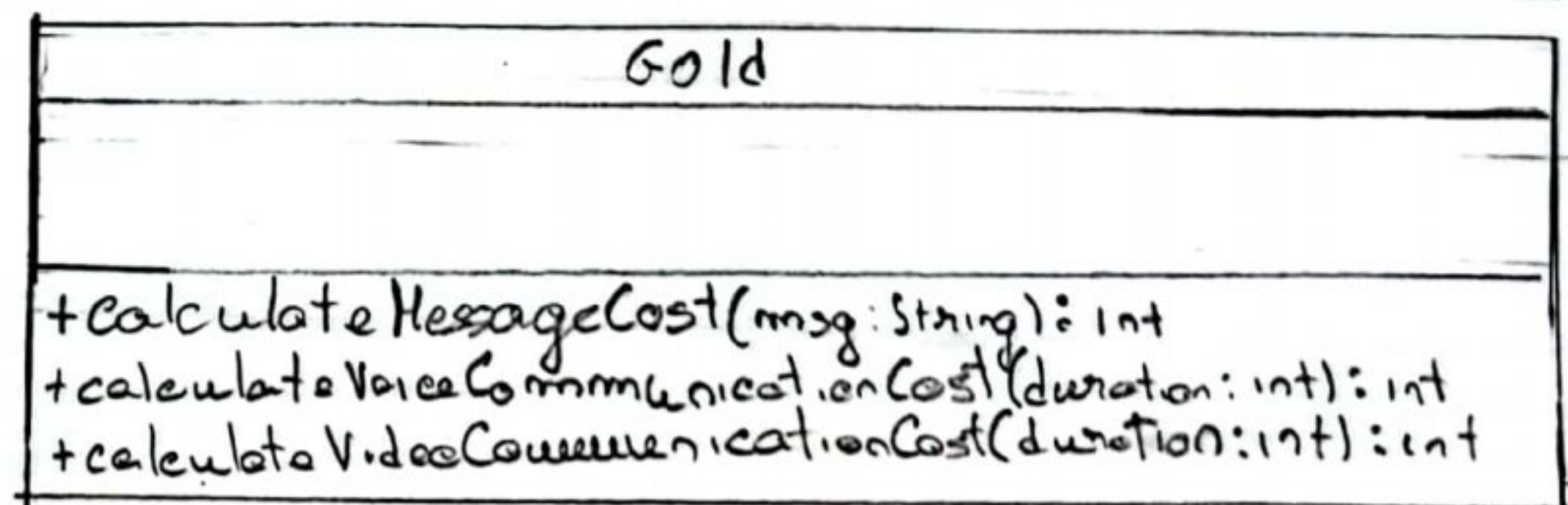
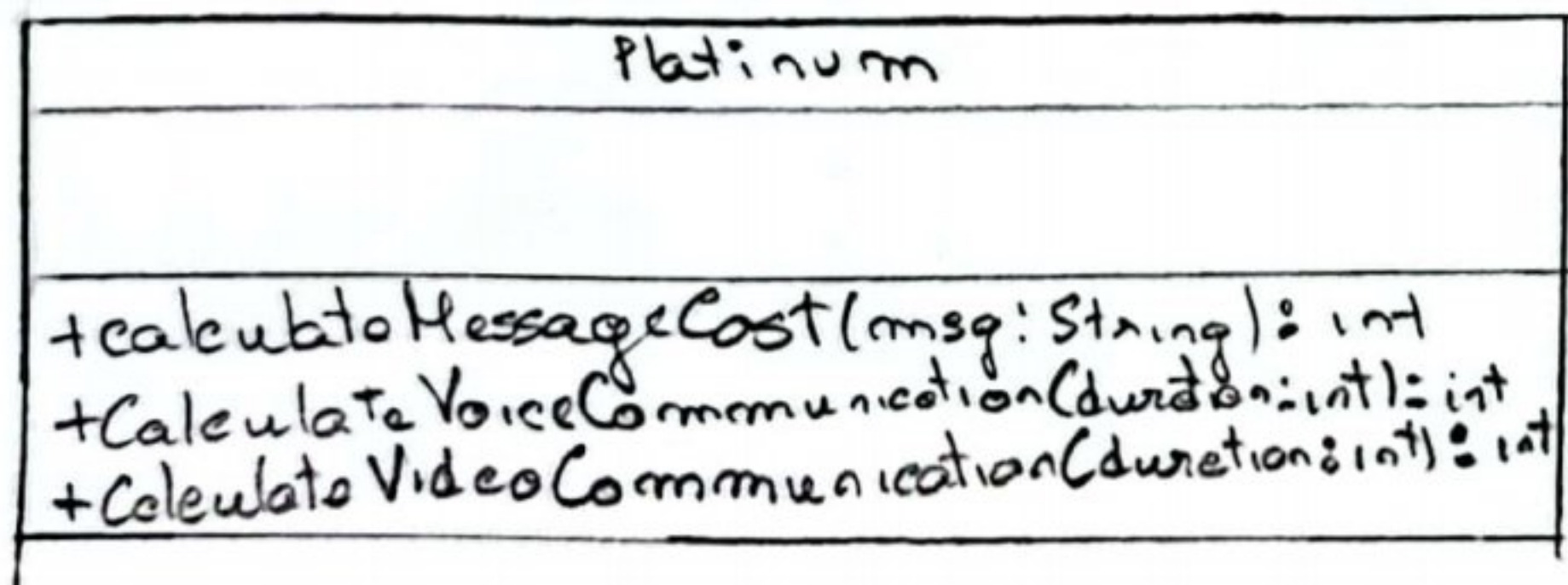
