Documentatie Laborator 2

Ghimpu Lucian Eduard

**Cerinta pe scurt:**

Se cere o generalizare a laboaratorului 1, atat operatia cat si operatorul trebuie sa fie generice. Trebuie observant care is diferentele atunci cand se schimba parametrii dar si intre limbaje (Java, c++).

Detalii de implementare:

Am folosit o clasa wrapper deasura unui array bidimensional **Matrix<T>,** clasa ofera metode auxiliare pe langa accesarea datelor (ex: clear(), toString() etc…) pentru a facilita implementare.

Pentru a facilita implementare am creat diferete clase auxiliare: **Utils, Constants, FileManager.** Aceste difera si de limbaj.

In implementarea din Java am mostenit din **Thread** si am pasat ca parametrii, matricile, indicia si operatorul (**o interfata functionala**). In **C++** am instantiat thread-uri la care am pasat direct metoda si parametrii, iar operatorul a fost dat folosind **std::function**.

In ambele implementari m-am folosit de o clasa **“Manager”** pentru a combina restul claselor si pentru a gestiona executia (crearea de threaduri, impartirea de date, etc…).

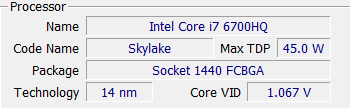
OBS: majoritatea claselor sunt facute cu Generics/Template astel in fac ofera posibilitatea de a lucra cu diferite tipuri de date.

De asemenea numarul de coloane, randuri si threaduri sunt citite de la tastatura.

**Un flow al programului:**

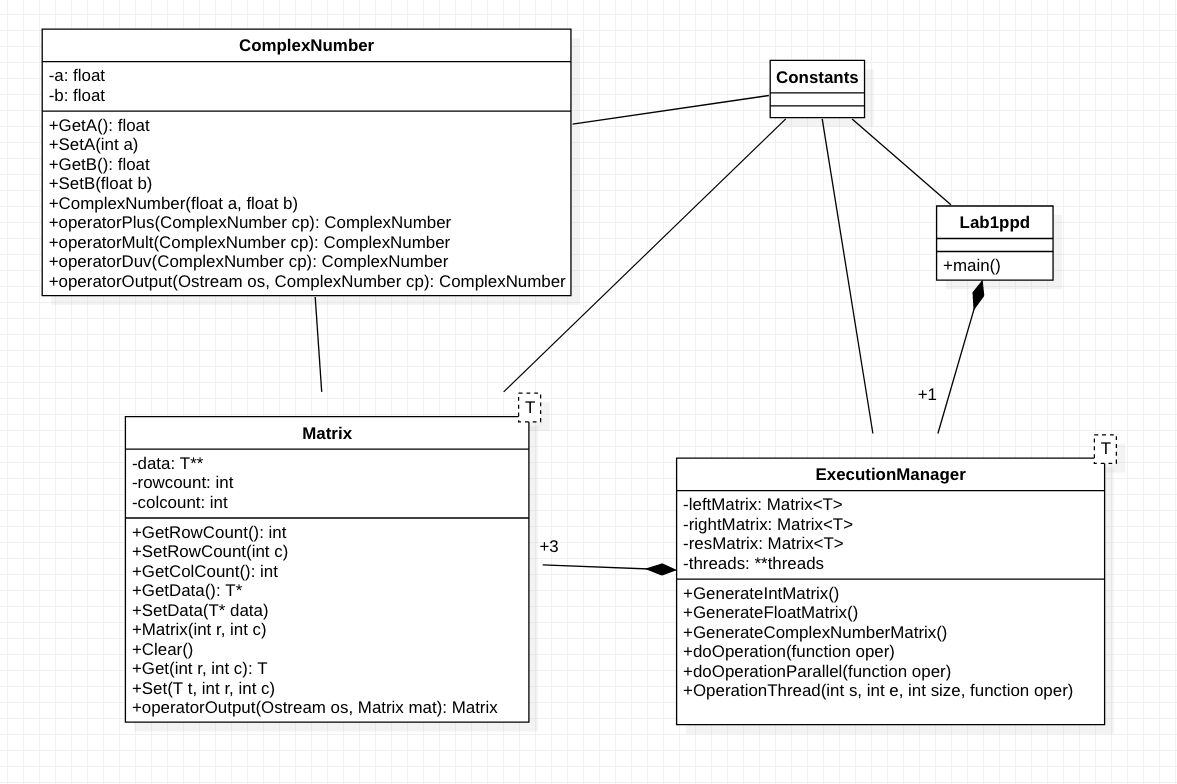
1. Se creeaza fisierul pentru I/O
2. Se citesc de la tastatura parametrii
3. Se genereaza cele 2 matrici
4. Se executa prin intermediul managerului executia operatiei secventiale
5. Se printeaza rezultatele in fisier
6. Se creeaza threadurile si se executa paralel operatia
7. Se printeaza rezultatele in fisier

