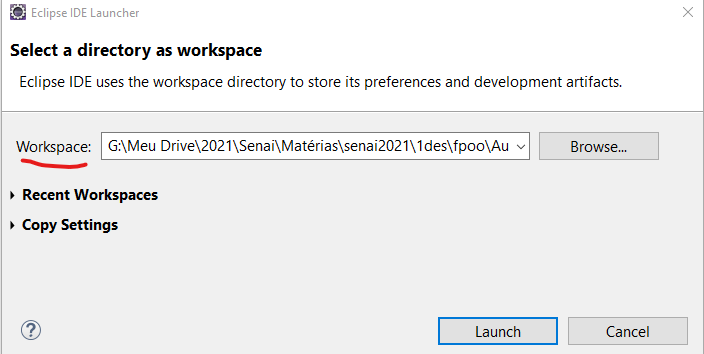
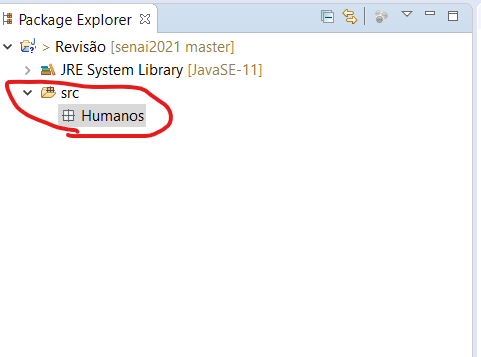
* **Workspace (área de trabalho)**

A pasta principal do projeto. Pode mudar o workspace do projeto.



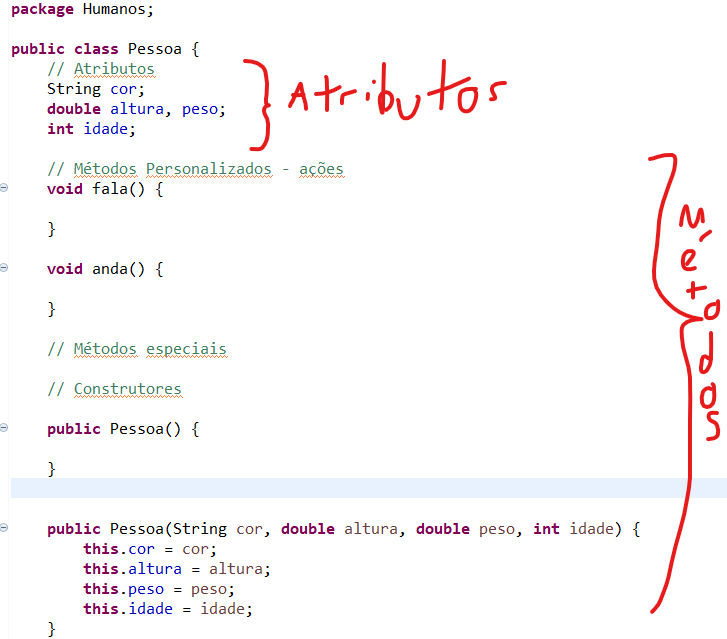
* **Package (Pacote)**

Subpasta do projeto para organização das classes. O pacote deve ficar dentro do workspace, na pasta src (source – fonte/origem). Exemplo de um pacote **Humanos**:



* **Class (Classe)**

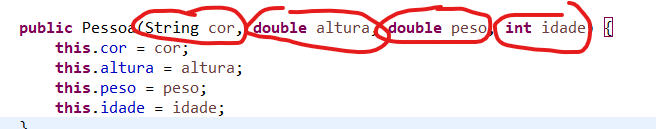
Classe é um conjunto de características (atributos) e ações (métodos) de algo. Exemplo classe Pessoa:



