**Laborator 2 PPD**

**- documentație-**

Cerință: Implementați în Java și C++ (11 sau >11)

A) înmulțirea a două ‘numere mari’

1)secvențial

2)paralel - folosind multithreading

**Algoritm secvențial:** Pentru a implementa înmulțierea a două numere mari în varianta secvențială am folosit algoritmul lui Karatsuba. Acesta este un algoritm recursiv de forma divide et impera. Astfel, cele două numere mari se împart în jumătăți => numărul 1 e de forma [high1,low1], iar numărul 2 [high2,low2]. Urmează apoi 3 apeluri recursive care au ca parametrii (low1,high1), (low1+high1,low2+high2) respectiv (high1,high2). Condiția de oprire este ca unul din numere să fie cifră, returnându-se rezultatul înmulțirii cifrei cu celălalt număr.

**Algoritm paralel:** În cazul algoritmului paralel am plecat de la ideea obișnuită de înmulțire, înmulțirea fiecărei cifre din al doilea număr cu primul număr și apoi adunarea rezultatelor.

Înmulțirea cifrelor celui de-al doilea număr cu primul număr se face în paralel, adică un thread are de înmulțit un număr de cifre din al doilea număr cu primul număr și să salveze rezultatul.

Apoi se parcurge vectorul rezultat și se adună în paralel numerele, două câte două, primul număr este adunat cu cel de-al doilea, rezultatul cu cel de-al treilea ș.a.m.d.

Diagrama de clase C++:

