Documentație Laborator 2

Muntea Andrei – Marius (235)

I Cerință

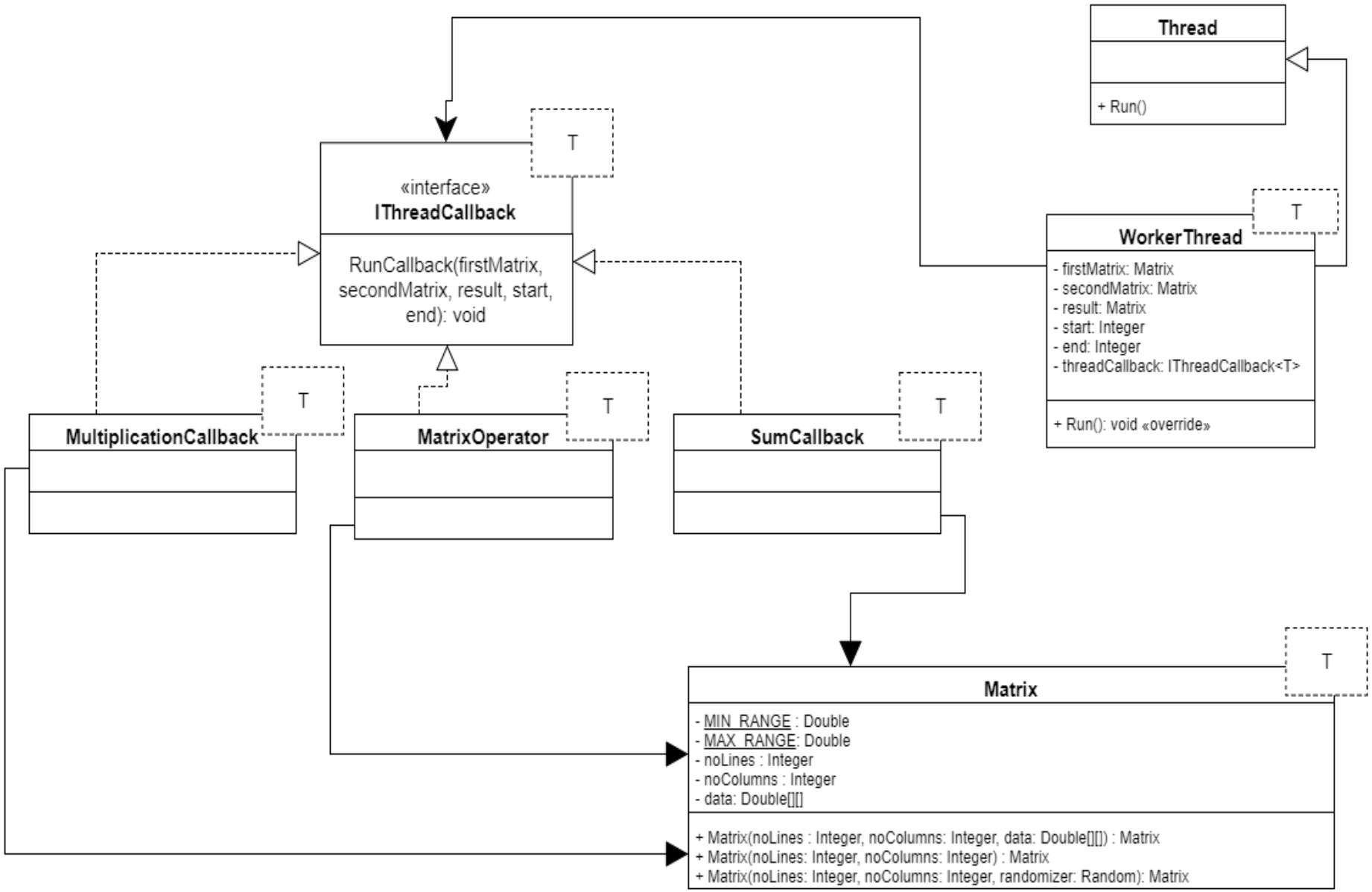
Generalizați programul de la laboratorul 1 prin înlocuirea operației de adunare cu un operator. Implementați un program pentru adunare matrice de numere complexe. Testați programul corespunzator adunării matricilor pentru diferiți operatori și pentru numar diferit de threaduri și analizați timpul de execuție. Transcrieți programul in C++11.

II Descrierea soluției

Vom implementa o clasă *Matrix<T>* care va încapsula numărul de linii, numărul de coloane respectiv matricea de date. Pentru a folosi o distribuție cât mai balansată a datelor pe care operează fiecare thread, vom implementa operația de adunare respectiv de înmulțire între matrice specificând indicele liniei de început respectiv indicele liniei de sfârșit. Fiecare thread va procesa *nrLinii / nrThreaduri (+1)* linii. Pentru genericitate, fiecare thread va primi un callback cu parametrii*: Matrix<T> firstMatrix, Matrix<T> secondMatrix, Matrix<T> result, Integer start, Integer end* pe care il va apela.

III Rezultate testare

	Java (ms)	C++ (ms)	Numărul de threaduri
Înmulțire matrice de numere complexe 1000 x 1000	155973.12	18874.25	1
	77390.41	9852.13	2
	41875.79	5987.59	4
	28493.03	5929.96	8
	29945.72	5958.94	64
Înmulțire matrice de numere reale 1000 x 1000	72382.06	15245.09	1
	3739.32	7595.52	2
	23521.13	3872.99	4
	19649.39	3680.56	8
	19493.54	3825.46	64
Operator de compunere pe matrice de numere complexe 1000 x 1000	1956.95	66.12	1
	185.92	32.96	2
	190.01	16.75	4
	202.23	13.50	8
	293.83	14.47	64
Operator de compunere pe matrice de numere reale 1000 x 1000	1385.55	13.63	1
	46.00	6.71	2
	43.07	4.38	4
	111.04	4.30	8
	179.63	6.59	64


Thread
+ Run()
T
«interface»
IThreadCallback
RunCallback(firstMatrix, secondMatrix, result, start, end): void
T
WorkerThread
- firstMatrix: Matrix
- secondMatrix: Matrix
- result: Matrix
- start: Integer
- end: Integer
- threadCallback: IThreadCallback<T>
+ Run(): void «override»
T
MultiplicationCallback
T
MatrixOperator
T
SumCallback
T
Matrix
- MIN_RANGE : Double
- MAX_RANGE: Double
- noLines : Integer
- noColumns : Integer
- data: Double[][]
+ Matrix(noLines : Integer, noColumns: Integer, data: Double[][]) : Matrix
+ Matrix(noLines: Integer, noColumns: Integer) : Matrix
+ Matrix(noLines: Integer, noColumns: Integer, randomizer: Random): Matrix