Programming
Academic Year: 2021/22

Practical Case

Phase 2: Functions and Data Validation

Content

1 Introduction

This document contains the statement of the Practical Case: Phase 2, in relation to the topics: functions, data validation and the use of iterative statements. This must be understood as a continuation of the statement for Phase 1, meaning that all information from the statement from Phase 1 is valid for this phase.

1.1 Evaluation

Phase 2 scores for **10% of the total grade** of this course.
The teacher will evaluate the deliverable and publish the grades at the end of the semester.
The students can request a revision of the grade after its publication.

1.2 Delivery

The deliverable must be uploaded to Aula Global before the end of **November 14th**.

The deliverable consists of **a single compressed file** (.zip or .rar) with the following documents:

- **Report (2%)**: a report of the practical case in **PDF format**, including the sections described below. **Important:** it is not necessary to include source code in the report, although students can include some code lines in order to explain a fragment of the project itself.
  - Cover: phase of the practical case, students' name, group and course number.
  - Introduction: brief description of the work, adding any specific comments about the practical case.
  - Implementation: for each task to implement, include validation tests to verify that the program always executes correctly (including exceptional cases).
  - Conclusions: about the completeness of the solution provided, as well as comments about the practical case. Students may include personal comments about the current phase, difficulties and problems encountered during the implementation.
- **Source code (8%)**: one or more .py files with the source code that implements the practical case. The code must include program execution examples to confirm the validation tests, including exceptional cases.
  - Note: it is recommended to store all .py files in the same project folder in order to facilitate the delivery of the practical case. That project must be compressed into a .zip or .rar file and uploaded to Aula Global.

1.3 Changes with respect to Phase 1

Unlike Phase 1, in this phase it is mandatory to use loop statements and functions. It is not allowed the use of lists, vectors or any other complex data type.

2 Phase 2 Statement

Phase 2 adds three main functionalities to the program delivered on Phase 1: iteration in the execution of the program, data validation and use of functions.

2.1 Tasks to implement

The current document shows the tasks to be implemented on Phase 2 of the project, built on top of the previous deliverable. The tasks to implement are the following:

2.1.1 TASK 1: Iteration

Continue using the menu created on Phase 1, as presented on Figure 1.

```
************ MAIN MENU ************
   1. New Lifting Equipment
   2. New client
   3. New Equipment Rental
   4. Exit

Please, enter an option (1-4):
```

Figure 1. Displaying the main menu

However, when the user enters option 1, 2 or 3, the corresponding operations must be executed and the menu **must be displayed again**.
In other words, instead of the program ending after entering any of the options, the program will end only when **option 4 is selected**.

Figure 2 depicts a flow chart diagram of the program for a better understanding of this task.

2.1.2 TASK 2: Data validation

It is recommended to read the TASK 3 description in the following section before implementing this task, given that both tasks can be performed for options 2 and 3.
On Phase 1 there is a data verification for the Brand of the equipment, which can only be one of three values. If data validation is not implemented, it can cause data inconsistency on the equipment information.
Another example can be when a user is asked to enter their DNI (identification number); if the user enters `12345678L` a first time, `12.345.678-L` the second time, and `12345678-l` on the third time, and the program does not perform any data validation, the system will consider the same person as three different customers.

Similarly, if there's not a data validation implemented, the user can make a mistake and enter an incorrect value like `123L3123L`, which would not be a valid DNI.

