Available online at www.sciencedirect.com

ScienceDirect

Procedia - Social and Behavioral Sciences 171 (2015) 172 – 177

Procedia
Social and Behavioral Sciences

ICEEPSY 2014

Development of programming capabilities inspired by foreign language teaching

Eva Milková*

University of Hradec Kralove, Rokitanského 62, Hradec Kralove 500 03, Czech Republic

Abstract

Education of subjects related with computer science is from the perspective of other for centuries taught subjects, still in its infancy. Let us consider, for example, a teaching method aimed at developing algorithmic thinking of students. Even nowadays this area is still the subject of extensive discussions and teachers are looking for different ways how to access it to students. As the ability to create algorithms is the cornerstone to understand syntax and semantics of programming languages, it is obvious that a deeper look into learning a language can provide stimulating and interesting ideas for teaching algorithms.

Peer-review under responsibility of the Organizing Committee of ICEEPSY 2014.

Keywords: Education; algorithms; programming capabilities; testing; English language teaching and learning;

1. Introduction

The essential part of studies at the faculties preparing students in the area of computer science is the development of a student's ability to think algorithmically. There are many different approaches to the education whose aim is to enhance algorithmic thinking. It is not easy to find the right way and the most difficult one seems to be the one dealing with novices.

Looking at teaching subjects that are close to the mentioned area, drawing inspiration from them, the appropriate implementation in the targeted teaching process along with good teaching experience are good presumptions to succeed and reach the aim. Speaking about algorithms, the development of the ability to create algorithms is the

* Eva Milková. Tel.: +420 49333 1337.
E-mail address: eva.milkova@uhk.cz

Peer-review under responsibility of the Organizing Committee of ICEEPSY 2014.
doi:10.1016/j.sbspro.2015.01.104

cornerstone to understand syntax and semantics of programming languages. Thus, a deeper look into learning a language, especially a foreign language, can provide stimulating and interesting ideas for teaching algorithms.

The paper first introduces our approach to teaching algorithms based on many years´ successful experience in teaching the subject Algorithms and Data Structure taught at University of Hradec Králové, Czech Republic. We also introduce the idea that has led us to this approach. Subsequently, we focus on the relationship between algorithms and teaching a foreign language. We pay a particular attention to the part of teaching process that relates not only to the knowledge acquired in lessons but also to students´ individual preparation. The validation of acquired knowledge in teaching algorithms is conducted especially by program Algorithms. This application and possibilities of its practical use is briefly presented at the end of the paper.

2. Subject Algorithms and Data Structures

The first programming course, Algorithms and Data Structures (ALGDATS shortly), teaches our students their very first steps in algorithm design. The process of algorithm design is explained using a modular, object-like paradigm – students are building interesting objects out of a few basic elements. This imagination is based on a brick-box, a nice and useful game for children, where are only several base elements available from which children are able to create incredible buildings. (Milková, 2007)

Students create algorithms in the Czech pseudo-language (based on Pascal programming language), they write them on papers.

Using three *simple commands* (`read`, `write` and assign `:=`) together with four *structured commands* (blocks of commands, incomplete and complete branching and loop construction) containing only seven keywords (`begin`, `end`, `if`, `then`, `else`, `while`, `do`) students are able to understand soon the *typical algorithmic structures* (for example determining the number of elements with the given property, calculating the sum and the product of some elements, finding the first and last element with the given property etc.) and to use these structures when solving both *easy as well as more difficult tasks*.

Algorithmic structures are thoroughly explained and introduced on practical examples in lectures. During the lessons students apply the acquired knowledge to a variety of tasks.

2.1. Lectures

In lectures we explain all structures of algorithms, at first only those which use single variables. We always try to use names of variables that describe their usage. Obviously, in the beginning, examples of algorithms are demonstrated graphically by developing diagrams. In the diagrams we use only two types of shapes: a rectangular for commands and a rhombus for conditions. We properly explain the idea of each presented algorithm and usually demonstrate it with the help of students (e.g. each student tells a value and the action of the algorithm is introduced), as well as we illustrate the action of each algorithm by a step-by-step procedure for suitable initial values.

After the thorough practising basic algorithmic structures on problems using single variables we proceed further and explain the data structure of one-dimension and two-dimension array. Using these data structures all previous matter is repeated together with a careful attention to the work with array indices.

2.2. Lessons

During lessons students apply the acquired knowledge to a variety of tasks. They work in groups of two or three and each group is responsible for solving one of the given tasks. After some time when students prepare their solutions on a piece of paper, each task is illustrated and presented by two students (or three, depending on the number of groups, each group responsible for the task-solution deputes one student) on the blackboard and their solutions are compared and discussed by all students. On the one hand this means that students are led to try and