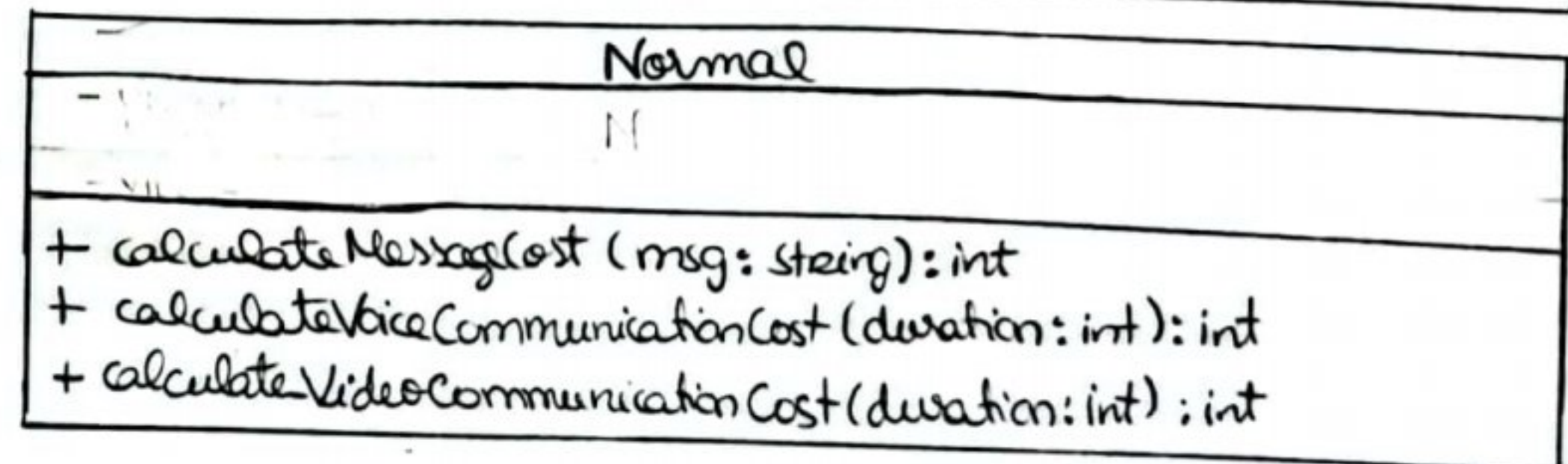
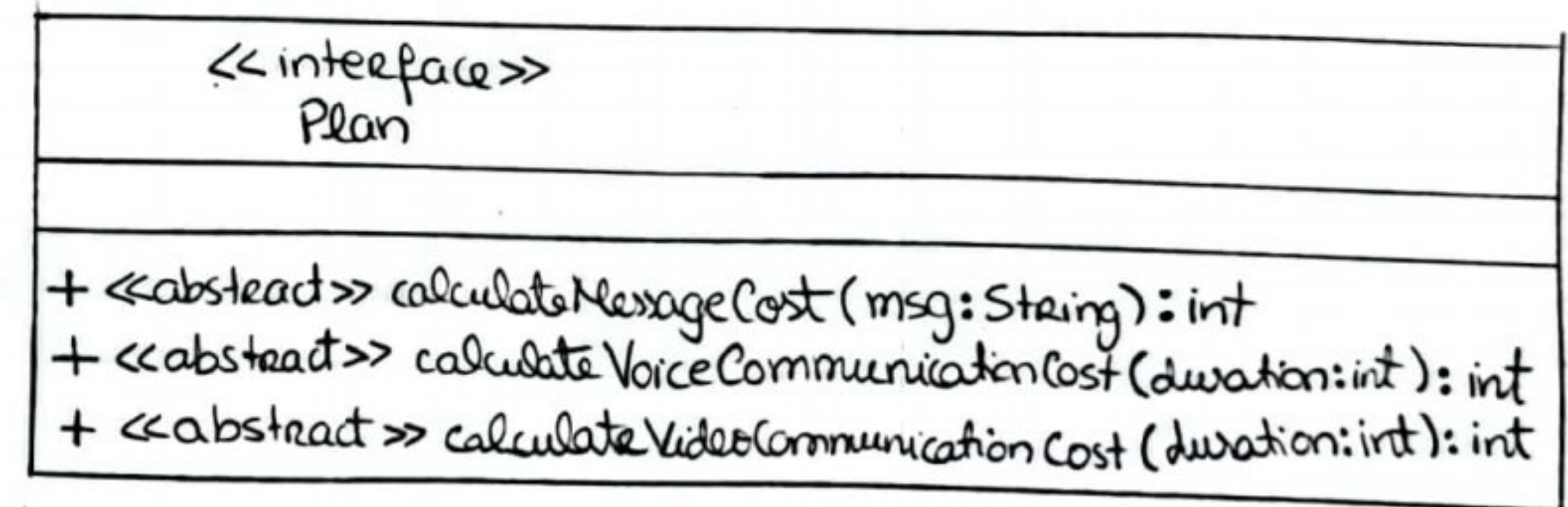
