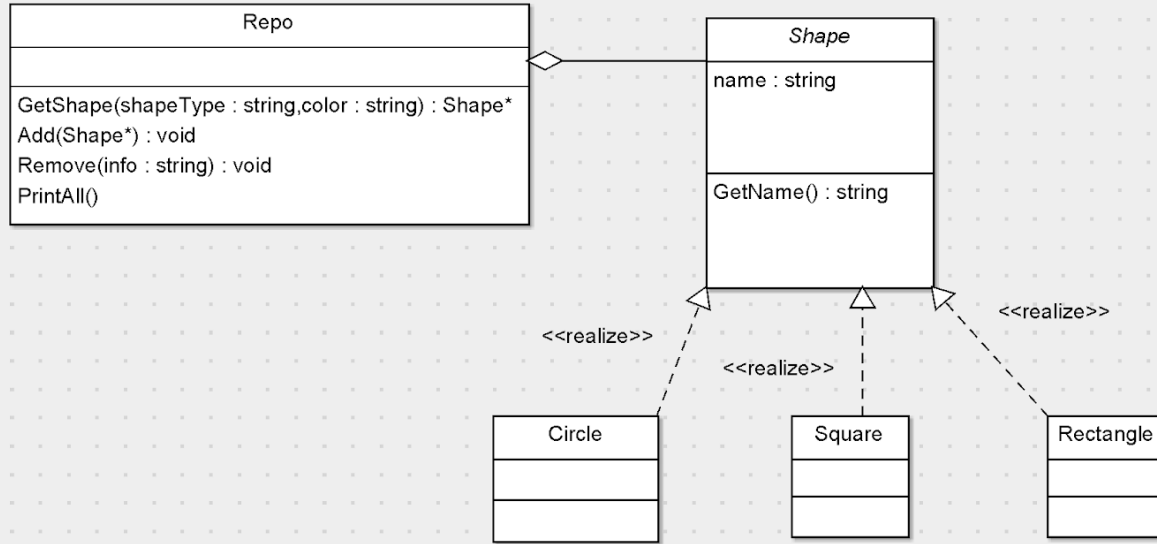


## SUBIECT 4

Fie următoarea diagrama UML:



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

### COD main()

```
int main()
{
    Repo *m = new Repo();

    m->Add(m->GetShape("Circle", "red"));
    m->Add(m->GetShape("Square", "yellow"));
    m->Add(m->GetShape("Rectangle", "blue"));
    m->Add(m->GetShape("Rectangle", "red"));
    m->Add(m->GetShape("Circle", "green"));

    m->PrintAll();
    m->Remove("Circle");
    m->PrintAll();

    m->Add(new Circle("green"));
    m->Add(new Rectangle("red"));
    m->Remove("red");
    m->PrintAll();

    return 0;
}
```

să afișeze:

```
Shape: Circle
Color: red
Shape: Square
Color: yellow
Shape: Rectangle
Color: blue
Shape: Rectangle
Color: red
```

```
Shape: Circle
Color: green
----
Shape: Square
Color: yellow
Shape: Rectangle
Color: blue
Shape: Rectangle
Color: red
----
Shape: Square
Color: yellow
Shape: Rectangle
Color: blue
Shape: Circle
Color: green
----
```

Barem:

- 5p - Aranjarea codului în headere și fișiere cpp corespunzătoare.
- 4p - Scrierea clasei abstracte Shape
- 2p - Scrierea clasei Rectangle
- 2p - Scrierea clasei Square
- 2p - Scrierea clasei Circle
- 5p - Scrierea corectă a funcției GetShape
- 3p - Scrierea corectă a funcției Remove
- 1p - Scrierea corectă a funcției Add
- 1p - Scrierea corectă a funcției PrintAll