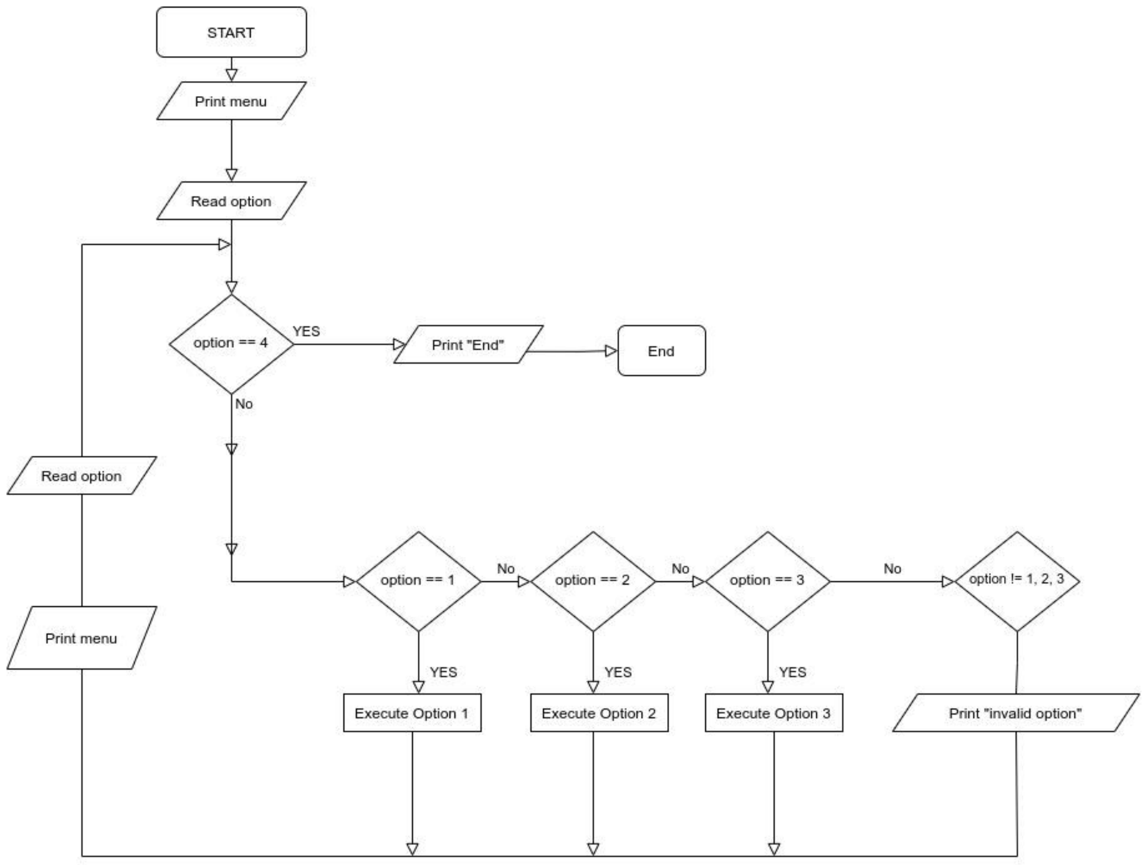

Figure 2. Flow Chart Diagram for the menu

In order to avoid these situations, two tasks must be implemented: normalization of some of the data (like the Brand data on Phase 1); and implementation of the instructions that validate the following:

- The brand must be one of the three previously defined: POLFINGER, KONEG y SMALZ
- The ID of the equipment must be an integer number that, when incorporating a new lifting equipment (option 1 of the menu), the program automatically assigns the next value to the one used for the last equipment (i.e. incremented by 1). Initially, this value will be 1, the second product will have an ID of 2, etc.
- Moreover, it is important to verify that when the user is asked to enter a number (menu option, prices, etc.), the user can only enter numbers (integer or float)[1].

When creating new clients, the data to be entered by the user is the following:

- First Name
- Last Name
- DNI: it must be verified that the DNI of the client is valid and that is written in the common format (NNNNNNNNL) without leading zeros (on the left) for cases with less than 8 digits[2].

[1] In order to request an integer number to the user, an initial source code can be found in the "initial code" of this phase (uploaded to Aula Global), on a function called *enter_integer* that requests data from the user until a valid integer number is entered. The function returns an integer number. This function can be used at any point of the project. Other similar functions may be implemented for other data types to verify, e.g. floating point numbers.

[2] To determine if a DNI is valid or not, a function in the initial code can be found with the name *check_dni* that, given a string, returns a boolean value (True or False) if the DNI entered is valid or not.

- Address
- Phone number: all phone numbers must be stored as integer numbers of 9 digits.

```
Please, enter an option (1-4): 2
NEW CLIENT

Enter the first name of the new client: Pepe

Enter the last name of the new client: Fernandez

Enter the DNI of the new client using the format 'NNNNNNNNL': 6273514H
6273514H is not a valid DNI

Enter the DNI of the new client using the format 'NNNNNNNNL':
31734403S

Enter the address of the new client: Calle Mayor, 1, Madrid

Enter the phone of the new client: 913526271

### CLIENT DATA ###
Name of the client: Pepe Fernandez
DNI: 31734403S
Address of the client: Calle Mayor, 1, Madrid
Phone number of the client: 913526271
```

Figure 3. Entering a new client

2.1.3 TASK 3: Functions

As mentioned in the introduction, an important change on Phase 2 is that functions are allowed to be implemented in order to complete the program logic. In Phase 2 some execution tests of the program can be performed with provisional data, until Phase 3 where the data structure will be implemented.

In this task, it is required to implement **at least** the following functions:

- A function that displays the menu and returns the number entered by the user (only valid options).
- A function that executes option 1 (add new lifting equipment). Request the data to the user and validate the data inside the function. It returns the set of data from the new product.
- A function that executes option 2 from the menu (new client). This function must check that the DNI is valid by using the function described above. It returns all of the data from the client as a set.
- A function that requests the user for a floating point number until a valid number is entered.
- A function that displays, based on the execution example from Figure 4, all the data from a lifting equipment; and another function that displays, as depicted on Figure 5, all the information from a given client.

```
*** LIFTING EQUIPMENT ***
ID: 512
Name: Polfinger Maxi Lifter
Brand: POLFINGER
Model: PCC 115.002
Type: All-terrain crane
Rental price (without VAT): 12500
Rental price (with VAT): 14500
Units available: 4
```

Figure 4. Display data for a lifting equipment

```
*** CLIENT INFORMATION ***
First name of the client: Pepe
Last name of the client: Fernandez
DNI of the client: 31734403S
Address of the client: Calle Mayor, 1, Madrid
Phone number of the client: 913526271
```

Figure 5. Display Client information

2.2 Important: Exclusions

Under any circumstance it is allowed to:

- use global variables or constants.
- use external libraries or objects, including complex data types or lists, vectors, dictionaries, sequences, among others.