JÓZSEF UDVAROS

THE ROLE AND POSSIBILITIES OF DIGITAL TECHNOLOGIES IN THE TEACHING OF PROGRAMMING

THESES OF THE DOCTORAL DISSERTATION

PhD School of Informatics
Eötvös Loránd University Faculty of Informatics

The title of the PhD Program: Basics of informatics and its methodology

Leader of the PhD School: Dr. Erzsébet Csuhaı̈ Varjú
Leader of the PhD Programme: Dr. András Benczúr

Advisors:
Prof. Ing. Veronika Stoffa, CSc.
Dr. László Zsakó

Introduction

Modern operational systems and developer environments implement OOP principles and today’s developments could not be imagined without the use of OOP. For practicing programmers and developers the knowledge of OOP and system development methods related to it is indispensable. According to this, we can say that teaching of computer science and especially programming should be based on OOP. In basic computer science education the use of OOP can be avoided, although when teaching programming it is unavoidable. From teaching experiences it is known that teaching traditional sequential programming faces lot difficulties, but teaching OOP, which is closer to reality, is even harder. This comes from the fact that further modelling and theoretical knowledge is needed, at the same time traditional sequential programming principles need to be taught. A question arises as to when it is subservient to start teaching OOP. Especially when we think of the fact that in order to learn programming it is important to develop an algorithmic mode of thinking. The answer to this question is very important, but other questions arise as well: with what method we shall teach the student, how we can teach the students this technique and technology in the most efficient and most timely matter.

My research is based on teaching-methodology, and proves the importance of visualizing OOP concepts.

Motivation of choosing the theme

Computer science is one of the most important subjects at the Kodály Zoltán Gymnasium of Galanta. The school is improving their computer science curriculum according to their opportunities. Buying and using new technology puts the education on higher level. This is not enough though, a lot depends on the attitude of the teachers and the students.

With our help, our students take part in Slovak as well as Hungarian programming, use of programming and program management competitions. In the past years our students have always had excellent performance on the programming competitions. They have taken their place on podiums on multiple state and international competitions.

The teacher’s job is to provide the students with the necessary materials, presentations and examples needed for their studying at home. The simplest way for this is using a wide range of frameworks, as well as websites.
The development of talented individuals and preparation for competitions unfortunately does not fit into the curriculum, this is why the talented individuals are embraced and we prepare them for the competitions during extracurricular activities. In my point of view this is not enough. Students solve only one or two examples which do not give them enough programming experience. The programming facility as well as the algorithmic thinking can only be achieved by a tremendous amount of practice, and writing a great deal of programs. The number of computer science classes is decreasing and the examples at competitions are getting more and more demanding, therefore it is necessary for students to practice programming even at home.

Based on my experience, the amount of problems regarding students preparation for competitions as well as their mastering the curriculum has grown in the past year. When teaching OOP the problems regarding students capabilities to master the curriculum have come to surface, the reason behind this is the fact that it requires abstract thinking.

**Antecedents and objectives**

The world around us is dynamically evolving. We are witnesses of the modern technology and the fast evolution of communication, which brings new challenges and means responsibility for future strategies and the formation of the human approach. The solutions of practical problems cannot be imagined without serious modelling and computer science knowledge. Modern problem solving methods and application of artificial intelligence methods involves a serious background in computer science.

Although this level of computer science knowledge can only be expected from the best students, the current level at which computer science is taught means that it is not sure the best students will end up in the computer industry. Generally speaking, we can say that in the beginning of the 21st century, in this technological society, it is unacceptable for a student to finish secondary education without learning the basics of algorithms and not to understand programming, in other words the basics of program development.

Based on my experience there is a large backlog in this area. Day after day we can see that the students’ computer science knowledge is on a very low-level, at the same time the market for computer science experts is increasing. This problem can only be solved by adjusting the curriculum to the needs of the students. Naturally, for this we need to examine that our experience with the secondary school computer science level is not just a feeling, but that in
fact the situation is really that bad. This is why the first step in my research I measured the
computer science knowledge-levels, within that the knowledge of concepts of programming
and OOP of Hungarian secondary school students studying in Slovakia.

