TEACHING SUBROUTINES: AS EARLY AS POSSIBLE*

Mariana Krushkova, Yordanka Stoynova, Hristo Krushkov

Abstract. *Students on introductory courses in programming languages often experience difficulty understanding the basic principles of procedural programming. In this paper we discuss the importance of early understanding of the subroutine mechanism. Two approaches for self-training – static and dynamic - are presented and compared. The static approach is appropriate for written text in a paper textbook. The dynamic approach is suitable for interactive training using a computer. An interactive module was developed for teaching subroutines.*

Keywords: Teaching methods and classroom techniques, Programming languages, Subroutines in procedural programming languages

2010 Mathematics Subject Classification: 97-00, 97Q60, 97P40

1. Introduction

Improving the Bulgarian higher educational standards is one of the strategic targets of our country as an equal member of the European family. The implementation of new scientific methods and modern technological means in the process of education is a precondition for achieving this purpose. The teaching of informatics, one of the core disciplines, makes a significant contribution to the creation of knowledge-based economy. This is why several departments of Plovdiv University „Paisii Hilendarski“ offer courses in informatics. However, for various reasons, the majority of students have difficulties understanding the basic principles of procedural programming taught in introductory courses. This is due to the need to introduce too many notions at the same time, e.g. data representation, control statements, definition and use of subroutines. Bearing in mind these difficulties, many instructors do not introduce students to subroutines. They postpone this issue because subroutines are hard to master in a short period of time. Unfortunately, learning subroutines at a later stage leads to a habit of avoiding the use of subroutines and to an improper programming style.

Different approaches and techniques are used for improving the teaching process, e.g. problem-oriented approach [4], methodological approach [13], integration and synthesis [1], modern information technologies [10], learner autonomy [11], investigations on the notion [12]. A system of basic tasks was developed for teaching event-driven programming [2,3]. The role of the visualization of algorithms in support of education in Informatics is reported in [6].

* This paper is partly supported by the Scientific Research Fund at Plovdiv University “Paisii Hilendarski”, Contract RS09-FMI-014

The idea for using intelligent computational methods and tools for training is discussed in [7]. An architecture of computer-based system providing support for the whole process of learning and teaching of informatics is presented in [8]. An analysis of student views on the methodology used to raise their interest in the discipline, the attractiveness of the presented material and its intelligibility is described in [9]. The good approach to programming numerical algorithms [5] is to use subroutines.

A survey was set up in order to identify the biggest challenges for the students. Answering the question "What part of the material in programming was the most difficult to learn and why?", the respondents indicated 3 main problems: basic concepts, subroutines and arrays [9]. Analyzing the most common mistakes, students make when they create or call subroutines, we made the following conclusions:

- the mechanism of transfer of parameters remains a black box for the students for a long time period;
- too many global variables are used in subroutines;
- local variables often appear in the list of formal parameters.

Even though these problems are often discussed with the students during lectures and exercises, the need for additional self-training over a long time period is obvious. To that end, in this paper we propose two approaches. The static approach is appropriate for written text in a paper textbook. The dynamic approach, however, is suitable for interactive training using computer.

2. A static approach for self-training

The integration of different concepts (scope of the variables, the mechanism of parameters transfer) in an example supports the teaching process. A set of examples has been proposed to students in order to assimilate these concepts and use them properly. One of them is shown below.

Example: What is the result of the following program if ***a***=2, ***b***=3, ***c***= –1, ***d***= –2 ?

```
program izpit;
var a,b,c,d:integer;
procedure p1(x:integer; var y:integer);
   var a:integer;
{The local variable a hides the global variable a}
   begin
      a:=1;
      c:=y-2*a;
      x:=y+2*a;
      writeln(a:4,b:4,c:4,d:4);
      y:=a+x;
   end;{p1}
begin {Main program}
   write('a,b,c,d='); readln(a,b,c,d);
   p1(a,b);
   writeln(a:4,b:4,c:4,d:4)
end.
```

Explanation: We have 4 global variables: ***a***, ***b***, ***c*** and ***d***. Four different cells in the memory are allocated for them. The local variable ***a***, and the formal parameter ***x*** have a local scope. The formal parameter ***y*** does not require a specific cell. It uses the cell of the actual parameter after the subroutine call (in this case - the global variable ***b***). The table below shows the step by step execution of the program in the example by considering the values of the variables at each step. We start from the main program.

