Programare paralela si distribuita

Laboratorul 2

**Analiza cerintei:**

Enuntul problemei:

• A) Generalizati programul de la laboratorul 1 (pct. A) prin inlocuirea operatiei de adunare cu un operator binar asociativ. Generalizati si tipul operanzilor! Implementati un program pentru adunare matrice de numere complexe. Testati programul generalizat pentru diferiti operatori si pentru numar diferit de thread-uri si analizati timpul de executie.

• B) Transcrieti programul in C++11. Recomandari: Folositi expresii lambda pentru definirea operatorilor!

La cazurile de testare see vor folosi matrici de 1000x1000 generate aleator si ca numar de threaduri: 2,4,6,8.

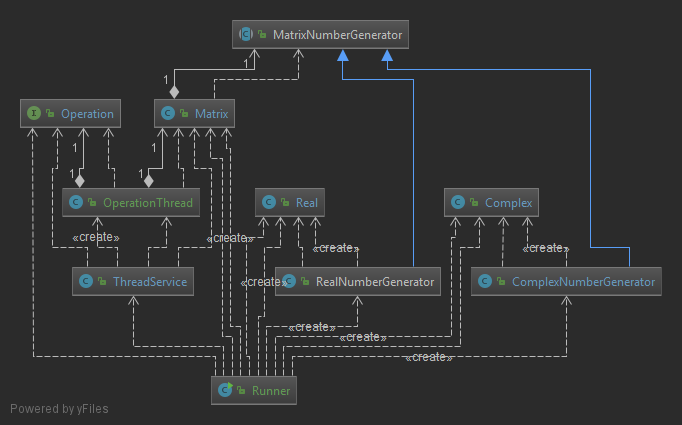
**Important**: folositi o incarcare echilibrata pentru fiecare thread !

**Rezolvarea problemei:**

Se creeaza un o clasa Matrix care va primi un tip de date Generic. Se creeaza 2 clase : Complex si Real care vor fi folosite de catre matrice. Fiecare tip de data va avea un generator de numere random pentru initializarea matricei. Service-ul de threaduri va primi cele 3 matrici, 2 initializate random si o a treia pentru rezultat, numarul de threaduri si o operatie . Aceasta operatie va fi definite sub forma unei functii lambda si apoi pasata service-ului . Pentru fiecare tip de data se va calcula inmultirea si o operatie binara asociativ intre cele 2 matrici.

**Graficele ce indica comportamentul operatiilor:**

**Structura programului - Java**:

****