CLIENT-SERVER APPLICATION FOR TESTING AND TEACHING MODULAR WRITING OF SOFTWARE PROGRAMS

Alexandru ENE, Ioana-Cristina ROBEA
FECC, University of Pitesti, Romania
alexandru.ene@upit.ro, oanarobea@yahoo.com

Keywords: educational software, C functions, client-server application, dynamically generated tests

Abstract: *Today, in learning process, in addition to traditional teaching methods, educational software is increasingly used. This paper presents a Java application that allows the user to learn better and faster, modular programming in a specific programming language (C language). The application is intended to be used in the teaching process (university or high school). Unlike other such applications, test questions are generated dynamically, at the beginning of a new test.*

1. INTRODUCTION

Present day projects in engineering and science are too large and complex to implement as single large programs. The problem should be divided into smaller problems along logical lines and the smaller problems analyzed in order to understand the big picture of the entire problem. In the development of very large programs, which can vary from a few thousand to millions of lines of code, development can be undertaken by teams of programmers, with each team working with a defined subgroup of the functions that make up the whole program. The project can be easily managed if it is divided into specific tasks and further divided into subtasks that can be easily understood and programmed. Tasks and subtasks are defined as the functional units of the program. Modular programs have the following advantages:

- Smaller modules are easier to understand and maintain.
- Separate modules can be tested independently before the entire program is written and executed.
- Different programmers can write different modules.
- Modules can be modified or replaced by new modules to tune the program to the needs of a new application without changing the rest of the program.
- Repeated code can be replaced by a function call.
- A modularized program is simpler and easier to read, understand, and maintain because each module only does one specific task.
- Independent modules are modules that communicate with the rest of the program only through data that is explicitly passed to them or returned from them. Independent modules are easier to debug. [1]

E-Learning domain directly resulted from the development of communication techniques and started its popularity since 1980'. With the increase performance of computing equipment, education systems tend to impose themselves as indispensable tools for a modern education. E-Learning is an extremely effective way to implement educational programs for both higher education and for other forms of education, because it adapts students needs, but also for adults who want to train in areas and technologies, in the lack of time, and different capacity of assimilation.

It can be self paced or instructor led and includes media in the form of text, image, animation, streaming video and audio. [2]

E-learning applications and processes include Web-based learning, computer-based

learning, virtual classrooms and digital collaboration. Content is delivered via the Internet, intranet/extranet, audio or video tape, satellite TV, and CD-ROM.

Electronic education covers a broad category of applications and processes. Acronyms like CBT (*Computer-Based Training*), IBT (*Internet-Based Training*) or WBT (*Web-Based Training*) have been used as synonyms to e-Learning. [3]

2. APPLICATION DESCRIPTION AND FUNCTIONALITY

The main objective of this application is testing and teaching modular writing programs in the C programming language. The application aims to test the user's ability to declare the header of a C function correctly.

The application was developed in Java, using the multiple facilities provided by it, such as network programming using sockets. Socket classes are used to represent the connection between a client program and a server program, the java.net package providing two classes: Socket and ServerSocket that implement the client side of the connection and the server side of the connection, respectively.[4]

Other facilities are: the possibility to achieve a friendly graphical user interface, using well known interactive graphic elements (pushbuttons, edit panes or text panes, comboboxes), processing XML documents used in this application to store data input (authentication data: name, password, year of study, group, teacher, number of questions of the test), but also to save the outputs of the test (test date, the number of correct answers of the test, eventually a score), the possibility of calling system processes (notepad.exe) for learning.

When a client connects to the server, test questions are generated by the server, it uses random values for the elements contained in the header of a C function, namely: the type of return value after calling this function, the function name and list of parameters.

Each user can create an account at the beginning of the application, based on which the results obtained after testing for each user are recorded. The connection at the test is based on the name (first name and second name) and password. Authentication data and results are stored by the server application in an XML document.

The test may contain a variable number of questions, which is selected by the user. The user go through the test in both directions ("Next" and "Back") and he can change any of the previously answers.

When the user ends the test, his answers are collected and submitted to the server application which will transmit the feedback represented by the set of the correct answers. The user can browse the test again, but this time to make a comparison of given answers against the correct ones received from the server. On the basis of wrong answers given by the user the server establishes the user's weaknesses, and guides him to go through a document to help him acquire new knowledge, specifically in the area which has proved to have difficulties.

