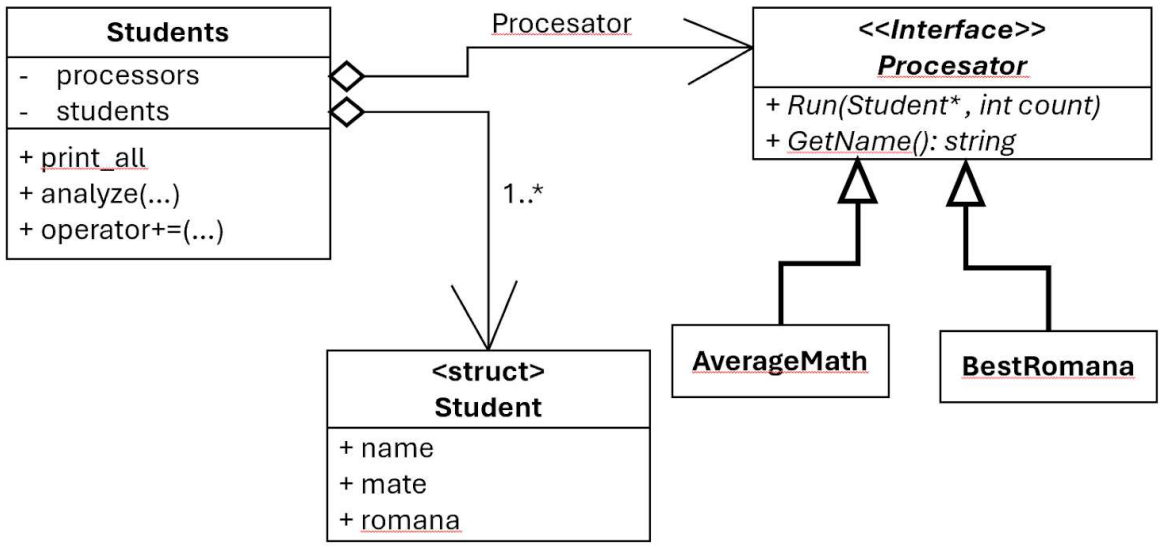
Lab exam (part 2) - P2

Consider the following UML diagram / Fie urmatoarea diagrama UML:



Build all the files with the classes described in the above diagram so that the following code / Construiți fișierele header si cpp corespunzătoare diagramei de mai sus astfel incat:

```
#include <iostream>
#include "Students.h"
#include "BestRomana.h"
#include "AverageMath.h"

int main()
{
    Students s({{"Dan",10,9},{"John",9,4},{"Mike", 5,8}});
    (s += new AverageMath())+=new BestRomana();
    s.print_all();
    s.analyze("AverageMath");
    s.analyze("BestRomana");
    return 0;
}
```

will output upon execution / va scrie pe ecran în urma execuției:

```
Dan Math: 10 Rom: 9
John Math: 9 Rom: 4
Mike Math: 5 Rom: 8
AverageMath = 8
Best romana = Dan with grade: 9
```

Observations/Observatii:

- You **can** use std containers
- You will need to deduce the constructors (if any) for each class from by looking into the main.cpp provided

Grading:

G1	Organize your project in 8 files: **main.cpp**, **Procesator.h**, **Student.h**, **Students.h**, **Students.cpp**, **AverageMath.h**, **AverageMath.cpp**, **BestRomana.h**, **BestRomana.cpp**	2p
G2	The files **Procesator.h, AverageMath.h, BestRomana.h, Procesator.h** includes a correct C++ implementation of the UML diagram	3p
G3	Implementation of the method **Students::analyze** method	3p
G4	Implementation of the method **Student** structure	1p
G6	Implementation of the method **print_all** for **Students**	3p
G6	Implementation of the method **operator+=** for Students(to add processors)	3p
G7	**Students** constructor	4p
G8	Correct implementation of **AverageMeth** procesator	3p
G9	Correct implementation of **BestRomana** procesator	4p
G10	Correct implementation for Procesator.h	1p
G11	The program compiles and upon execution produces the expected results	3p