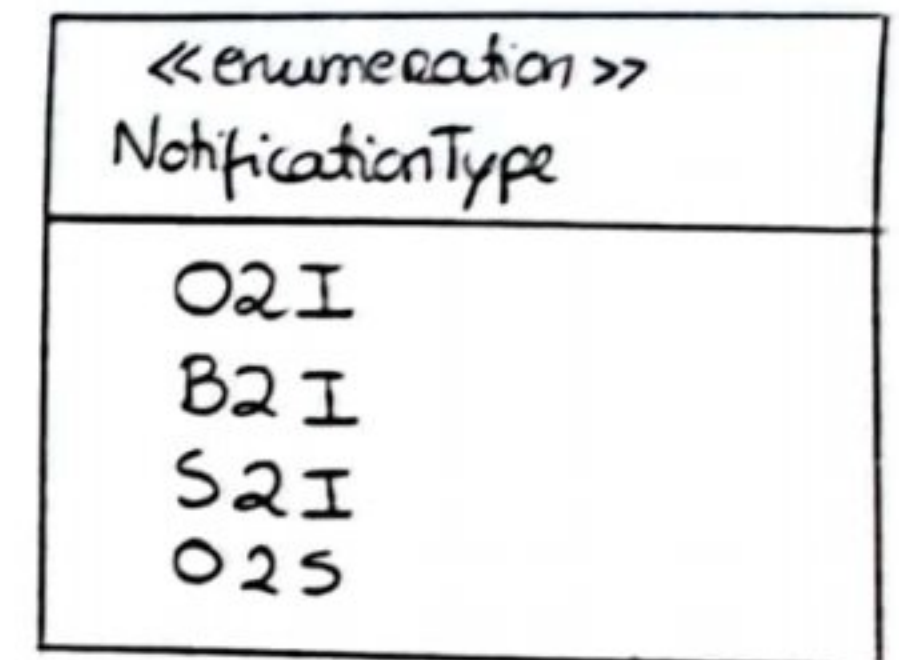