For both, server and client application, is necessary that the virtual machine Java to be installed, using the standard package provided by Sun, and can be downloaded free from its official website.

The server application is responsible for receiving, verification and validation of user authentication data (data that are stored in an XML file: "utilizatori.xml"), but also for generating and correcting tests for each user. On the other side, the client application provides the necessary graphical user interface.

At the beginning of the application the user has to create an account, by entering the necessary data in the edit boxes with suggestive labels, for each field separately, as shown in Figure 1.

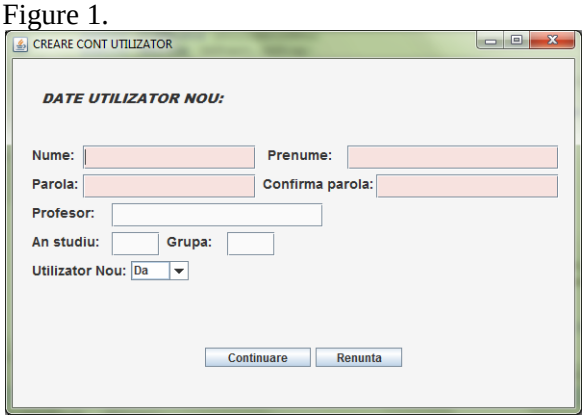

Fig.1 New account window

If the user has already an account, he can select the option "Nu" of the list labeled

"Utilizator nou" and click the button "Continuare".

The graphic window called by pressing the button “Continuare” is shown in Figure 2, and it is used to introduce the data necessary for the user login to the test (full name and password). Besides these data it is needed the selection of the number of questions in the test, using the list labeled "Numar intrebari test”.

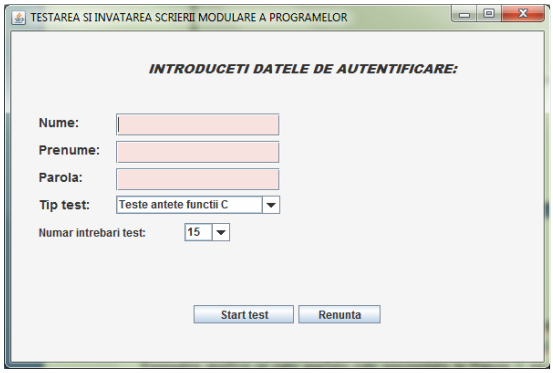

Fig.2 Login window

To start the test itself it is necessary to click "Start Test" button. At this point the client aplication sends the command to start the test. If the login data of current user are correct, the server will generate the test questions and transmits them to client application. If login informations aren’t correct, it will generate a warning message and the user will have to reintroduce them.

Fig.3 Effect of wrong login information

Test questions will be displayed on the surface of a graphic window, with the structure shown in Figure 4. On each window will be displayed three questions, and the user will enter the answers in the edit text boxes corresponding to each question.

To complete the test the user will have to click "Rezultat", and test results are saved in the file "utilizatori.xml" in appropriate fields, specific for current user.

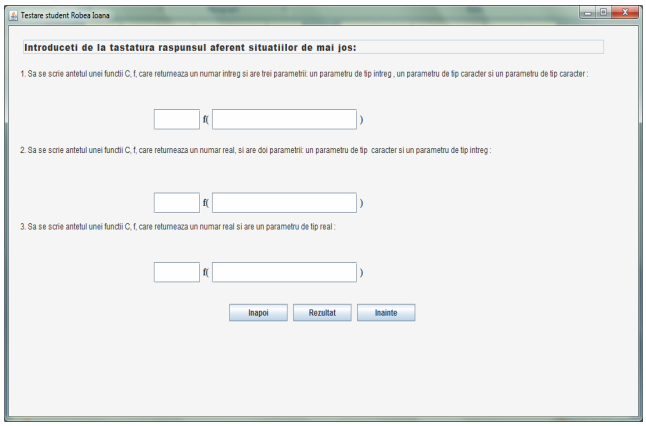

Fig.4 Test window

At the end of the test the user has the possibility to compare his responses against the correct ones (correct answers are marked with green and the wrong ones with red).

