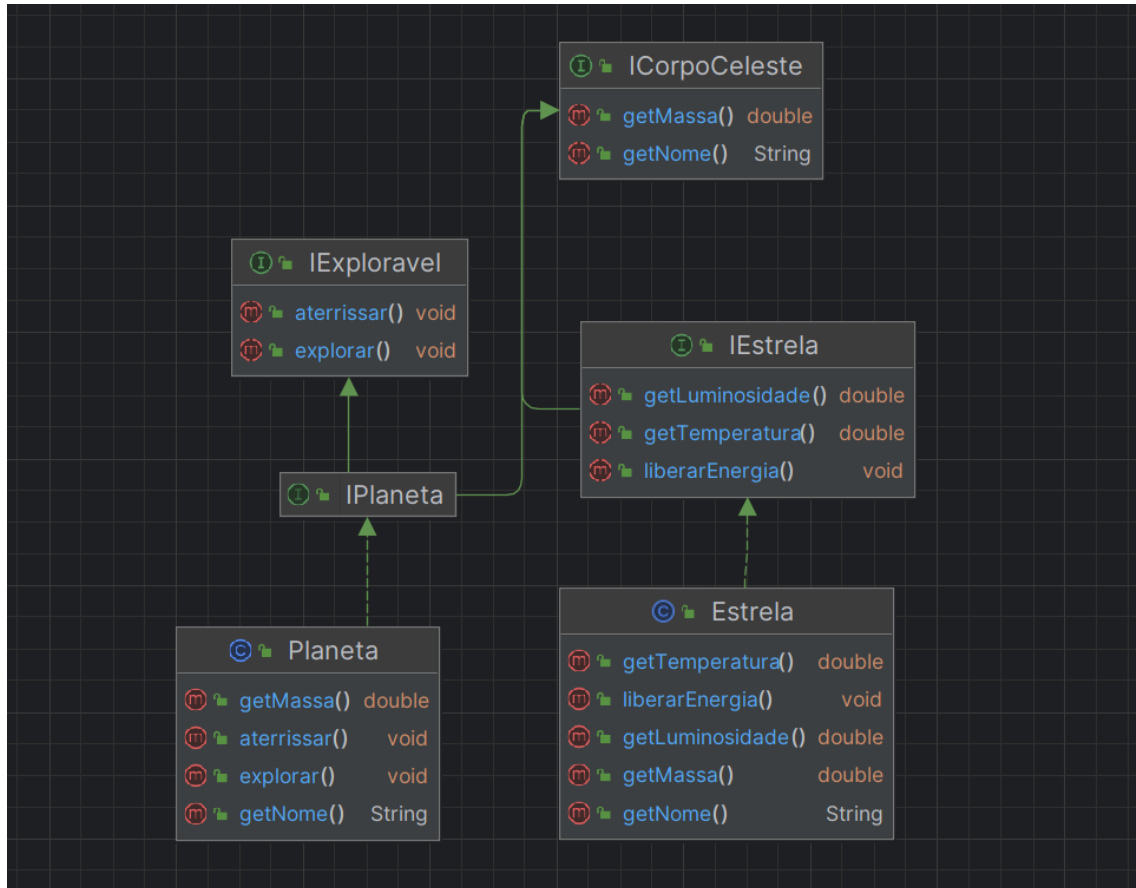


Aqui nesse exemplo a gente tem uma interface corpo celeste que vai ser usada para implementar as classes planeta e estrela, fazendo dessa forma tanto a classe estrela quanto a classe planeta são obrigadas a implementar métodos que não fazem sentido pra sua classe o que vai gerar exceções

```
no usages
37 ④ public void aterrissar() {
38      throw new UnsupportedOperationException("Método 'aterrissar' não implementado.");
39  }
40
no usages
41  @Override
42  ④ public void explorar() {
43      throw new UnsupportedOperationException("Método 'explorar' não implementado.");
44  }
45 }
```

```
no usages
31  @Override
32  ④ public double getLuminosidade() {
33      throw new UnsupportedOperationException("Método não implementado.");
34  }
no usages
35  ④ public void liberarEnergia(){
36      throw new UnsupportedOperationException("Método não implementado.");
37  }
38  }
39 }
```

Então seguindo o princípio da segregação de interfaces a implementação correta seria essa, em que separamos as interfaces para que cada classe tenha que seguir apenas os métodos que vão utilizar:



```
© Main.java  ⓘ ICorpoCeleste.java × ⓘ IExploravel.java
2 usages 4 implementations
1 ⓘ public interface ICorpoCeleste {
    no usages 2 implementations
2 ⓘ     String getNome();
    no usages 2 implementations
3 ⓘ     double getMassa();
4     }
```

```
© Main.java  ⓘ IExploravel.java × ⓘ ICorpoCeleste.java
1 usage 2 implementations
1 ⓘ public interface IExploravel {
    no usages 1 implementation
2 ⓘ     void aterrissar();
    no usages 1 implementation
3 ⓘ     void explorar();
4     }
```

```
© Main.java  ⓘ IPlaneta.java × ⓘ IExploravel.java ⓘ ICorpoCeleste.java
1 usage 1 implementation
1 ⓘ public interface IPlaneta extends ICorpoCeleste, IExploravel{
2     }
3
```

```
© Main.java  ⓘ IEstrela.java × ⓘ IPlaneta.java ⓘ IExploravel.java
1 usage 1 implementation
1 ⓘ public interface IEstrela extends ICorpoCeleste{
    no usages 1 implementation
2 ⓘ     double getTemperatura();
    no usages 1 implementation
3 ⓘ     double getLuminosidade();
    no usages 1 implementation
4 ⓘ     void liberarEnergia();
5     }
```

```
© Main.java  © Planeta.java ×  ⓘ IEstrela.java  ⓘ IPlaneta.java  ⓘ IExploravel.java  ⓘ ICorpoCeleste.java

no usages
1 public class Planeta implements IPlaneta{
    4 usages
2     private String nome;
    2 usages
3     private double massa;
4
    no usages
5     public Planeta(String nome, double massa) {
6         this.nome = nome;
7         this.massa = massa;
8     }
9
    no usages
10    @Override
11    public String getNome() {
12        return nome;
13    }
14
    no usages
15    @Override
16    public double getMassa() { return massa; }
17
    no usages
20    public void aterrissar() { System.out.println("Aterrissando no planeta " + nome); }
21
    no usages
24    @Override
25    public void explorar() { System.out.println("Explorando o planeta " + nome); }
26
27 }
28
29
```

```
© Main.java  © Estrela.java ×  © Planeta.java  ⓘ IEstrela.java  ⓘ IPlaneta.java  ⓘ IExploravel.java

no usages
1 public class Estrela implements IEstrela {
    2 usages
2     private String nome;
    2 usages
3     private double massa;
    2 usages
4     private double temperatura;
    2 usages
5     private double luminosidade;
6
    no usages
7     public Estrela(String nome, double massa, double temperatura, double luminosidade) {
8         this.nome = nome;
9         this.massa = massa;
10        this.temperatura = temperatura;
11        this.luminosidade = luminosidade;
12    }
13
14
```

```
14         no usages
15         @Override
16         public String getNome() { return nome; }
17
18
19         no usages
20         @Override
21         public double getMassa() { return massa; }
22
23
24         no usages
25         @Override
26         public double getTemperatura() { return temperatura; }
27
28
29         no usages
30         @Override
31         public double getLuminosidade() { return luminosidade; }
32
33
34         no usages
35         public void liberarEnergia() { System.out.println("Liberando energia estelar."); }
36     }
37
38
```