**Detalii masina:**

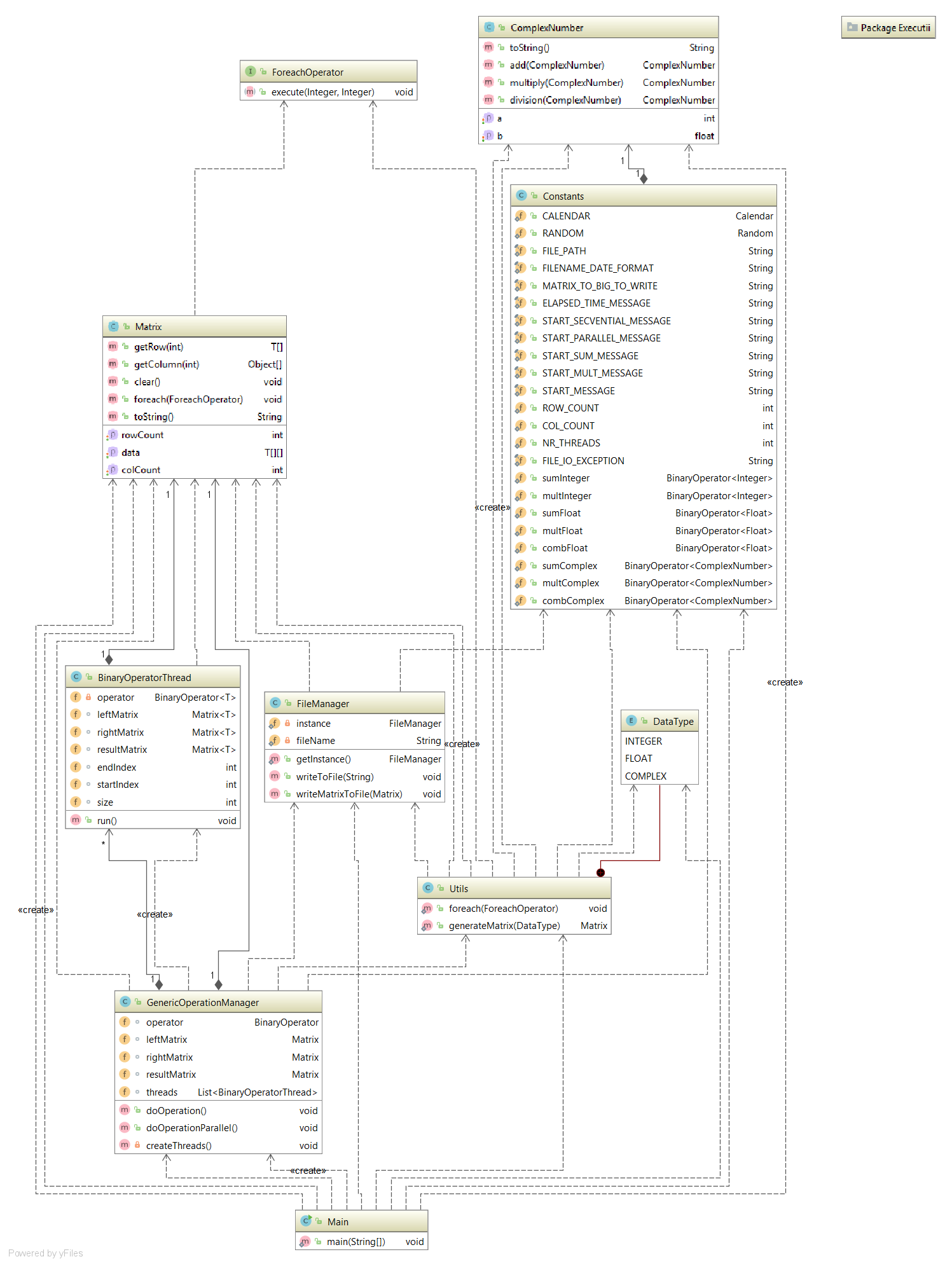
****

**Diagramele de clase:**

1. **C++**

****

1. Java



Limbaj**:** **C++**

Operand: **Numar Real**

Operator: **a\*b**

|  |  |  |
| --- | --- | --- |
| Dimensiune Matrice | Nr. Thread-uri | Timp (ms) |
| 10x10 | 1 | 1 |
| 10x10 | 2 | 2 |
| 10x10 | 4 | 4 |
| 10x10 | 6 | 7 |
| 10x10 | 8 | 8 |
| 10x10 | 32 | 20 |
| 10x10 | 128 | 77 |
| 10x10 | 256 | 158 |

|  |  |  |
| --- | --- | --- |
| Dimensiune Matrice | Nr. Thread-uri | Timp (ms) |
| 500X500 | 1 | 122 |
| 500X500 | 2 | 70 |
| 500X500 | 4 | 50 |
| 500X500 | 6 | 43 |
| 500X500 | 8 | 36 |
| 500X500 | 32 | 52 |
| 500X500 | 128 | 106 |
| 500X500 | 256 | 199 |

|  |  |  |
| --- | --- | --- |
| Dimensiune Matrice | Nr. Thread-uri | Timp (ms) |
| 1000x1000 | 1 | 555 |
| 1000x1000 | 2 | 294 |
| 1000x1000 | 4 | 184 |
| 1000x1000 | 6 | 140 |
| 1000x1000 | 8 | 129 |
| 1000x1000 | 32 | 153 |
