VNIVERSITAT DE VALÈNCIA

# STRUCTURES Laboratory

## DATA AND ALGORITHMS

***Degree in Data Science (1st)***

**Academic year 2023-24**

**Practice No. 4: Classes and Sequences**

## Classroom exercise (L1)

Once the class is implemented*Bank account*a program must be written (“**“Pr4_final.py”**) that processes the receipt collection operations issued by an energy supplier company (electricity, gas, etc.) to its customers based on consumption. The program must meet the following requirements:

1. Read the file “accounts_L1.dat”to have a catalog (in list form) of bank accounts. The file is in CSV format, where the information for each account is on the same line separated by “;” (string with IBAN code;holder;available balance). In the list each account will have the data read and no operations recorded. Please note that if an IBAN code is incorrect, the account cannot be created. Check that the catalog is correct according to the content of the file (there are 30 accounts).

2. Read from file “receipts_L1.dat”the details of the receipts that must be collected. This file is in CSV format, where the information for each transfer is on a line separated by “;”. Each line contains:

   - String with IBAN code of the account in which the receipt should be charged.
   - Receipt amount

   The read data must be stored in a queue (each line an element of the queue), which maintains the order of the file. Temporal order is an important factor in banking processes, therefore, it must be strictly respected.

   In the AV there is an implementation of the Queue class.

3. Write a function*CollectReceipts*that allows payments recorded in the queue to be made effective. To do this, this function must have as arguments the queue with the information on the receipts and the catalog of bank accounts and must follow the following scheme:

   - While there are pending receipts in the queue:
     1. Extract the first receipt from the queue and separate its 2 data.
     2. Check that the payment account is in the bank account catalog.
     3. If the account has been found:
        to. If the account balance is greater than or equal to the amount received then perform and record a transactionwithdrawal of money in

the check , for the amount indicated and with the concept "Energy receipt".

b. If the above is not met (insufficient balance), then reject the receipt and record in a list of unpaid receipts due to insufficient balance.

4. If the account is not found in the catalog then reject the receipt and record a list of unpaid receipts due to account error.

This function should return the number of receipts collected correctly, the list of receipts rejected due to lack of balance, and the list of receipts rejected due to incorrect account.

To perform step 2 of the previous algorithm (*"Check that both the account originating the transfer,..."*) it is necessary to look for an IBAN in the chart of accounts and know the position in which it is located (important to do step 3.a.). To do this, you have to do a (sequential) search function on the list. So that the IBAN identifiers (which is a class) compare well, the == operator has been overloaded (__eq__)in the IBAN class. The search function should return the position of the catalog where the bank account that has that IBAN is located or -1 in case it is not found (you have already done this in previous practices).

4. In the main program, the 3 previous tasks must be carried out sequentially, showing the number of receipts successfully paid and the number (the complete list is not necessary) of each of the two types of rejected receipts.

5. Finally, at the end of the process, the program must show the information of each of the accounts included in the catalog to be able to check the available balance and the registered payment operations (it shows each account and pauses on the screen to be able to check the results). Several receipts will have been loaded into each account. It is not an error.

## testing bench

- Receipts collected correctly: 499
- Unpaid receipts due to lack of balance: 498
- Unpaid receipts due to account error: 3

Final results for the first 3 accounts in the catalog:

**Account details: ES79 0933 5560 5283 0308**
**MARIA PASTORA LADRA EZQUERRO**
**Balance: 36.17999999999999 EUR**
== Operations ==
Refund: -30.03 EUR (Energy Receipt)
Refund: -6.52 EUR (Energy Receipt)
Refund: -58.27 EUR (Energy Receipt)

**Account details: ES10 3387 8296 2951 9160 ELOY JESUS REDON MUNTANER**
**Balance: 414.7300000000075 EUR**
== Operations ==
Refund: -669.05 EUR (Energy Receipt)
Refund: -996.46 EUR (Energy Receipt)
Refund: -976.43 EUR (Energy Receipt)
Refund: -150.45 EUR (Energy Receipt)
Refund: -636.03 EUR (Energy Receipt)
Refund: -1662.66 EUR (Energy Receipt)
Refund: -1640.01 EUR (Energy Receipt)
Refund: -2451.92 EUR (Energy Receipt)
Refund: -1587.41 EUR (Energy Receipt)
Refund: -235.6 EUR (Energy Receipt)
Refund: -0.26 EUR (Energy Receipt)
Refund: -924.59 EUR (Energy Receipt)
Refund: -2536.58 EUR (Energy Receipt)
Refund: -1498.22 EUR (Energy Receipt)
Refund: -568.56 EUR (Energy Receipt)
Refund: -583.87 EUR (Energy Receipt)
Refund: -249.55 EUR (Energy Receipt)
Refund: -3809.69 EUR (Energy Receipt)
Refund: -2332.87 EUR (Energy Receipt)
Refund: -108.89 EUR (Energy Receipt)
Refund: -681.17 EUR (Energy receipt)

**Account details: ES65 4104 8338 6121 3913 SAMIRA RITA CAMIRUAGA**
**Balance: 10.53000000000003 EUR**
== Operations ==
Refund: -91.93 EUR (Energy Receipt)
Refund: -1446.77 EUR (Energy Receipt)
Refund: -53.93 EUR (Energy Receipt)
Refund: -7.53 EUR (Energy Receipt)
Refund: -285.35 EUR (Energy Receipt)
Refund: -46.65 EUR (Energy Receipt)
Refund: -528.43 EUR (Energy Receipt)
Refund: -1390.09 EUR (Energy Receipt)
Refund: -257.66 EUR (Energy Receipt)
Refund: -705.13 EUR (Energy Receipt)