Designing Intelligent Technology Applications using the Visual Basic Files

Dr. Dan Mircea Trana[1]

[1] Spiru Haret University

Received: 10 December 2014 Accepted: 1 January 2015 Published: 15 January 2015

Abstract

This paper proposes an overview of the approach to design management computer applications using advanced programming languages and having the Visual Basic demonstration as support, which is one of the most popular object-oriented programming languages, in the area of programmer training. After a brief introduction to the Visual Basic syntax instructions for working with files, the description of how to build applications out of school activities for students - application that is set up as an illustration of features offered by the Visual Basic in working with collections of independent data - the files.

Index terms— file, open, close, sequential, random, binary, line input, print, get, put.

1 Introduction

omputer programming for the development of management applications can be achieved through advanced programming languages, the most widely currently used being the object oriented ones. One of the most recommended in this category, especially in terms of teaching is Visual Basic. If we relate to computer management applications, the huge amount of data they manipulate leads to the necessity of working with collections of data, especially files, usually regarded as independent, their relationship being done just by the help of the computer programs that exploit them.

Visual Basic has several peculiarities in the use of data files, which is why we approached the present study, providing the following aspects to the reader: the concept of files in Visual Basic, general selfdocumenting format of instructions that allow translation into programming language operations with files and obviously the needed illustration. For the latter approach we chose an example that has a strong teaching purpose, the problem solved being intentionally simplified by proposing a Visual Basic project.

We describe the operations in detail, for the programmer to enable any reader, that has minimal knowledge of programming in Visual Basic, to achieve the proposed project on their own, including the completion of its functions with code-programs, that are missing from the material.

The material is on the one hand, a systematization of knowledge that is necessary for the development of projects using Visual Basic files and, on the other hand, the setting out of the proper conditions for training the programming skills in this area, even for amateur or novice programmers.

2 II. Review of the Paper on the Subject Matter

The specialized literature in programming in Visual Basic is very generous lately. The current trend in the development of documentation for help in programmer training is to provide the computer user, willing to make a major step towards becoming a programmer, a self-documented documentation and accompanied, on each major section, by examples that enhance the attractiveness of the reader.

We considered that such work is welcome, and we wanted it to be an effective means of learning and, at the same time, a model for the development of the documentation for current or prospective programmers.

3 III.

4 Theoretical Basis

In the development of the present material, we approached the theoretical and practical issues, having the documentation connected to the bibliography in Paragraphs 1 and especiallly 2, as a starting point. a) Working with files in VB [2] The large amount of the data to be processed requires the storage of the data in collections that have records of a well defined structure. As we mentioned before these are called files. Like other advanced programming languages, VB offers the possibility of processing the data stored in files.

To enable the writing or reading of information to /from files and to protect the data stored in files, they must undergo some opening, respectively closing tasks, before or after their "use". For the description of the tasks as open, close, read or write of information stored in files, in the VB programs, the program has specific instructions defined. i. Open a file [2] Prior to input / output operations on the data stored in files, VB requires the opening of the file. This process is described in VB by the Open instruction, whose general syntax is the following one:

Open file specifier For mode **??** To open multiple files simultaneously, it is recommended that the number of work area attached to the file, to contain the value returned by Free Find (this function returns the next number of a free zone in which the file can be opened). To determine the record length of a file RecLength function can be used.

Open instruction can be used only for opening files from VB applications, not other applications such as Word, Excel or Access.

ii. Closing a file [2] To ensure the security of data stored in files, after their use, these should be subject to the closing operation. The shutdown operation of files in VB is achieved by Close instruction having the follwing general syntax:

Close [list of file numbers] Close instruction without parameter has the effect of closing all files that are active at the time. For files opened with Append or Output, all data in the buffer area associated with the file are transferred before closing. Closing a file results in ending the relation between the file and the associated buffer zone, which is thus released.

iii. Accessing a file [1] [2] Files can be accessed through VB programs in several ways: ¬Sequentially ¬Randomly ¬Binarily processing data stored in files on several lines of different lengths. To accomplish this way of using VB, access instructions / O Line Input # and Print # are used. They have the following general format: Line Input # file number, variable name Print # file number, the list of output variables Line Input instruction has the effect of reading a record from the file indicated by file number and storing the data read in the internal memory (buffer) associated to the variable whose name appears as a parameter of the instruction. Following the execution of this instruction on an open file with the Open instruction, the data are read from the file, character by character, to meet the CRLF code sequence (Carriage Return Line Fide), which marks the end of a line in the file, and are transferred to the buffer are associated to the file.