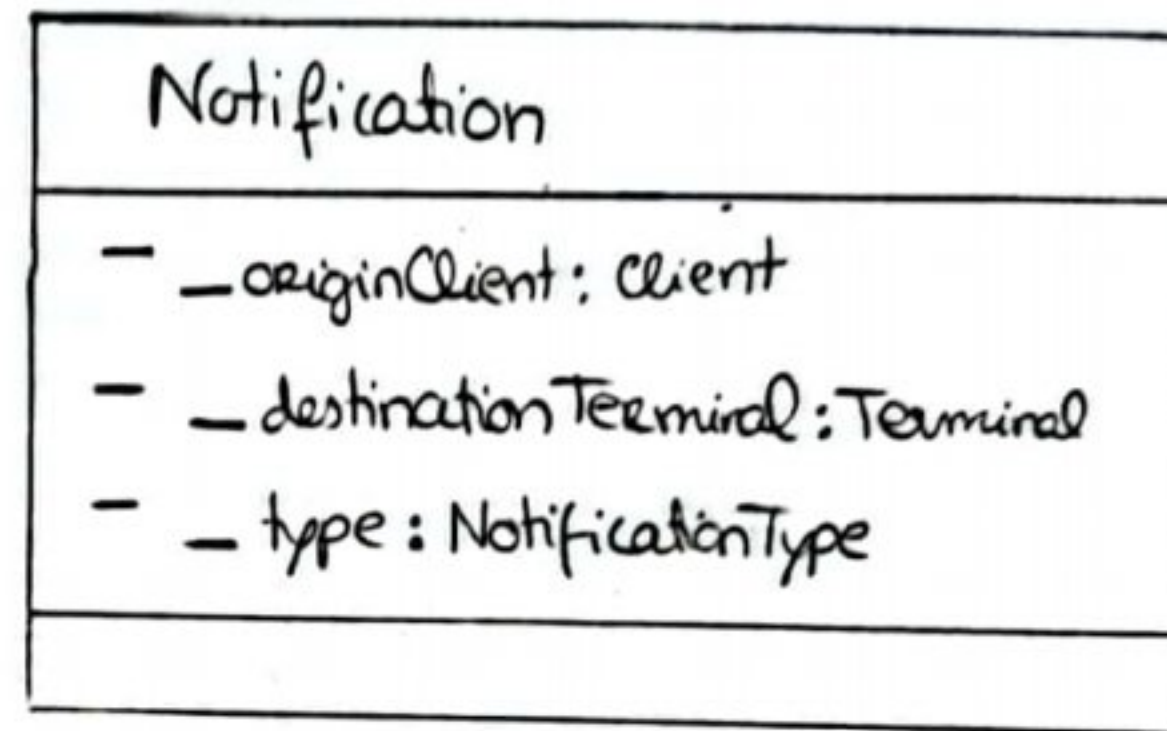
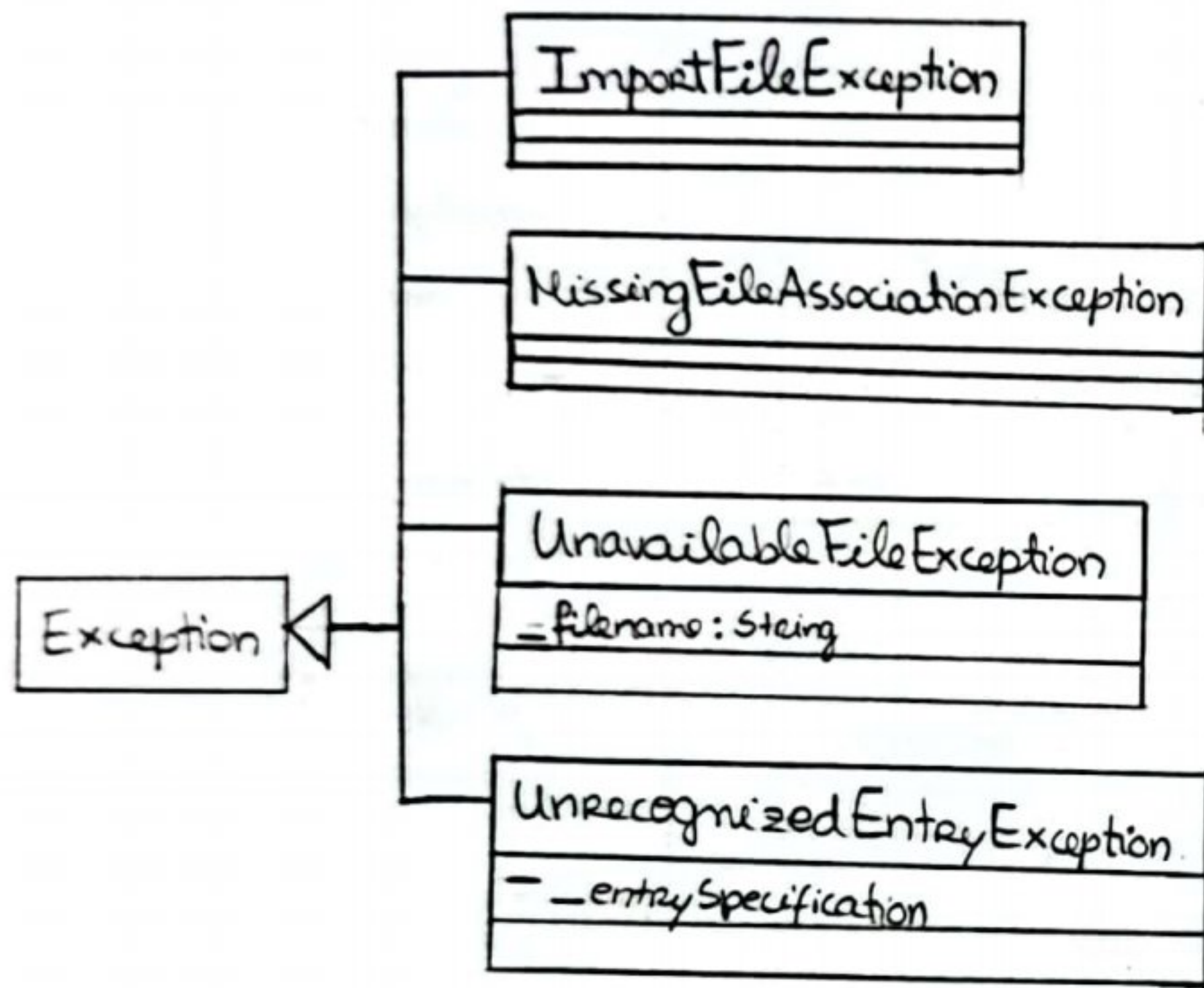


