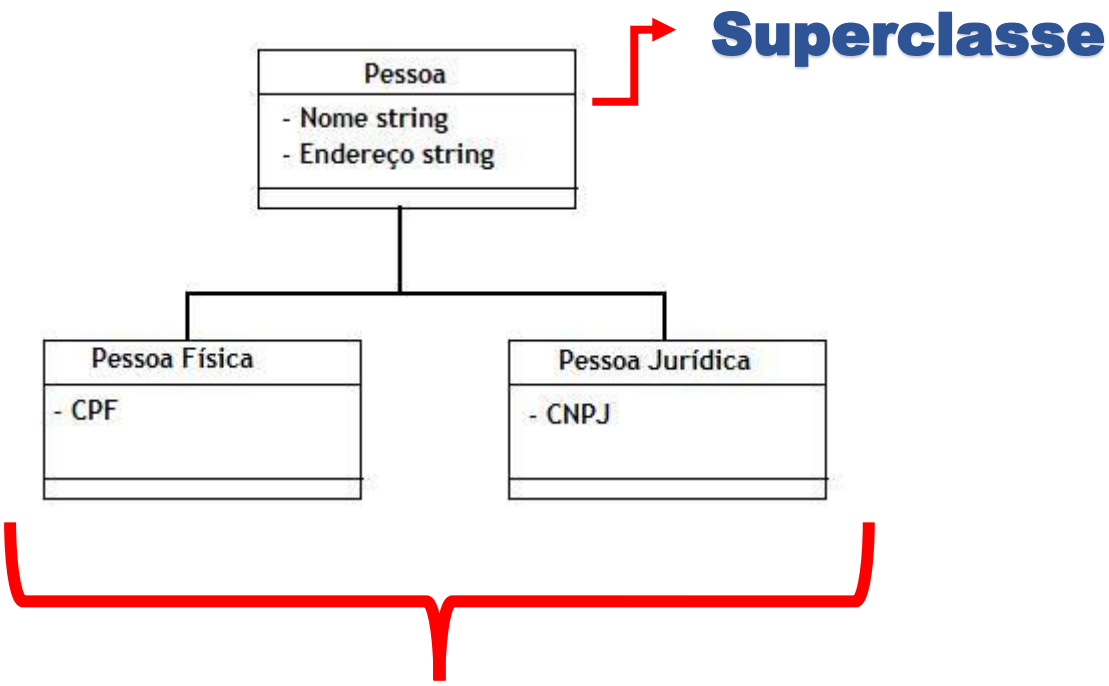
Atividade 4

1. Herança

Com a definição de classes, obtemos o princípio de reutilização de código através da herança, assim podemos definir um objeto genérico e dividi-lo em diversos tipos daquele mesmo modelo, como o exemplo abaixo:



Com o conceito de herança, temos uma classe Mãe que carrega seus atributos comuns daquele tipo de classe, mas também através de herança, uma classe filha, contem não apenas seus atributos específicos, mas também herda todas os atributos e métodos da superclasse Mãe. Através da denotação:

```
extends
```

2. Desenvolvendo nosso script.

Abra o seu editor de códigos fonte e desenvolva o seguinte script no mesmo diretório que a atividade anterior:

script-exemplo-herança-pessoa.ts

```
1.    export class Pessoa {
2.
3.        // Definindo atributos
4.        public nome: string;
5.        public endereco: string;
6.
7.        constructor(nome: string, endereco: string){
8.            this.nome = nome;
9.            this.endereco = endereco;
10.       }
11.
12.
13.
14.   }
```

Próxima página, classe filha!

script-exemplo-herança-classe-fisica.ts

```
15.   import { Pessoa } from 'script-exemplo-heranca-pessoa';
16.
17.   export class PessoaFisica extends Pessoa{
18.
19.       // Definindo atributos
20.       public cpf: string;
21.
22.       constructor(nome: string, endereco: string, cpf: string){
23.           super(nome, endereco);
24.           this.cpf = cpf;
25.       }
26.
27.
28.   }
```

Próxima página classe filha Jurídica!

script-exemplo-herança-juridica.ts

```
29.   import { Pessoa } from 'script-exemplo-heranca-pessoa';
30.
31.
32.   export class PessoaJuridica extends Pessoa{
33.
34.       // Definindo atributos
35.       public cnpj: string;
36.
37.       constructor(nome: string, endereco: string, cnpj: string){
38.           super(nome, endereco);
39.           this.cnpj = cnpj;
40.       }
41.
42.
43.   }
```

FIM!