Print instruction allows indication of the the operation of VB program writes data in a file identified by the file number currently the command parameter. List of output variables contains variables, arithmetic expressions or strings that are intended to be stored on a new line of the file. If this parameter is missing, the file will be written a blank line. Random access (Random) is used when data records are structured the same length and the same structure in fields. In this case access to the data stored in the file is done directly on the registration sought. To ensure that the description of the input / output files opened in Random mode have been designed in VB following instructions: Get # file number [number of article] variable name Put # file number, [number of article] variable name Get instruction allows reading from the file identified by the file number of the article with the number indicated as a parameter and storing it in a variable whose name is mentioned.

Put instruction allows data transfer from the variable indicated in an article name of the file that was specified via the file number.

Next we consider an issue that requires the use of files to solve it in VB: the record of student academic situation of a university regarding the computer science subjects. Each student must store the following information: registration number, first name second name, faculty, group and year of study, grade connected to practical examination1, grade connected to practical examination 2, final grade for each semester.

5 Fig. 1 : VB project structure

We associated two forms and one module to the application as shown in Fig. **??** In Module1 (stored on the hard disk in file articol.bas) we described the structure of an article in the student file. It is created in the Random way to store information regarding a student as required by the issue. The program code associated with the article module is the following one: Fi?iere.vbp project consists of fi?iere.frm and afi?are.frm forms and articol.bas way, whose content was previously presented. We remind you that if the first form is automatically created a project, for the introduction of other forms in the project we proceed as follows: click the mouse Right on Forms from Project Explorer and access the Add function. You can then select Form and a new form in the project is included.

Form1 form of the project will have the interface shown in Fig. 2 . We can notice the presence of a label type controller which has the task to enter a title for the window managing the project, of 5 control buttons for the implementation of the project functions and a text casset type controller for the transmition of the parameters necessary to Form2 form. s.grupa = InputBox(”Grupa si anul de studiu” Group and year of study) s.notap1_1 = InputBox(”Nota la prima proba practica sem. I” Grade for the first practical examination -1st semester)

s.notap2_1 = InputBox(”Nota la a doua proba practica sem. I” Grade for the second practical examination -1st semester)

s.notas_1 = InputBox(”Nota la proba scrisa sem. I” Grade for the written examination -1st semester)

s.notap_2 = InputBox(”Nota la a doua proba practica sem. II” Grade for the second practical examination -2 nd semester)

s.notas_2 = InputBox(”Nota la proba scrisa sem. II” Grade for the written examination -2nd semester) Put # End Sub It should be mentioned that we only developed the software codes associated to the first three of them and to the last of the control buttons. The other two (changing or adding a new record) were left as an exercise for the reader.