Statements	Global scope				Local	
begin {Main program}	**a**	**b**	**c**	**d**	**a**	**x**
write('a,b,c,d='); readln(a,b,c,d);	2	3	–1	–2		
p1(a,b); {**a**→**x**, **y** and **b** have the same address}		**b,y**				2
begin						
a:=1;{Local variable **a** }					1	
c:=y-2*a; {3-2*1→1 }			1			
x:=y+2*a; {3+2*1→5 }						5
writeln(a:4,b:4,c:4,d:4); {1 3 1 –2 }		3	1	–2	1	
y:=a+x; {1+5→6 }		6				
end;{p1}						
writeln(a:4,b:4,c:4,d:4) { 2 6 1 –2 }	2	6	1	–2		
end.						

Answer:
1 3 1 –2
2 6 1 –2

3. An interactive self-training module

The dynamic approach is combined with appropriate visualization of the process of subroutine execution. The subroutine has two formal parameters. Unlike the static approach, the dynamic one allows the student to interactively choose the kind of every parameter: value or variable. This is why we created an interactive module as a new component of a computer-based system designed to enhance the whole process of informatics training. This module cannot be replaced by the embedded debugger due to the following reasons:

- a beginner in programming languages is not keen on using the debugger along with many other new rules;
- the module is easy to use and track the execution process;
- there is no need to set up watches;

- there is no need to switch between the program window and input/output window;
- additional information appears for global and local variables.

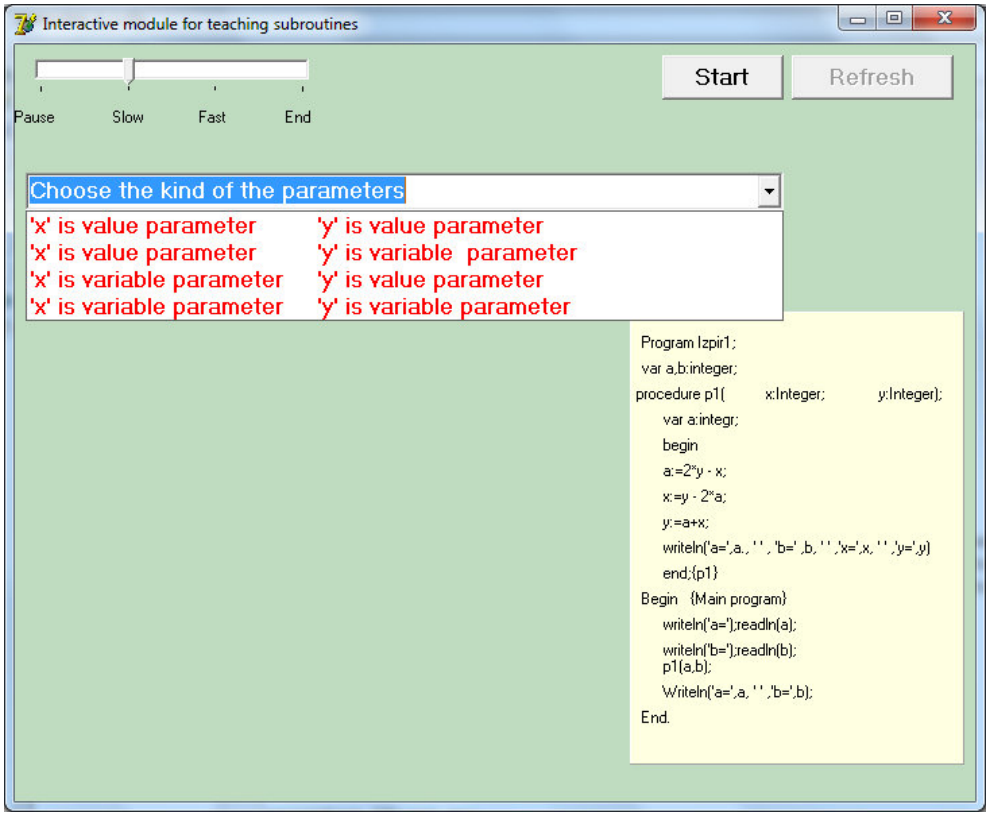

Figure 1. Beginning of the interactive module (screenshot)

The dynamic approach has the following advantages:

- many static examples are integrated in one dynamic example;
- different initial values of the global variables can be tested;
- the execution can be made step by step at a varying pace, including pauses

Some screenshots of the interactive module are shown in Figures 1-3.

