetestuj klasę **macierz** w funkcji **main**.

Normę macierzy definiujemy jako: $\|A\| = \sqrt{\sum_i \sum_j a_{i,j}^2}$

ETAP 2

Zastosuj klasę **macierz** do znajdowania macierzy **pseudoodwrotnej**.

W tym celu klasę **macierz** rozszerz o metodę wyznaczania **pseudoodwrotności** (*wykorzystaj już zdefiniowane elementy klasy*)

Algorytm

$A1, A2, I \in R^{n \times n}$

I – macierz jednostkowa

$$\alpha = \frac{1}{\|A\|^2}$$

$$A1 = \alpha \cdot A^T$$

dla $i = 1:ITER$

$$\begin{cases} A2 = A1 + A1 \cdot (I - A \cdot A1) \\ \text{gd}y \|A2 - A1\| < EPS, & break \\ A1 = A2 \end{cases}$$

return $A2$