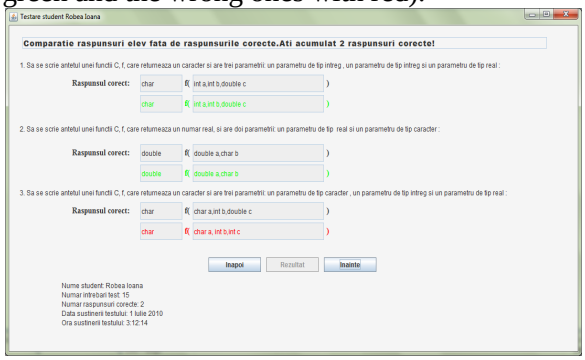

Fig.5 Comparison of results

Also, is generated another graphic window, as you can see in Figure 6, which allows the user to take a new test by choosing the number of the questions of new test and pressing the button "Test Nou", or whether to leave the application by pressing the button "Iesire".

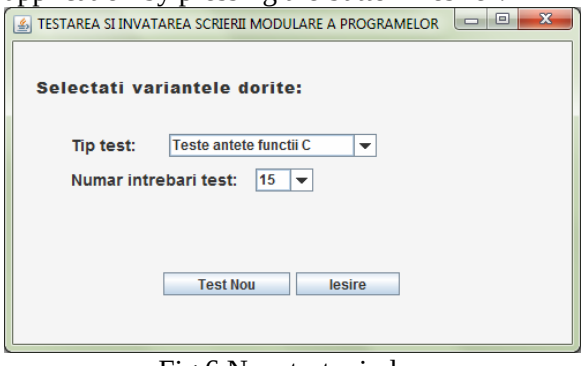

Fig.6 New test window

If the number of wrong answers is greater than 25% of the total number of questions, the server sends to the client a help text material containing information about modular programing in C. The server makes an intelligent selection of the needed help file, based on the statistic of the wrong answers. The help material will be presented by calling the system process notepad.exe.

3. THE ALGORITHM FOR THE GENERATION OF TEST QUESTIONS

As it was said, the test questions are constructed individually based on the random numbers generation. The application aims to test the correctness of the formulation of C functions headers for various situations considered. It is known that the header of a C function has the following structure:

type_of_returned_value function_name (list of formal parameters)

The name of the function (function_name) is generically established to **f**.

Let N be the total number of possible, distinct, returned values of function (void - if the function does not return any value, or int, double and char,etc. if the function returns an integer, a real number or respectively a character,etc.). It is generated a random number in the field [0.N-1] which has the following meaning: void - if the generated random number is 0, int - if the generated number is 1, double - if it is generated 2, and char – if it is generated 3,etc..

To establish the list of parameters it is necessary to first generate the number of function parameters and then their type. In the present application we considered the maximum number of parameters to be three (this could be easily modified). For this it is generated a random number in the field [1,3], wich will represent the number of function parameters, and then in a similar manner used in generating the type of returned value of function, are generated the parameters type. After obtaining all values involved in formulating the question, the algorithm makes sure that in the same combination of values will be generated only once in the current test.

4. CONCLUSIONS

In this paper we presented educational software, which can be considered to be part of the category of e-learning applications. This software is used for testing and accumulation of knowledge in the field of modular programming.

The client-server application is based on the network programming in java using sockets, and uses the Java GUI (Graphic User Interface) programming to achieve the graphical interface. The questions of the test are different for each test, and they are randomly constructed based on random number generation.

There is also an intelligent selection of a help file, based on the statistic of the wrong answers that will help the user to acquire new knowledge in the area in which he had encountered more difficulties.

5. REFERENCES:

[1]. De Rama N. Reddy, Carol A. Ziegler, C “Programming for Scientists and Engineers Applications”, p.266
[2]. Silviu Craciunas, Irina Ersek, “Modelul standard al unei platforme e-Learning” http://depmath.ulbsibiu.ro/chair2/craciunas/model%20standard%20platforma%20eLearning.pdf
[3]. Wikipedia, the free encyclopedia, “E-learning”, accessed: http://en.wikipedia.org/wiki/E-learning
[4]. Stefan Tanasa, Cristian Olaru, Stefan Andrei, “Java de la 0 la expert”, Editura Polirom, 2003