PROGRAMMING – PROJECT

Degree in Computer Science: 1st year, 2nd semester *2017/2018*

Guitars For Rent

Introduction:

A store that rents guitars, "Guitars For Rent", wants an application to manage their business. The rental of guitars can only be made by registered customers. After registration, each costumer can rent up to 5 guitars at a time (although store owners may change this value in the future). Each of the guitars available for rent has a specific value. Customers can only start renting expensive guitars (guitars worth more than 500€) when they have already successfully completed the rental of 6 cheap guitars.

Each guitar can be rented for periods of up to one week (7 days), subject to restriction never having more than 5 guitars in their possession. At the time of return, each guitar is inspected. If it is damaged, a fine equal to the value of the guitar is charged. If a customer damages more than 3 guitars, membership is revoked, and this customer is blacklisted. From this point onwards, that customer cannot rent more guitars or become a member again.

If a customer does not return a guitar in time, a 10€ fine will be charged for each day of delay. If a customer accumulates more than 20 days late delivery of guitars, their membership status is also revoked and blacklisted.

The store wants to keep a complete record of guitars and customers. It also intends to maintain an updated list of customers that have been banned for breaching the rules.

Information to Process:

The program must store and manage the following information:

- The stock of guitars that the store has
- Customers registered in the store and their rentals (present and past)
- Banned customers

Minimum features of each guitar:

- Unique ID
- Name
- Price per day of rental
- State (0 – available, 1 – rented, 2 – damaged)

Minimum characteristics of each customer:

- Name
- Unique ID
- List of rentals, where the characteristics of each rental are:
    - Guitar ID

- Start date
- Delivery date
- Rental status (0 – in progress, 1 – delivered, 2 – delivered with damage)

Files:

When the program is not running, information about active customers and guitars must be stored in 2 text files. Each line of the guitar file must have the following structure:

ID	DailyPrice	Value	State	Name

The file of active customers must have the following organization:

Customer1		ID	RentalsNumber	Name					
	Rental1	GuitarID	State	dayI	monthI	yearI	dayD	monthD	yearD
		…							
	RentalN	GuitarID	State	dayI	monthI	yearI	dayD	monthD	yearD
		…							
CustomerY		ID	RentalsNumber	Name					
	Rental1	GuitarID	State	dayI	monthI	yearI	dayD	monthD	yearD
		…							
	RentalK	GuitarID	State	dayI	monthI	yearI	dayD	monthD	yearD

In other words, each customer must have their information organized as follows:

- First line with: ID, number of rentals and name
- Next there is a line for each rental, where each of these lines contains: Guitar ID, status of the rental, day, month and year of the start date of the rental, day, month and year of the delivery.

The information of banned customers is stored in a binary file. When a customer is banned, their information is removed from the active customers section and passed to the binary file. It is at the discretion of the student to choose the organization of this file.

In-Memory Organization:

During the execution of the program, all information from active customers and guitars must be transferred to memory, dynamically managed. At the beginning of the execution, this information is transferred from the text files to memory. All operations described in the Functionalities section (see below) are performed on the data that is in memory. Immediately before the execution is finished, the text files are updated. The dynamic memory management must obey 2 basic requirements:

- The information of the guitars must be stored in a dynamic vector
- The information of the active customers and the respective rentals must be kept in a dynamic structure of the linked list with, at least, 2 levels (for example, a list of lists).

The information of banned customers should be kept in the binary file. Whenever a customer is banned, this file must be immediately updated.

Program Functionalities:

The program must have the following functionalities implemented:

- Guitars:
  - Add a guitar to the store's stock
  - Show a guitar rental history (customer, start and end dates of the rental and days of delivery delay)
  - List all guitars (name, state, ID, price per day and value)
  - List rented guitars (name, ID, price per day and value, name and customer's ID)
- Customers:
  - Add a new customer. The program should check if the customer already exists as an active or banned customer, not allowing its reintroduction
  - Remove a customer (it should not be possible to remove a banned customer)
  - Show the status of a given customer (how many guitars the customer has, how many rentals the customer made in total , how many deliveries completed with delay, how many guitars delivered damaged)
  - List all active customers
  - List banned customers (name and ID), indicating the reasons for which they were blacklisted (delay or damaged guitars)
  - Banish a customer: this is an automatic feature that results from breach of the rules. It should not be the user to explicitly ban a customer, but the program to find out if a given customer should be banned. It must be checked in the delivery or in the attempt of a new rental. The passage of a customer to the blacklist implies the return of guitars that may still have been rented. These guitars are available for new rentals.
- Rentals:
  - Create a rental for a given customer and guitar, showing the deadline for delivery and the maximum amount foreseen for the total to be charged.
  - Conclude a rental (guitar delivered by the customer), updating the corresponding guitar rental status, and showing the amount to pay
  - Produce a list of rentals taking place at that time, indicating for each: customer, guitar, start date, expected delivery date and any days of delay in delivery that are already occurring

For all operations, the identifiers to be used are the guitar ID and the customer's ID. The program does not automatically manage dates (it does not use the system date). The user must indicate the date (day, month, year) in the various operations where this is necessary.