

Para saber mais: Tipos Herança

Agora que já entendemos sobre herança e conhecemos o modificador `protected` , podemos falar sobre os tipos de herança: Pública, protegida e privada.

Repare, novamente, na sintaxe utilizada para realizar herança:

```
class Funcionario : public Pessoa
{
...
};
```

[COPIAR CÓDIGO](#)

Discutimos anteriormente o que significa o uso do `:` nesse contexto. E o uso de `public` aqui informa que tudo que é público na classe `Pessoa` permanecerá público na classe `Funcionario` .

Se fizéssemos `class Funcionario : protected Pessoa` , tudo que na classe `Pessoa` é público, se torna protegido (`protected`) na classe `Funcionario` .

E como você já deve imaginar, se fizermos `class Funcionario : private Pessoa` , todos os membros públicos e protegidos em `Pessoa` se tornam privados em `Funcionario` .

É muito mais comum termos herança usando `:` `public` pois ele não altera nada, mas é importante conhecer os demais tipos.

Também é válido ressaltar que os membros privados continuam existindo na classe `Funcionario` , mas não são acessíveis. Somente através de métodos

públicos e protegidos nós podemos acessar os membros definidos como privados em Pessoa .