









Network
<ul style="list-style-type: none"> <li>- payments: long</li> <li>- debts: long</li> </ul>
<ul style="list-style-type: none"> <li>+ disableClientNotifications(clientID: String): void</li> <li>+ enableClientNotifications(clientID: String): void</li> <li>+ registerClient(id: String, name: String, taxId: int): Client</li> <li>+ getAllClients(): Client[]</li> <li>+ getClient(clientId: String): Client</li> <li>+ getClientPayments(clientId: String): long</li> <li>+ getClientDebts(clientId: String): long</li> <li>+ getAllCommunications(): Communication[]</li> <li>+ getClientsWithDebts(): Client[]</li> <li>+ getClientsWithoutDebts(): Client[]</li> <li>+ getCommunicationsFromClient(clientId: String): Communication[]</li> <li>+ getCommunicationsToClient(clientId: String): Communication[]</li> <li>+ getTerminalsWithPositiveBalance(): Terminal[]</li> <li>+ getUnusedTerminals(): Terminal[]</li> <li>+ getGlobalBalance(): long</li> <li>+ registerTerminal(terminalId: String, type: String, clientId: String): Terminal</li> <li>+ getAllTerminals(): Terminal[]</li> <li>+ importFile(filename: String): void</li> </ul>

Client
<ul style="list-style-type: none"> <li>- &lt;&lt;final&gt;&gt; id: String</li> <li>- &lt;&lt;final&gt;&gt; name: String</li> <li>- &lt;&lt;final&gt;&gt; taxId: int</li> <li>- payments: long</li> <li>- debts: long</li> <li>- receiveNotifications: boolean</li> <li>- tariffPlan: TariffPlan</li> </ul>

GoldType
<ul style="list-style-type: none"> <li>- consecutiveVideoCommunications: int</li> </ul>

PlatinumType
<ul style="list-style-type: none"> <li>- consecutiveTextCommunications: int</li> </ul>

<<abstract>> ClientType
<ul style="list-style-type: none"> <li>- name: String</li> </ul>
<ul style="list-style-type: none"> <li>+ &lt;&lt;abstract&gt;&gt; checkTransitionAfterPayment(): void</li> <li>+ &lt;&lt;abstract&gt;&gt; checkTransitionAfterCommunication(communicationType: String): void</li> <li>+ &lt;&lt;abstract&gt;&gt; upgrade(): void</li> <li>+ &lt;&lt;abstract&gt;&gt; downgrade(): void</li> </ul>

Network Manager
<ul style="list-style-type: none"> <li>+ load(filename: String): void</li> <li>+ save(): void</li> <li>+ saveAs(filename: String): void</li> <li>+ importFile(filename: String): void</li> </ul>

<<interface>> Notification Delivery Method
<ul style="list-style-type: none"> <li>+ deliver(notification: Notification): void</li> </ul>

Notification
<ul style="list-style-type: none"> <li>- &lt;&lt;final&gt;&gt; type: String</li> </ul>

Interfaces Label, Message and Prompt (various packages) have been omitted since they represent bags of global functions or variables (specific to the implementation and not relevant for the overall structure of the project). Class serial numbers for serialization, methods "toString" and "equals", getters and setters have also been omitted. All classes except NetworkManager implement Serializable.



«abstract» Terminal
<ul style="list-style-type: none"> <li>- «final» id: String</li> <li>- «final» type: String</li> <li>- payments: long</li> <li>- debts: long</li> </ul>
<ul style="list-style-type: none"> <li>+ addFriend (terminalId: String): void</li> <li>+ endInteractive Communication (duration: long): Communication</li> <li>+ perform Payment (commId: int): void</li> <li>+ removeFriend (terminalId: String): void</li> <li>+ sendText Communication (terminalId: String, message: String): Communication</li> <li>+ getOngoing Communication(): Communication</li> <li>+ getTerminal Balance(): long</li> <li>+ silenceTerminal(): void</li> <li>+ startInteractiveCommunication (terminalId: String, commType: String): Communication</li> <li>+ turnOffTerminal(): void</li> <li>+ turnOnTerminal(): void</li> <li>+ canEndCurrentCommunication(): boolean</li> <li>+ canStartCommunication(): boolean</li> <li>+ registerContactAttempt (terminalId: String, type: String): Notification</li> <li>+ notifySourceTerminal (notification: Notification): void</li> </ul>

«abstract» Terminal State
<ul style="list-style-type: none"> <li>- name: String</li> </ul>
<ul style="list-style-type: none"> <li>+ «abstract» toIdle(): void</li> <li>+ «abstract» toSilent(): void</li> <li>+ «abstract» toBusy(): void</li> <li>+ «abstract» toOff(): void</li> </ul>

«abstract» Communication
<ul style="list-style-type: none"> <li>- Communication Counter: int</li> <li>- «final» id: int</li> <li>- inProgress: boolean</li> <li>- «final» type: String</li> <li>- «final» cost: double</li> <li>- paid: boolean</li> </ul>

Text Communication
<ul style="list-style-type: none"> <li>- «final» message: String</li> </ul>

Interactive Communication
<ul style="list-style-type: none"> <li>- «final» duration: long</li> </ul>

«interface» Tariff Plan
<ul style="list-style-type: none"> <li>+ calculateCommunicationCost (comm: Communication): double</li> </ul>

Declaro por minha honra que este diagrama foi realizado apenas pelos elementos que constituem o grupo de projeto. Afonso Ribeiro 102763

Grupo 053

Declaro por minha honra que este diagrama foi realizado apenas pelos elementos que constituem o grupo de projeto. Nuno Gonçalves 103392