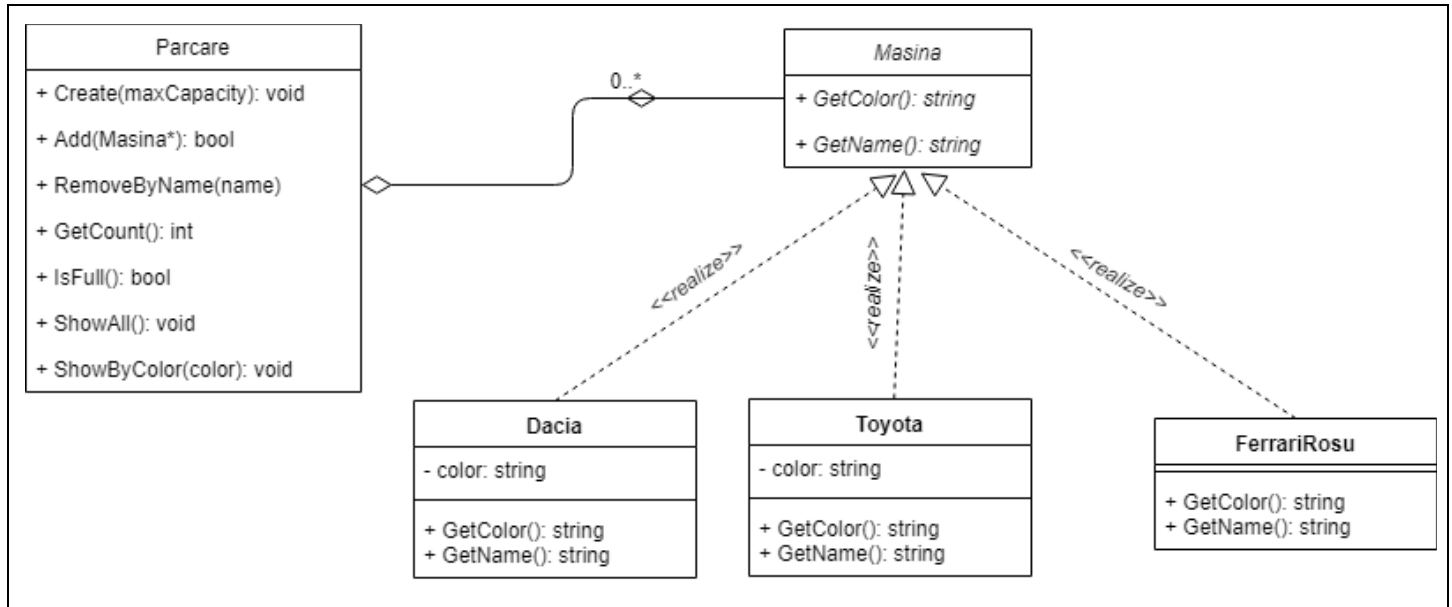


SUBIECT 1

Fie următoarea diagrama UML:



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

COD main()

```
int main() {
    Parcare p;
    p.Create(4);
    p.Add(new Dacia("rosu"));
    p.Add(new Toyota("verde"));
    cout << "Masini in parcare:" << p.GetCount() << boolalpha << " , Parcare plina: " << p.IsFull() << endl;
    p.Add(new FerrariRosu());
    p.Add(new Toyota("albastru"));
    p.Add(new Toyota("rosu")); // returns false - s-a atins capacitatea maxima
    cout << "Masini in parcare:" << p.GetCount() << boolalpha << " , Parcare plina: " << p.IsFull() << endl;
    p.ShowAll();
    p.ShowByColor("rosu");
    p.RemoveByName("Toyota");
    p.Add(new Dacia("gri"));
    p.ShowAll();
    cout << "Masini in parcare:" << p.GetCount() << boolalpha << " , Parcare plina: " << p.IsFull() << endl;
}
```

să afișeze:

```
Masini in parcare:2 , Parcare plina: false
Masini in parcare:4 , Parcare plina: true
SHOW-ALL:Dacia(rosu), Toyota(verde), Ferrari(rosu), Toyota(albastru),
SHOW-COLOR (rosu):Dacia, Ferrari,
SHOW-ALL:Dacia(rosu), Ferrari(rosu), Dacia(gri),
Masini in parcare:3 , Parcare plina: false
```

Barem:

- 5p - Aranjarea codului în headere și fișiere cpp corespunzătoare.
- 3p - Scrierea clasei abstracte Masina
- 6p - Clasele Dacia, Toyota si FerrariRosu (cate 2 pct pentru fiecare)
- 8p - Cate 2 pct pentru implementarea corecta a functiile ShowByColor, ShowAll, RemoveByName, Add din clasa Parcare
- 3p - Cate 1 pct pentru implementarea corecta a funcțiile Create, IsFull, GetCount din clasa Parcare