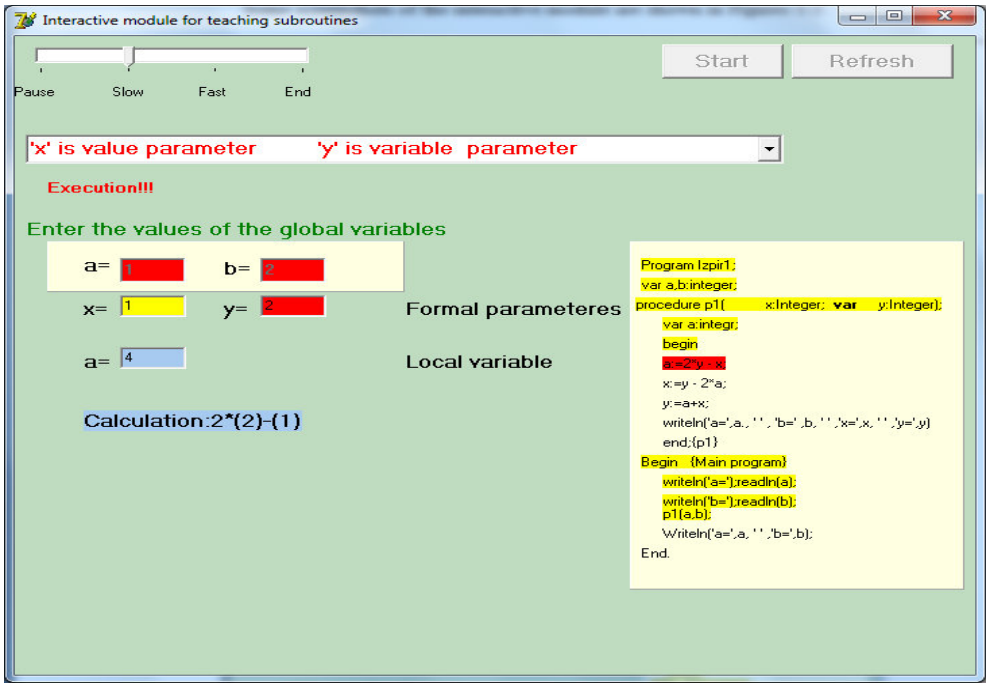

Figure 2. Middle of the interactive module (screenshot)

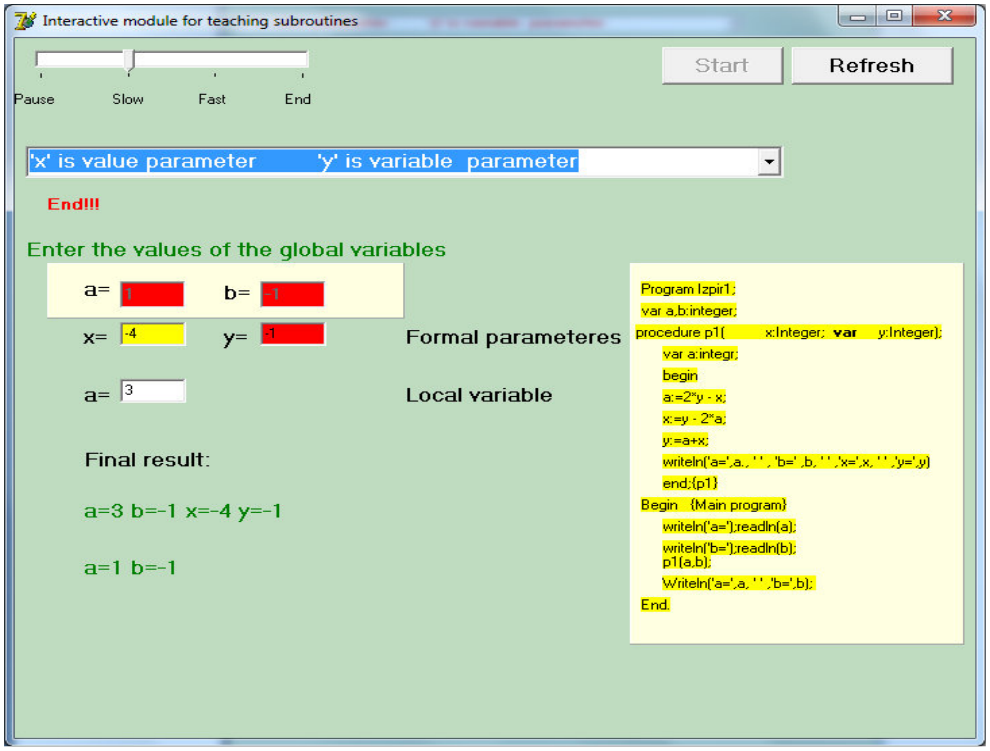

Figure 3. End of the interactive module (screenshot)