| 1000x1000 | 128 | 235 |
| 1000x1000 | 256 | 309 |

Limbaj**:** **C++**

Operand: **Numar Real**

Operator: **a ¤ b = 1/ (1/ a + 1/b)**

|  |  |  |
| --- | --- | --- |
| Dimensiune Matrice | Nr. Thread-uri | Timp (ms) |
| 10x10 | 1 | 1 |
| 10x10 | 2 | 3 |
| 10x10 | 4 | 5 |
| 10x10 | 6 | 8 |
| 10x10 | 8 | 11 |
| 10x10 | 32 | 23 |
| 10x10 | 128 | 101 |
| 10x10 | 256 | 177 |

|  |  |  |
| --- | --- | --- |
| Dimensiune Matrice | Nr. Thread-uri | Timp (ms) |
| 500X500 | 1 | 136 |
| 500X500 | 2 | 68 |
| 500X500 | 4 | 49 |
| 500X500 | 6 | 40 |
| 500X500 | 8 | 48 |
| 500X500 | 32 | 70 |
| 500X500 | 128 | 106 |
| 500X500 | 256 | 201 |

|  |  |  |
| --- | --- | --- |
| Dimensiune Matrice | Nr. Thread-uri | Timp (ms) |
| 1000x1000 | 1 | 562 |
| 1000x1000 | 2 | 311 |
| 1000x1000 | 4 | 173 |
| 1000x1000 | 6 | 151 |
| 1000x1000 | 8 | 133 |
| 1000x1000 | 32 | 143 |
| 1000x1000 | 128 | 234 |
| 1000x1000 | 256 | 304 |