Declaro por minha honra que este diagrama foi realizado apenas pelos elementos que constituem o grupo de projeto Eduardo Norário
Declaro, por minha honra, que este diagrama foi realizado apenas pelos elementos que constituem o grupo de projeto. *Beatriz*
Grupo 025 1st1102415 1st1102463



<<abstract class>> Terminal
<ul style="list-style-type: none"> - owner: Client - key: String - state: State - friends: Collection<Terminal> - pastCommunications: Collection<Communication> - ongoingCommunication: Communication - textNotif: Collection<Notification> - interactiveNotif: Collection<Notification>
<ul style="list-style-type: none"> + <<abstract>> canMessage(): boolean + <<abstract>> canVoiceCommunicate(): boolean + <<abstract>> canVideoCommunicate(): boolean + canEndCurrentCommunication(): boolean + canStartCommunication(): boolean + sendTextCommunication(destinationID: String, msg: String): void + startInteractiveCommunication(destinationID: String, type: Communication): void + addFriend(-id: String): void + isFriend(terminal: Terminal): boolean + turnOnTerminal(): void + turnOffTerminal(): void + endInteractiveCommunication(duration: int): void + performPayment(communicationKey: int): void + removeFriend(id: String): void + showOngoingCommunication(): void + silenceTerminal(): void + switchState(state: State): void + calculateBalance(): int + showTerminalBalance(): void

