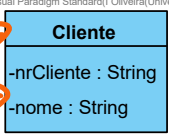
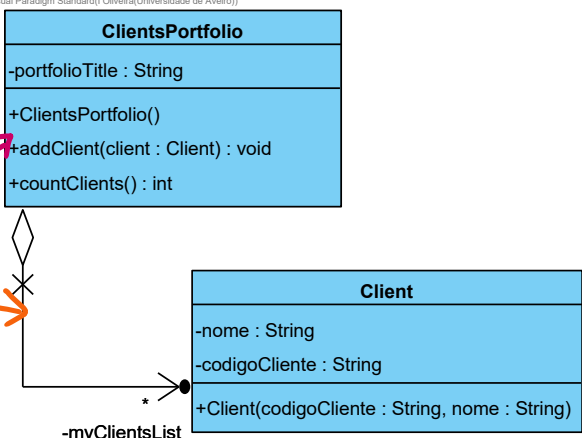
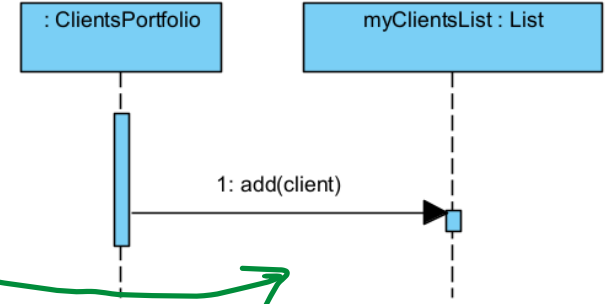


Suporte: correspondência de conceitos entre código por objetos (em Java) e as construções da UML

<pre>public class Cliente { private String nrCliente; private String nome; }</pre>	<p>Visual Paradigm Standard (Universidade de Aveiro)</p>  <pre>classDiagram class Cliente { -nrCliente : String -nome : String }</pre> <p>→ As classes em código são definidas num bloco “class”</p> <p>→ A classe “Cliente” tem o atributo “nrCliente” do tipo <i>String</i> e nível de acesso privado (sinal “-”).</p>
<pre>public class ClientsPortfolio { private String portfolioTitle; private List<Client> myClientsList; public ClientsPortfolio(String initialTitle) { portfolioTitle = initialTitle; myClientsList = new ArrayList<>(); } public void addClient(Client client) { myClientsList.add(client); } public int countClients() { return myClientsList.size(); } }</pre>	<p>Visual Paradigm Standard (Universidade de Aveiro)</p>  <pre>classDiagram class ClientsPortfolio { -portfolioTitle : String +ClientsPortfolio() +addClient(client : Client) : void +countClients() : int } class Client { -nome : String -codigoCliente : String +Client(codigoCliente : String, nome : String) } ClientsPortfolio "1" *-- "*" Client : -myClientsList</pre> <p>→ (Objetos de) A classe ClientsPortfolio guarda dois atributos; um deles, representa uma coleção de objetos Client e pode ser modelado como uma associação. (O nome do atributo em ClientsPortfolio pode ser usado como nome do <i>role</i> na associação, do lado Client)</p> <p>→ A classe incluir as operações addClient() com um parâmetro e sem retorno; e a operação countClients() que retorna um inteiro.</p>
<pre>public class ClientsPortfolio { private List<Client> myClientsList; // partes omitidas public void addClient(Client client) { myClientsList.add(client); } }</pre>	<p>Visual Paradigm Standard (Universidade de Aveiro)</p>  <pre>sequenceDiagram participant CP as : ClientsPortfolio participant ML as myClientsList : List CP->>ML: 1: add(client)</pre> <p>→ Um objeto pode solicitar uma operação noutro, i.e., envia uma mensagem (no caso, no contexto do objeto de <u>ClientsPortfolio</u>, está a pedir ao objeto <u>myClientsList</u> a execução da operação <u>add</u>.)</p>

