

=====  
=====

Zadanie ze struktur i alokacji.

Celem zadania jest napisanie struktury Tab2D mającej funkcjonalność tablicy dwuwymiarowej.

Potrzebne są następujące funkcjonalności występujące w kodzie:

Tab2D ta;

init(ta, 5, 5, 0.1); // inicjalizuje tablice 5 x 5 wartościami 0.1

set(ta, 0.2); // ustawi wartości w tablicy

print(ta); // wypisuje tablicę w formie jak poniżej

pyramid(ta, 0); wypełnia tablicę wartościami tak jak poniżej, to ma działać dla każdego rozmiaru tablicy

0 0 0 0 0

0 1 1 1 0

0 1 2 1 0

0 1 1 1 0

0 0 0 0 0

Tab2D tb;

init(tb, 5, 5, 1.7);

Tab2D tc = add( ta, tb ); // wytwarza nowa tablice, której elementy są sumą

Tab2D td = mul( ta, tb ); // mnoży macierz

del(ta); // kasuje pamiec

Punktacja:

- struktura Tab2D i funkcja init: 2pkt

- print: 1pkt

- set: 1pkt

- pyramid: 1pkt

- add: 2pkt

- mul: 2pkt

- del: 1 pkt

Działanie każdej z tych funkcji musi być pokazane na dwóch przykładach o różnych rozmiarach tablic.

Funkcjonalność nie może być zaimplementowana z użyciem zaawansowanych struktur danych. Konieczne jest dokonanie alokacji i dealokacji pamięci.

=====  
=====

Zadanie z klas.

Należy powyższą funkcjonalność zaimplementować z użyciem klas. Do przechowania danych mogą być użyte bardziej zaawansowane struktury danych.

```
Tab2D ta(5, 5, 0.1); // tworzy tablicę
ta.set(0.3); // ustawia wartości
ta.print(); // ma wypisywać tablice jak wyżej.
ta.pyramid(); // zobacz zadanie wyżej
Tab2D tb = ta.add(ta); // tworzy nową tablicę o rozmiarach takich jak ta, której
elementu są suma odpowiednich elementów z tablic źródłowych
Tab2D tc = ta.mul(tb); // mnoży macierze
```

Działanie każdej z tych metod musi być pokazane na dwóch przykładach o różnych rozmiarach tablic.

Punktacja:

- klasa i konstruktor: 2pkt
- print: pkt
- set: 1pkt
- pyramid: 1pkt
  
- add: 2pkt
- mul: 2pkt
- destruktor: 1 pkt