Basic
<ul style="list-style-type: none"> + canMessage(): boolean + canVoiceCommunicate(): boolean + canVideoCommunicate(): boolean

Fancy
<ul style="list-style-type: none"> + canMessage(): boolean + canVoiceCommunicate(): boolean + canVideoCommunicate(): boolean

Client
<ul style="list-style-type: none"> - key: String - name: String - nif: int - plan: Plan - terminals: Collection<Terminal> - notifications: Collection<Notification>
<ul style="list-style-type: none"> + planTransition(): void + switchPlan(toSwitch: Plan): void + calculateBalance(): int + showClientBalance(): void

«abstract class» Communication
<ul style="list-style-type: none"> - _key: int - _origin: Terminal - _destination: Terminal - _ongoing: boolean - _paid: boolean - _cost: int

«abstract class» InteractiveCommunication
<ul style="list-style-type: none"> - _duration: int

Text
<ul style="list-style-type: none"> - _sentMessage: String

«interface» State
<ul style="list-style-type: none"> + canReceiveTextCommunication(): boolean + canReceiveInteractiveCommunication(): boolean + canStartCommunication(): boolean

Busy
<ul style="list-style-type: none"> + canReceiveTextCommunication(): boolean + canReceiveInteractiveCommunication(): boolean + canStartCommunication(): boolean

Idle
<ul style="list-style-type: none"> + canReceiveTextCommunication(): boolean + canReceiveInteractiveCommunication(): boolean + canStartCommunication(): boolean

Off
<ul style="list-style-type: none"> + canReceiveTextCommunication(): boolean + canReceiveInteractiveCommunication(): boolean + canStartCommunication(): boolean

Silent
<ul style="list-style-type: none"> + canReceiveTextCommunication(): boolean + canReceiveInteractiveCommunication(): boolean + canStartCommunication(): boolean

Network

- _clients: Collection<Client>

importFile(filename: String): void
+ findTerminalById(id: String): Terminal
+ findClientById(id: String): Client
+ showClient(clientId: String): void
+ enableClientNotifications(id: String): void
+ disableClientNotifications(id: String): void
+ registerClient(clientId: String, name: String, nif: int): void
+ showAllClients(): void
+ showClientPaymentsAndDebts(id: String): void
+ showAllCommunications(): void
+ showClientsWithDebts(): void
+ showClientsWithoutDebts(): void
+ showCommunicationsFromClient(id: String): void
+ showTerminalsWithPositiveBalance(): void
+ showUnusedTerminals(): void
+ showGlobalBalance(): void
+ openMenuTerminalConsole(): void
+ registerTerminal(terminalId: String, type: String, clientId: String): void
+ showAllTerminals(): void

Network Manager

- _network: Network

- _filename: String

+ load(filename: String): void
+ save(): void
+ saveAs(filename: String): void
+ importFile(filename: String): void