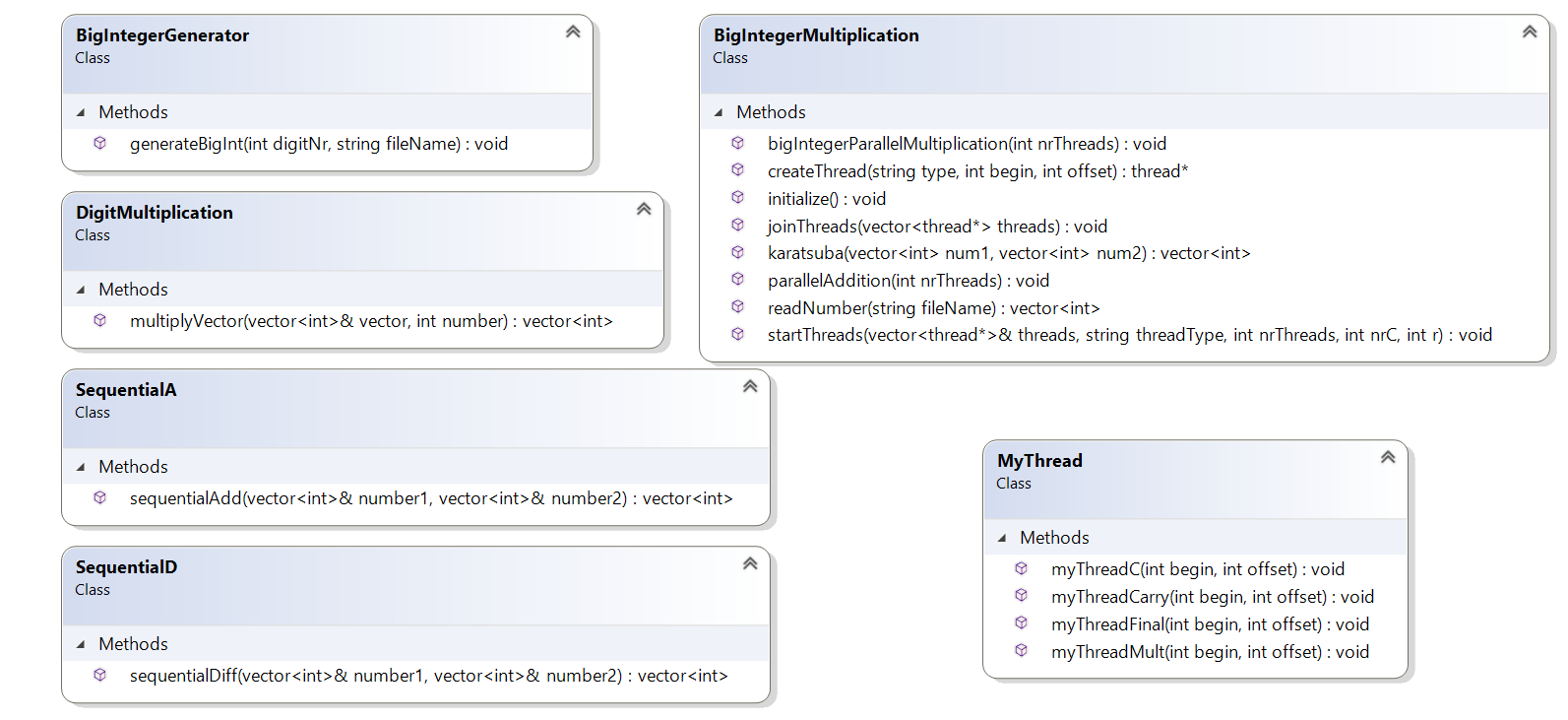
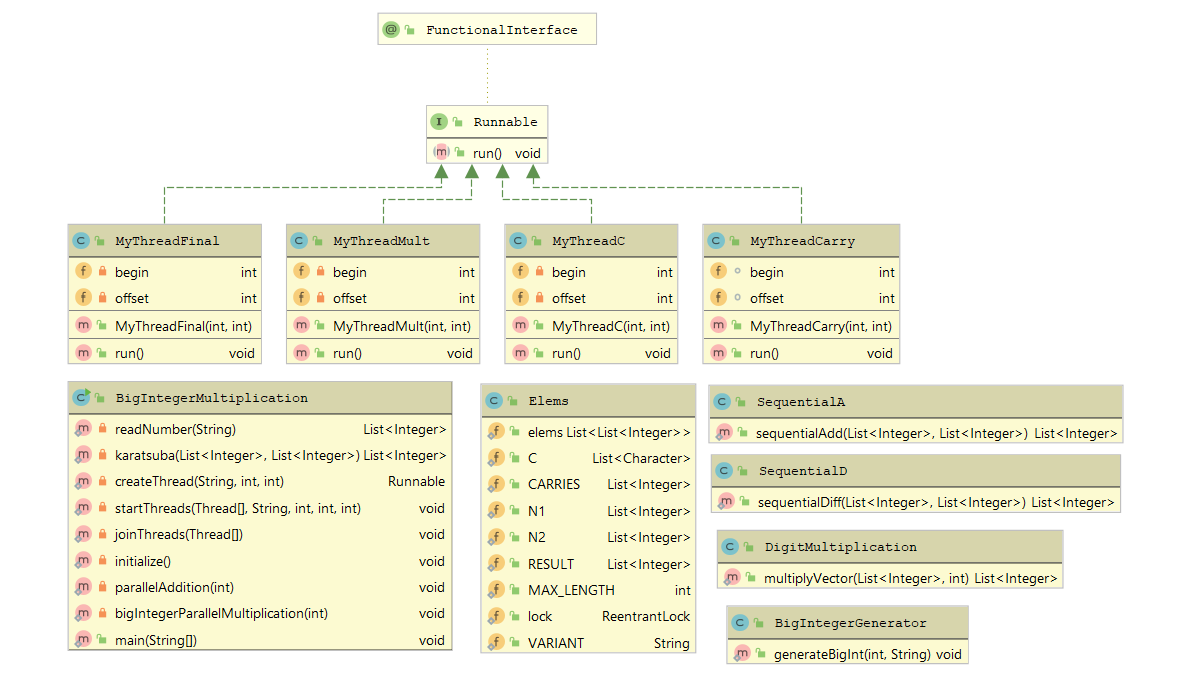


Diagrama de clase Java:



Cazuri de testare:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. cifre | Limbaj | Nr.thread-uri | Secvential | Paralel |
| 4 | JAVA | 2 | 0.004 | 0.011 |
| 10 | JAVA | 2 | 0.004 | 0.02 |
| 10 | JAVA | 4 | ------- | 0.032 |
| 100 | JAVA | 2 | 0.024 | 0.168 |
| 100 | JAVA | 4 | ------- | 0.272 |
| 500 | JAVA | 4 | 1.32 | 1.278 |
| 500 | JAVA | 8 | ------- | 2.207 |
| 1000 | JAVA | 4 | 3.17 |  |
| 1000 | JAVA | 8 | ------- |  |
| 10000 | JAVA | 4 | 9.907 |  |
| 10000 | JAVA | 8 | ------- |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4 | C++ | 2 | 3.0329e-05 |  |
| 10 | C++ | 2 | 0.000135197 |  |
| 10 | C++ | 4 | ------- |  |
| 100 | C++ | 2 | 0.00975011 |  |
| 100 | C++ | 4 | ------- |  |
| 500 | C++ | 4 | 0.0553248 |  |
| 500 | C++ | 8 | ------- |  |
| 1000 | C++ | 4 | 0.169888 |  |
| 1000 | C++ | 8 | ------- |  |
| 10000 | C++ | 4 | 7.32791 |  |
| 10000 | C++ | 8 | ------- |  |