While talking to computer science teachers, a question arises about the quality of Slovak
computer science education and within this the quality of programming education. With what
basics do the students arrive from the elementary schools to the secondary vocational schools
that end with a school leaving exam and to high schools? How do the students appreciate the
basic computer science knowledge that is very important in today’s computer and technology
knowledge based lives? With how much practical knowledge about computer use do they
graduate with, and how much do they carry on to their universities and workplaces?
Programming, and within it OOP teaching is running into great hardships and the reasons
behind it need to be found out. Based on the above I had the idea to map the current situation.
This study shows how we should adjust the teaching methods in order to achieve the best
results in teaching learners.

**Applied methods**

My first research was to measure students’ capability of programming. The survey was done
online, because I wanted to see the opinions of as many students as possible. I achieved this
with a practical online survey. I sent the survey to almost all schools taught in the Hungarian
language, so the answers were collected from the entire Hungarian populated area. While
putting together the survey I focused not only on their faculty for programming but on their
general computer usage habits as well.

The contents of the survey were pieced together with the help of multiple experts. The
questions were sorted into multiple categories.

- The first category revolved around personal characteristics.
- The second category revolved around their attitude towards computer science, such as
  how much time they spent working with computer science in a day.
- The third category revolved around what level the student estimated themselves at.
- The following questions tested their basic computer science knowledge.
- The last category tested their knowledge of programming and OOP using concepts of
  the same.

The questionnaire was completed by 405 students in 3 weeks, according to the following:
<table>
<thead>
<tr>
<th></th>
<th>Boys</th>
<th>Girls</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st grade</td>
<td>40</td>
<td>42</td>
<td>82</td>
</tr>
<tr>
<td>2nd grade</td>
<td>34</td>
<td>62</td>
<td>96</td>
</tr>
<tr>
<td>3rd grade</td>
<td>38</td>
<td>56</td>
<td>94</td>
</tr>
<tr>
<td>4th grade</td>
<td>14</td>
<td>18</td>
<td>32</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>126</td>
<td>178</td>
<td>304</td>
</tr>
</tbody>
</table>

Respondents who have filled the questionnaire (Gymnasium)

<table>
<thead>
<tr>
<th></th>
<th>Boys</th>
<th>Girls</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st grade</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2nd grade</td>
<td>35</td>
<td>5</td>
<td>40</td>
</tr>
<tr>
<td>3rd grade</td>
<td>22</td>
<td>0</td>
<td>22</td>
</tr>
<tr>
<td>4th grade</td>
<td>37</td>
<td>1</td>
<td>38</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>94</td>
<td>7</td>
<td>101</td>
</tr>
</tbody>
</table>

Respondents who have filled the questionnaire (Secondary vocational school)

During my first research when mapping students’ OOP knowledge, I tested the familiarity of the phrases: class, object, object-oriented program, method and inheritance. The evaluation of the answers returned a very weak result. Based on the research I could state that the teaching of OOP in Slovak secondary schools taught in Hungarian was on an entry level. When evaluating the reasons behind the poor results I came to the conclusion that it would not be enough to increase the number of computer science classes but a strategy would need to be developed (and methods and methodology along with it), as well as complementing this method by making it more interactive and providing e-learning opportunities.

In addition to all of this, a well built application, thanks to which we could make the teaching more efficient, was necessary. My research was built on this method and application effectiveness.

Compared to my earlier research, as a step forward besides the use of the application I tested how much the understanding of OOP evolved in the students, as well as proved my hypotheses.

The following 4 figures illustrate the students’ knowledge about the basic concepts of OOP:
Figure 1. Students’ knowledge about the concept of the class

Figure 2. Students’ knowledge about the concept of the object

Figure 3. Students’ knowledge about the concept of the method

Figure 4. Students’ knowledge about the concept of the inheritance
For my second research, I created an application. The program is not final yet, it is only a beta version, which serves only to prove or disprove the test and hypotheses.

The research’s goal was not to create a complex tool, but to show what kind of tool should be developed and for what age category.

