I2 *Algorithmique et programmation structurée*
Mars 2019

TD n°6	**Top-down analysis**

Objective of the session :
Divide a relatively simple program into subprograms. Use of files.

1 Account statement

Consider text files containing a set of banking operations in the form :
1735 DRFIP salary
-632 rent
-67.32 gasoline
⋮

Write a program that allows to :
- create a new account (and therefore the associated text file, checking if it already exists) ;
- view an account statement ;
- enter a new transaction ;
- display the balance of an account ;
- Exit ;

The statement view, a transaction entry and the balance display can be done on any existing account.

Remarks :
- Check if the file already exists using the FileExists(filename :String) :Boolean function which returns True if the file with the name given as parameter already exists.

1.1 Global design

From the specifications given above, perform the global design of the program according to the structured programming paradigm, by defining a top-down analysis and then the signatures of the functions and procedures to be written.

To perform a top-down analysis, follow the four steps below :

Exercices

1. **Informally define the tasks to be performed by the program, and possibly sub-tasks within each task.**
2. **Write this functional decomposition in the form of a top-down analysis diagram.**
3. **Identify the inputs and outputs of each subprogram, defining if it is necessary your own types.**
4. **Choose to represent each subprogram with a function and write its signature.**

1.2 Detail design

Perform the detailed design of each function and procedure for which you have established the signature, as well as for the main program.

1.3 Implementation (TD7)

Exercices

1. **Implement the previously defined program.**
2. **Modify the transaction addition functionality so that the user can add one or more transactions.**
3. **Add a functionality to delete transactions.**