E-Wallet

A **digital wallet** also known as "**e-Wallet**" refers to an electronic device or online service that allows an individual to make electronic transactions.

You have to write a program that simulates the creation and administration of an e-wallet.
The program must include the declaration of a class „ewallet“

The e-wallet is defined by:
- an alphanumeric code which represents the Client reference
- a numeric code (PIN) which protects the e-Wallet (by default, at the creation of an e-Wallet, the code is a random number between 100 and 99999.
- A value which represents the available money on the e-Wallet
- A value which represents the authorized overdraft on the e-Wallet
- A value which represents the processing fee for every transaction
- The processing fee in % and the authorized overdraft are fixed by the bank, so you can treat them as constant values which are used for all e-Wallets.

An e-Wallet has to the following behavior:
- A method which verifies that the client reference and the PIN (which are parameters of the method) are equal to those of the e-Wallet. The method return „true“ if correct, else „false“

```
def verifieClient(codeclientele : Int codepin : Int) : Bool
// codeclientele contient la valeur à comparer à la valeur du code client•e du
ewallet
// codepin contient la valeur à comparer à la valeur du code PIN du ewallet
```

- A method which allows to debit an amount of the wallet if possible. The method checks if the operation is possible (by comparing the amount to debit with the money available and the authorized overdraft). Attention: You have to calculate the processing fees and include them in your equation of the debit. If the operation is possible, it is executed.

```
def operationDebit(montantadebiter : Double) : Double
// montantadebiter contient la valeur du montant à débiter.
// La méthode renvoie la nouvelle valeur du montant disponible si l'opération a pu
être réalisée. NaN sinon.
```

The main Program has to allow the client the following options: a) Create an e-Wallet,
b) Transfer some amount from one e-Wallet to another. The program will be the interface between the client and e-Wallet.
It includes an Array or an ArrayBuffer of object of the class „ewallet“. This Array or ArrayBuffer stocks all the e-Wallets of the client.

When starting the program, the program has to read a file which contains the list of e-Wallets and insert this into the Array. At the end, the program has to save the list of all e-Wallets from this Array in this file (e.g. csv file).

The program proposes some multiple choice menus, which allow the user to:
-> Create an e-Wallet, Access to his e-Wallet, Quit the program [Those 3 are in the Main Menu]
-> A client who chooses to create an e-Wallet has to provide his client reference. If there already is an e-Wallet with this code, the creation of the e-Wallet has to be refused. (A client can only have one e-Wallet at a time) The program then has to ask the user for the initial amount to credit his e-Wallet. The e-Wallet is then already added to the list of existing e-Wallets. The necessary informations have to be communicated: The PIN and the amount available.
-> If the client chooses to access his existing e-Wallet, he has to enter his client reference and his PIN. Then the program verifies if the e-Wallet exists and if the PIN is correct. The program then proposes to transfer some amount from his e-Wallet to the e-Wallet of some other client. The user has to provide an amount to transfer and the client reference of this client. The program verifies that this e-Wallet of the other client exists & executes the transfer (by using the method operationDebit). If the transfer isn't possible, the program has to announce this and also give a reason. Once a transfer is executed (or not), the user has to be brought back to the Main Menu.

Then the program has to inform the user about whether or not the transactions have been executed or not, what the new amount available is, etc.
If the user decides to do another transaction, he has to start again in the Main Menu, but all transactions are being calculated.

Note: It is clear that during all steps of this program (in the file as well as in the programm), the PIN codes are stocked and visible. You don't need to crypt them to hide them!