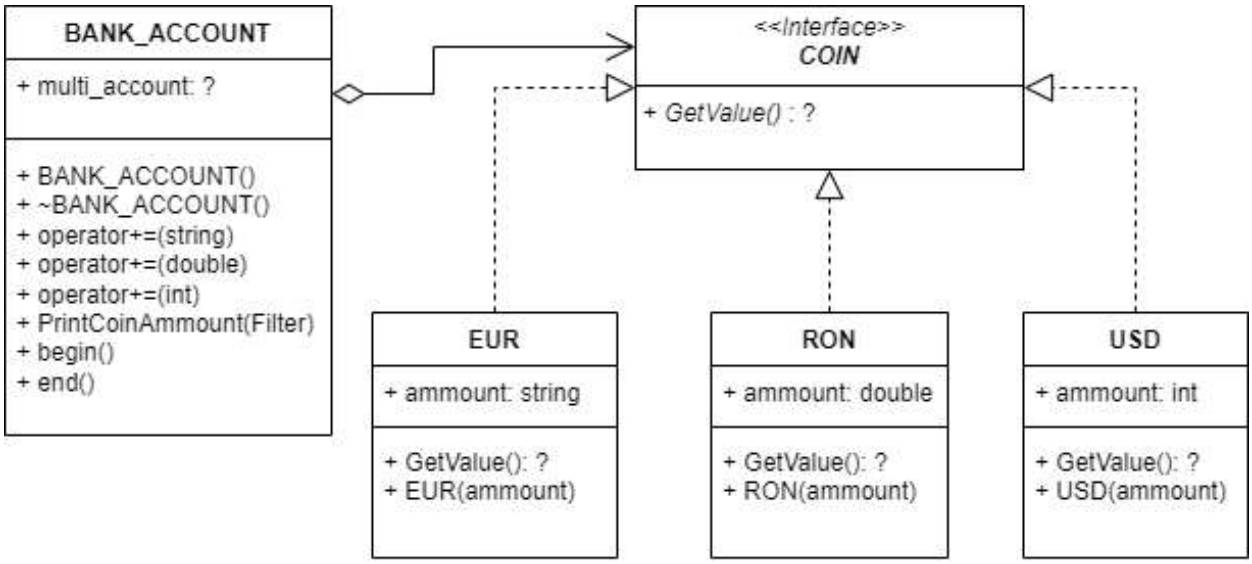
Lab exam (part 2) - P3

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:

```
int main()
{
    BANK_ACCOUNT my_account;

    my_account += "210";
    my_account += 411.99;
    my_account += 455;
    my_account += 300;
    my_account += 607.25;
    my_account += "1210";

    for (auto it : my_account)
    {
        cout << it->GetValue();
        if (dynamic_cast<EUR*>(it) != 0)
            cout << " EUR";
        else
            if (dynamic_cast<RON*>(it) != 0)
                cout << " RON";
            else
                cout << " USD";
        cout << endl;
    }

    cout << "In my multi bank account I have ";
    my_account.PrintCoinAmmount([](COIN* param)->bool {
        return dynamic_cast<USD*>(param) != 0;
        });
    cout << " $ and ";
    my_account.PrintCoinAmmount([](COIN* param)->bool {
        return dynamic_cast<RON*>(param) != 0;
        });
    cout << " RON" << endl;
    return 0;
}
```

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

```
210 EUR
411.99 RON
455 USD
300 USD
607.25 RON
1210 EUR
In my multi bank account I have 755.00 $ and 1019.24 RON
```

Observations/Observatii:

- You **can** use std containers
- You will need to deduce the constructors and some data types for each class by looking into the main.cpp provided, and the output

Grading:

G1	Organize your project in 10 files: **main.cpp**, **COIN.h**, **BANK_ACCOUNT.h**, **BANK_ACCOUNT.h**, **EUR.cpp**, **EUR.h**, **USD.cpp**, **USD.h**, **RON.cpp, RON.h**	2p
G2	The code inside **BANK_ACCOUNT.h, EUR.h, RON.h, USD.h, COIN.h** includes correct C++ dependencies found in the UML diagram	3p
G3	Implementation of the method **BANK_ACCOUNT::PrintCoinAmmount** method	4p
G4	Implementation of the **begin(), end()** methods in **BANK_ACCOUNT**	3p
G6	Implementation of the method **operator+=(string)**	3p
G6	Implementation of the method **operator+=(double)**	3p
G7	Implementation of the method **operator+=(int)**	3p
G8	Correct implementation of USD class	2p
G9	Correct implementation of EUR class	2p
G10	Correct implementation of RON class	2p
G11	The program compiles and upon execution produces the expected results	3p