* **Argumento (sinônimo de parâmetro)**

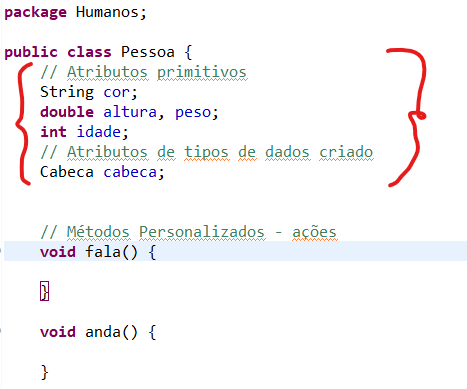
Variável que receberá valor de forma temporária em um método (função). Sempre os argumentos estão entre parênteses. Separar os argumentos por vírgula e colocar o tipo do argumento corretamente.

Exemplo de um método construtor e seus parâmetros



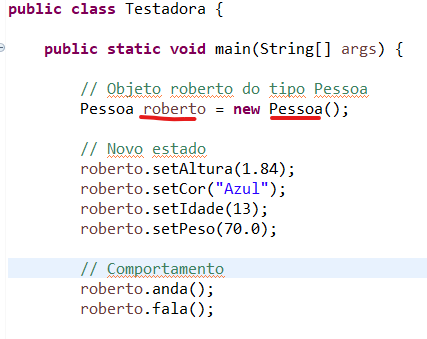
* **Atributo**

São as características de uma classe. Deve-se definir o tipo de dados dos atributos, tipos primitivos (int, double, char,String) ou novos tipos (exemplo classe criada no projeto: Humano, Funcionario,...).



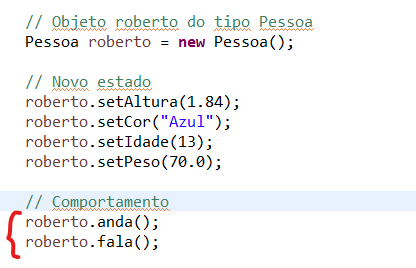
* **Objeto**

Uma entidade de uma classe que possui **estado** e **comportamento**. Os **estados** são os valores que os atributos assumem. O **comportamento** é o método (ação). O objeto é criado/instanciado com a palavra reservada new + o nome da classe.



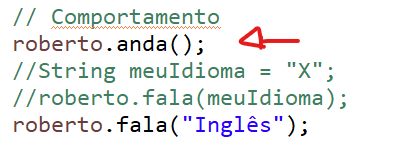
* **Método**

Ação do objeto.



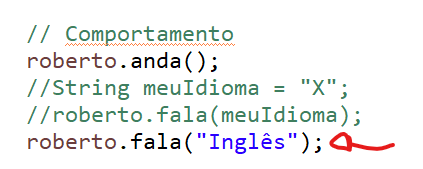
* **Método sem argumento**

O objeto roberto com o método anda **não tem** nenhum argumento dentro dos parênteses.



* **Método com argumento**

O objeto roberto o método fala **com** argumento dentro dos parênteses.

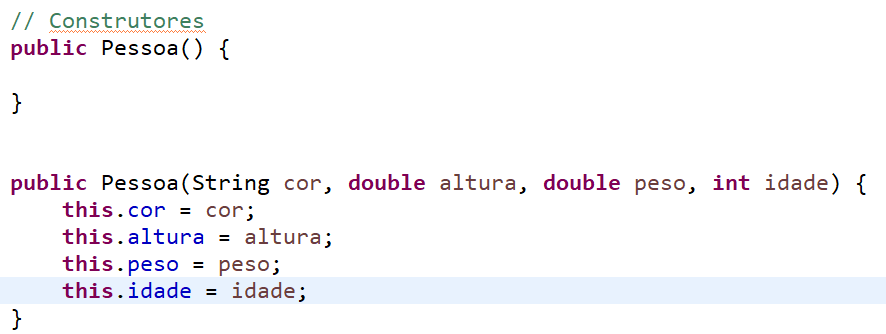


* **Método construtor**

Método que é chamado logo que é criado/instanciado um objeto. **Método construtor tem o mesmo nome da classe**. Métodos construtores não retornam.