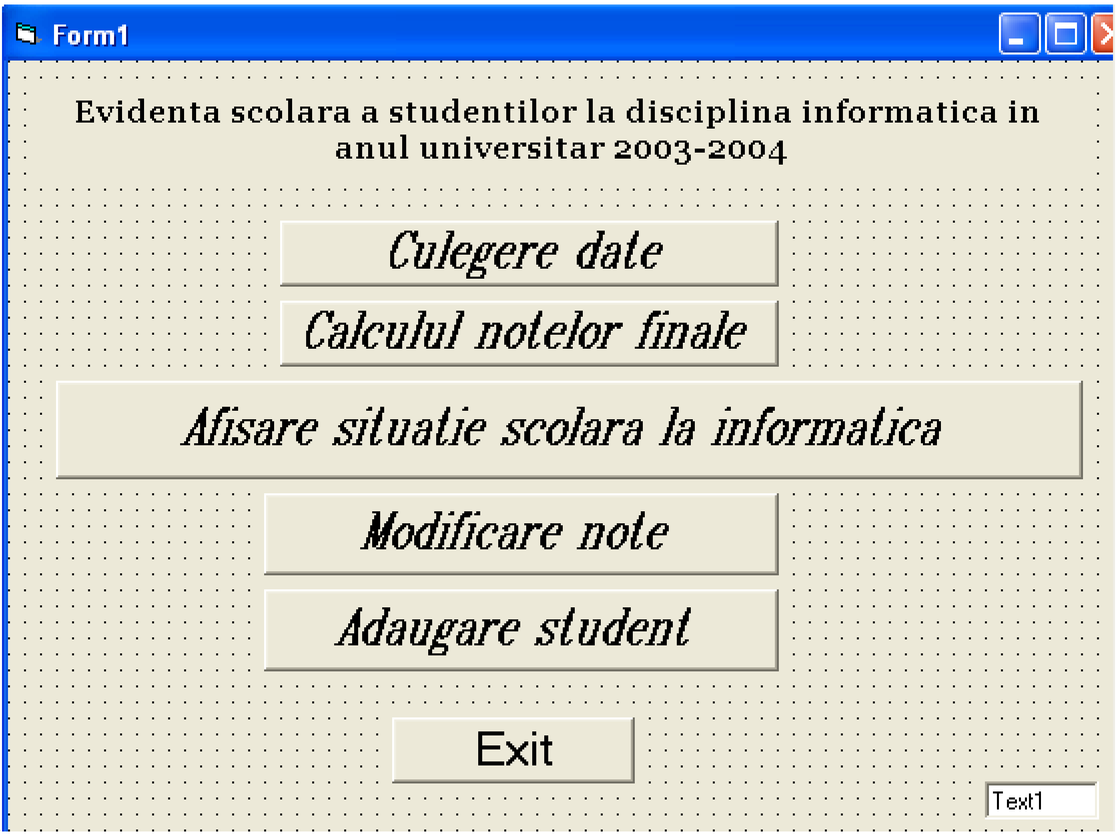
Form2 form meets only the function to ensure the school situation display for each and every student. Data are displayed on the screen in the form afisare.frm interface whose interface is shown in FIG. **??** The program code contains the receiving of the number of students for which display is wanted, the opening of the file and the displaying of the student academic situation for all students, beginning with the second one. At the end of the listing process, the next button is disabled (the last instruction of the code).

The program code associated to the ies button has the purpose to revert to the first form and is given as follows: Private Sub Command2_Click() Form2.Hide Form1.Show End Sub

We leave the designing of a program code associated to the command buttons Change student grades and Add student in window shown in Fig. 2, as an exercise to the reader. [1]

2

Figure 1: Fig. 2 :



Evidenta scolara a studentilor la disciplina informatica in anul universitar 2003-2004

31

Figure 2: Fig. 3 : 1 20

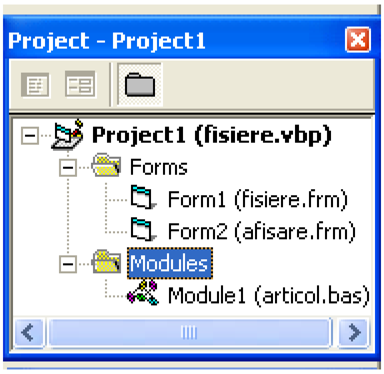


Figure 3:

The significance of the parameters that appear
in the command syntax is as follows:
? file specifier: specifies the file to be submitted to
the opening operation. Let us remember that by file
specifier we understand: [<username> disc>]
[<path of
Access] access
[lock] As [#] file number [Len = length article]

[Note: Thus for the Input or Random modes, we can open a file without closing a copy of the attached work elsewhere, namely another file number. For the Append and Output modes, opening the file can be done only after closing the copy which is already opened into another area.]

Figure 4:

[Tutorial ()] , *** Tutorial . 2010. Basic. 19. (desc?rcat gratuit la adresa(chapter 19)

[Designing Intelligent Technology Applications using the Visual Basic Files] *Designing Intelligent Technology Applications using the Visual Basic Files*,

[Mircea ()] Dan Mircea , Tran? . *Radu Dan Tran?: Informatic? economic?*, 2008. 2.