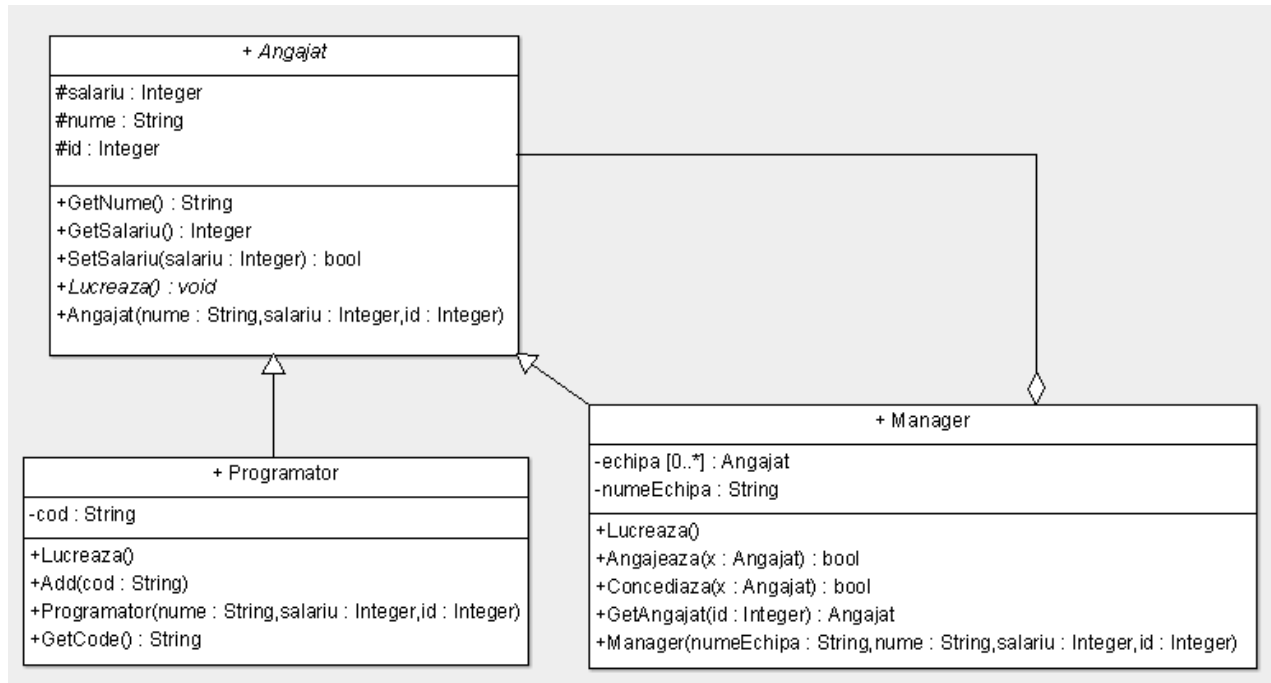


SUBIECT 2

Fie următoarea diagrama UML:



Sa se implementeze ierarhia de clase astfel încât codul din `main()` descris mai jos:

COD main()

```
int main()
{
    Programator petru("Petr", 1000, 0);
    Programator maria("Maria", 1100, 1);

    Manager mihai("C++", "Mihai", 1200, 2);

    mihai.Angajeaza(&maria);
    mihai.Angajeaza(&petru);

    petru.Add("int main()\n{\n");
    maria.Add("std::cout<<\nHello World\n";\n");

    mihai.Lucreaza();

    Programator george("George", 900, 3);
    george.Add(petru.GetCode());
    george.Add(maria.GetCode());
    george.Add("\n\n");

    mihai.Concediaza(&petru);
    mihai.Angajeaza(&george);

    mihai.Lucreaza();
}
```

să afișeze:

```
=====
C++
=====
C++
[Maria]:
std::cout<<"Hello World";
[Petru]:
int main()
{
=====
=====
C++
[Maria]:
std::cout<<"Hello World";
[George]:
int main()
{
std::cout<<"Hello World";

}
=====
```

Barem:

- 5p - Aranjarea codului în headere și fișiere cpp corespunzătoare.
- 4p - Scrierea clasei abstracte Angajat (constructor, membrii, metodele non-virtuale)
- 4p - Clasa Programator
- 2p- Clasa Manager (membrii si Constructor)
- 6p - Cate 2 pct pentru implementarea corecta a metodelor Angajeaza, Concediaza, Lucreaza
- 4p- Programul compilează și funcționează conform cerinței