Limbaj**:** **C++**

Operand: **Numar Complex**

Operator: **a ¤ b = 1/ (1/ a + 1/b)**

|  |  |  |
| --- | --- | --- |
| Dimensiune Matrice | Nr. Thread-uri | Timp (ms) |
| 10x10 | 1 | 1 |
| 10x10 | 2 | 2 |
| 10x10 | 4 | 5 |
| 10x10 | 6 | 6 |
| 10x10 | 8 | 8 |
| 10x10 | 32 | 23 |
| 10x10 | 128 | 112 |
| 10x10 | 256 | 164 |

|  |  |  |
| --- | --- | --- |
| Dimensiune Matrice | Nr. Thread-uri | Timp (ms) |
| 500X500 | 1 | 187 |
| 500X500 | 2 | 107 |
| 500X500 | 4 | 71 |
| 500X500 | 6 | 55 |
| 500X500 | 8 | 47 |
| 500X500 | 32 | 68 |
| 500X500 | 128 | 134 |
| 500X500 | 256 | 202 |

|  |  |  |
| --- | --- | --- |
| Dimensiune Matrice | Nr. Thread-uri | Timp (ms) |
| 1000x1000 | 1 | 787 |
| 1000x1000 | 2 | 425 |
| 1000x1000 | 4 | 252 |
| 1000x1000 | 6 | 201 |
| 1000x1000 | 8 | 194 |
| 1000x1000 | 32 | 188 |
| 1000x1000 | 128 | 263 |
| 1000x1000 | 256 | 334 |

Limbaj**:** **JAVA**

Operand: **Numar real**

Operator: **a \* b**

|  |  |  |
| --- | --- | --- |
| Dimensiune Matrice | Nr. Thread-uri | Timp (ms) |
| 10x10 | 1 | 1 |
| 10x10 | 2 | 42 |
| 10x10 | 4 | 41 |
| 10x10 | 6 | 61 |
| 10x10 | 8 | 71 |
| 10x10 | 32 | 23 |
| 10x10 | 128 | 73 |
| 10x10 | 256 | 147 |

|  |  |  |
| --- | --- | --- |
| Dimensiune Matrice | Nr. Thread-uri | Timp (ms) |
| 500X500 | 1 | 42 |
| 500X500 | 2 | 35 |
| 500X500 | 4 | 27 |
| 500X500 | 6 | 26 |
| 500X500 | 8 | 22 |
| 500X500 | 32 | 31 |
| 500X500 | 128 | 53 |
| 500X500 | 256 | 63 |

|  |  |  |
| --- | --- | --- |
| Dimensiune Matrice | Nr. Thread-uri | Timp (ms) |
| 1000x1000 | 1 | 62 |
| 1000x1000 | 2 | 41 |
| 1000x1000 | 4 | 43 |
| 1000x1000 | 6 | 47 |
| 1000x1000 | 8 | 49 |
| 1000x1000 | 32 | 74 |
| 1000x1000 | 128 | 108 |
| 1000x1000 | 256 | 119 |

Limbaj**:** **JAVA**

Operand: **Numar real**

Operator: **a ¤ b = 1/ (1/ a + 1/b)**

|  |  |  |
| --- | --- | --- |
| Dimensiune Matrice | Nr. Thread-uri | Timp (ms) |
| 10x10 | 1 | 1 |
| 10x10 | 2 | 27 |
| 10x10 | 4 | 24 |
| 10x10 | 6 | 23 |
| 10x10 | 8 | 26 |
| 10x10 | 32 | 28 |
| 10x10 | 128 | 39 |
| 10x10 | 256 | 47 |