Podem ter métodos construtores sem e com argumentos. Precisa colocar os argumentos na ordem que for definida na classe, para chamar corretamente o método.

Exemplo construtores da classe **Pessoa**:



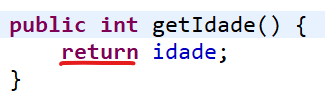
* **Método com retorno**

Método que retorna um valor. Dentro do método precisa ter a palavra **return**. Precisa especificar o tipo de dados que irá retornar antes do nome do método.

Chamada do método com retorno:

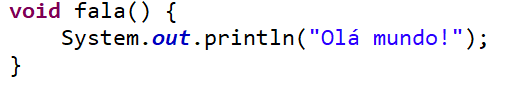


Definição do método com retorno:



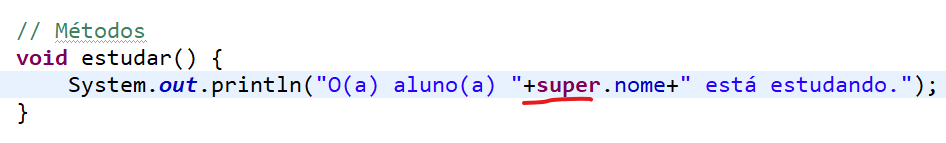
* **Void (vazia)**

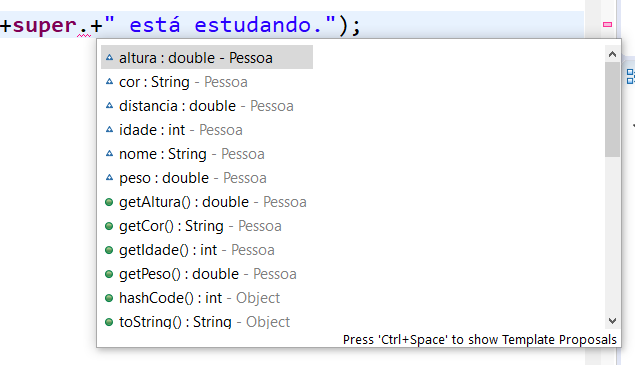
Informar que o método não retorna qualquer valor. Não utilizo a palavra **return** dentro do método.



* **Super**

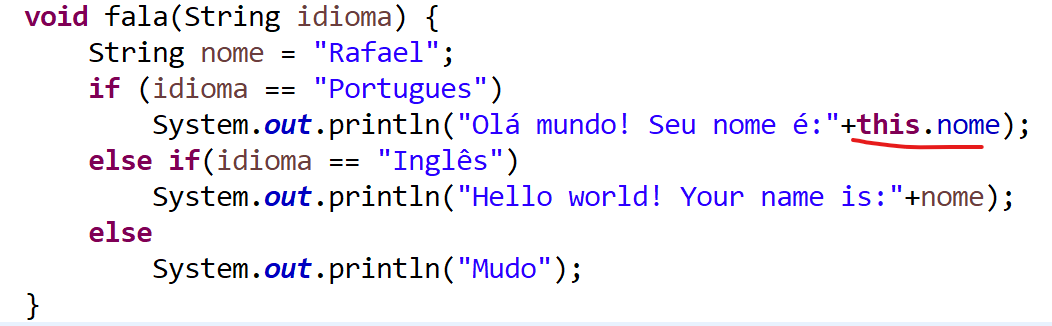
Referência a classe mãe (super classe), acessando seus atributos e métodos. No eclipse, após é sugerido os atributos (triangulo azul) e métodos (bola verde) ao lado.





* **This**

Força o acesso aos atributos e métodos da própria classe. No exemplo da imagem está acessando o atributo **nome** da classe Pessoa.

****

* Private
* Public
* Protected
* Extend (Herança)
* Import