4. Conclusion

Teaching subroutines as early as possible is an important prerequisite for obtaining good programming skills. Subroutines help students follow a more sophisticated programming style. They are an important part of the classes in object-oriented programming. The two approaches presented in this paper concern the possibilities for additional self-training in the process of learning subroutines. The dynamic approach is the better one and an interactive module for applying this approach has been developed. Implemented with Turbo Delphi IDE, it runs under WINDOWS'97, WINDOWS XP, WINDOWS VISTA and WINDOWS 7.

References

[1] Angelova, E., T. Ditcheva, Integration and Synthesis in Teaching Spreadsheet Processing to Information Technology Teachers, Scientific works of Plovdiv University „Paisiy Hilendarski", vol. 37, book 3, Mathematics, (2010), 5-12.

[2] Aneva, S., A system of basic tasks for teaching event-driven programming using Visual C# environment in high school, (In Bulgarian), Proceedings of the national conference "Education in Information Society", May 27-28, 2010, Plovdiv, pp.239-251.

[3] Aneva, S., The role of basic problems in learning of event-driven programming with Visual C# environment in high school, (in Bulgarian), Proceedings of the Anniversary International Conference, Synergetics and Reflection in Mathematics education, September 10-12, (2010), Bachinovo, Bulgaria, pp.353-363.

[4] Grozdev, S., E. Angelova, Problem-oriented approach in preparing teachers for teaching "Information Technologies" in schools, Pedagogics, 2, 2010.

[5] Iliev, A. N. Kyurkchiev, Nontrivial Methods in Numerical Analysis: Selected Topics in Numerical Analysis, LAP LAMBERT Academic Publishing, 2010.

[6] Krushkov, Hr., "Visualization of algorithms in support of education in Informatics" (In Bulgarian), Scientific Researches of the Interdisciplinary Forum "Bulgaria and Russia - the directions of reciprocity", 14-17 December, 2008, Russe, pp.166-178.

[7] Krushkov, Hr., M. Krushkova, D. Kolentzov, D. Krushkova, Y. Stoynova, "Development of intelligent computational methods and tools for training" (In Bulgarian), Scientific Researches of the Union of Scientists - Plovdiv, Plovdiv, 05-06 November, 2009, pp.412-415.

[8] Krushkov, Hr., M. Krushkova, V. Atanasov, M. Krushkova, "A computer-based tutoring system for programming" (In Bulgarian), Proceedings of the XXXIX-th spring conference of the Union of Bulgarian Mathematicians, April 6-10, 2010, Albena, pp.354-358.

[9] Krushkov, Hr., M. Krushkova, D. Krushkova, "How to improve the teaching in computer science" (In Bulgarian), Proceedings of the national conference “Education in Information Society”, May 27-28, 2010, Plovdiv, pp.110-116.
[10] Rahnev, A., E. Angelova, Training Teachers of Mathematics in the use of Modern Information Technologies for Teaching, Proc. of the 6th Mediterranean Conference on Mathematics Education, 22-26 April 2009,Plovdiv, Bulgaria, 79-83.
[11] Shotlekov I., P. Tanneva-Kafelova, Learner Autonomy in Teaching ICT Through English, Proceedings of the Vth Scientific Conference on Management, Oct. 30-31, 2008, Plovdiv, pp. 250-253.
[12] Terzieva, T. Results from investigations on the notion of algorithmic thinking, Proceedings of the Anniversary International Conference, Synergetics and Reflection in Mathematics education, September 10-12, 2010, Bachinovo, Bulgaria, pp.479-486.
[13] Valchanov, N., A. Iliev, T. Terzieva, One methodological approach for teaching in Information modeling, Proceedings of the Anniversary International Conference, Synergetics and Reflection in Mathematics education, September 10-12, 2010, Bachinovo, Bulgaria, pp. 487-484.

Mariana Krushkova, Yordanka Stoynova, Hristo Krushkov
Faculty of Mathematics and Informatics
University of Plovdiv
236 Bulgaria Blvd.
4003 Plovdiv, Bulgaria
e-mail: mik@uni-plovdiv.bg, dani_as@abv.bg, hdk@uni-plovdiv.bg