|  |  |  |
| --- | --- | --- |
| Dimensiune Matrice | Nr. Thread-uri | Timp (ms) |
| 500X500 | 1 | 43 |
| 500X500 | 2 | 36 |
| 500X500 | 4 | 29 |
| 500X500 | 6 | 26 |
| 500X500 | 8 | 27 |
| 500X500 | 32 | 40 |
| 500X500 | 128 | 58 |
| 500X500 | 256 | 62 |

|  |  |  |
| --- | --- | --- |
| Dimensiune Matrice | Nr. Thread-uri | Timp (ms) |
| 1000x1000 | 1 | 49 |
| 1000x1000 | 2 | 47 |
| 1000x1000 | 4 | 46 |
| 1000x1000 | 6 | 52 |
| 1000x1000 | 8 | 44 |
| 1000x1000 | 32 | 69 |
| 1000x1000 | 128 | 96 |
| 1000x1000 | 256 | 151 |

Limbaj**:** **JAVA**

Operand: **Numar Complex**

Operator: **a \* b**

|  |  |  |
| --- | --- | --- |
| Dimensiune Matrice | Nr. Thread-uri | Timp (ms) |
| 10x10 | 1 | 1 |
| 10x10 | 2 | 19 |
| 10x10 | 4 | 19 |
| 10x10 | 6 | 20 |
| 10x10 | 8 | 24 |
| 10x10 | 32 | 31 |
| 10x10 | 128 | 34 |
| 10x10 | 256 | 45 |

|  |  |  |
| --- | --- | --- |
| Dimensiune Matrice | Nr. Thread-uri | Timp (ms) |
| 500X500 | 1 | 31 |
| 500X500 | 2 | 33 |
| 500X500 | 4 | 32 |
| 500X500 | 6 | 28 |
| 500X500 | 8 | 30 |
| 500X500 | 32 | 34 |
| 500X500 | 128 | 48 |
| 500X500 | 256 | 61 |

|  |  |  |
| --- | --- | --- |
| Dimensiune Matrice | Nr. Thread-uri | Timp (ms) |
| 1000x1000 | 1 | 63 |
| 1000x1000 | 2 | 61 |
| 1000x1000 | 4 | 58 |
| 1000x1000 | 6 | 57 |
| 1000x1000 | 8 | 51 |
| 1000x1000 | 32 | 72 |
| 1000x1000 | 128 | 122 |
| 1000x1000 | 256 | 117 |

Limbaj**:** **JAVA**

Operand: **Numar Complex**

Operator: **a ¤ b = 1/ (1/ a + 1/b)**

|  |  |  |
| --- | --- | --- |
| Dimensiune Matrice | Nr. Thread-uri | Timp (ms) |
| 10x10 | 1 | 1 |
| 10x10 | 2 | 15 |
| 10x10 | 4 | 19 |
| 10x10 | 6 | 23 |
| 10x10 | 8 | 24 |
| 10x10 | 32 | 30 |
| 10x10 | 128 | 40 |
| 10x10 | 256 | 50 |

|  |  |  |
| --- | --- | --- |
| Dimensiune Matrice | Nr. Thread-uri | Timp (ms) |
| 500X500 | 1 | 38 |
| 500X500 | 2 | 36 |
| 500X500 | 4 | 29 |
| 500X500 | 6 | 27 |
| 500X500 | 8 | 27 |
| 500X500 | 32 | 25 |
| 500X500 | 128 | 48 |
| 500X500 | 256 | 50 |

|  |  |  |
| --- | --- | --- |
| Dimensiune Matrice | Nr. Thread-uri | Timp (ms) |
| 1000x1000 | 1 | 60 |
| 1000x1000 | 2 | 50 |
| 1000x1000 | 4 | 49 |
| 1000x1000 | 6 | 47 |
| 1000x1000 | 8 | 54 |
| 1000x1000 | 32 | 72 |
| 1000x1000 | 128 | 95 |
| 1000x1000 | 256 | 229 |