I created the application for secondary school students, where they can use it under the supervision of a teacher.

![Image](image.png)

**Figure 5. Introduction of the concept of the class and object**

Before undertaking my research, the students’ knowledge about OOP was equal to zero as they had never learned OOP before. I organized a programming course for students from 2nd to 4th classes (the students were 15-19 years old). I divided them into two groups: the first group — so called experimental group — could use the newly developed application, while the second group — so called control group — learnt without this aid. I chose students with different skills to prove the efficiency of the software. Both groups included very skilled and not so talented students as well.

I chose two schools — The Kodály Zoltán Gymnasium of Galanta and the Jedlik Ányos secondary vocational school in Nové Zámky — as there were very few schools in Slovakia where OOP is taught. In the research 167 students took part, where 79 students used the newly developed application, but 88 students didn’t use it.
The questionnaire was compiled based on several conversations with experts and specialists. The questionnaire contained both theoretical and practical questions.

**Hypotheses**

While analyzing the results of my second research, I developed my hypotheses, as follow:

H1: Students’ understanding and learning OOP, moreover, the effectiveness of teaching OOP can be significantly enhanced with the developed application for visualization.

H2: The students’ modelling and problem solving skills can be improved by the new visualization and gamification method.

H3: The developed application and the introduced method help to get practical training in programming and to increase learners’ programming skills.

![Figure 6. Comparison of the experimental and control groups’ average results (green - experimental group, blue - control group)](image)

The proof of H1 hypothesis:

Questions no. 1, 3, 4, 6, 8, 10, 12 and 15 of the survey focused on the theoretical knowledge of object-oriented programming, while the other questions examined the practical applicability of OOP. As Figure 6 above clearly shows, the experimental group performed much better for each question than the control group (they had better average). As a second step, I examined separately whether there was any link between the results of the experimental and control group responses. I determined Pearson's correlation coefficient using the IBM SPSS Statistics 22 statistical software. It could be concluded that the
correlation between the control and the experimental groups and the given answers to each question was positive, ranging from 0.646 to 0.790, so that there was a moderately strong stochastic relationship. In order to determine whether the correlation coefficients ranging from 0.646 to 0.790, obtained from 167 data, were sufficiently high to be generally considered as most likely to occur, the significance of correlation coefficients should be investigated. In our case, the 0.000 value in the Sig (2-tailed) line was less than 0.05, so the null hypothesis, that the relationship between the two variables was just a coincidence, could be rejected. That way I proved my H1 hypothesis.

The proof of H2 hypothesis:
The questions number 4, 5, 6 and 7 focused on the students’ knowledge about modeling. Using the visualization application, that included elements of gamification, significantly increased the students’ knowledge about practical usage of modelling. This was also demonstrated by the results of the survey as the experimental group achieved better results than the control group (see Figure 6 above). In addition, the same as during the verification of my first hypothesis, I examined separately whether there was any connection between the results of the experimental and control groups’ responses. I also examined Pearson's correlation coefficients. Correlation coefficients ranged from 0.771 to 0.790. The values showed a strong positive relationship between the data being investigated. Thereby, I proved my H2 hypothesis.

The proof of H3 hypothesis:
The experimental group's students had a chance to experience playful visual apps during the learning process. They also had an opportunity to use online tutorials at home, they were also given solved examples and unresolved tasks. It was obvious from the answers given for practical questions (questions 2, 5, 7, 9, 11 and 13) that the application and the whole method helped to get practical training in programming and helped to increase learners’ programming skills. It was clearly seen that the number of correct answers given to the practical questions both in the experimental and in the control groups significantly increased (see Figure 6). Thus, I proved my H3 hypothesis.
Summary

In my dissertation I wanted to find a solution, a method, to the pedagogical problem that had arisen. The teaching of programming had always been a hard job, mostly due to the decreasing number of classes as well as the students’ know-how. As a solution for this problem I invented a simple visualization application with the elements of gamification and proved with a simple program that education significantly became more efficient with the use of such an application. The students found it easier to deepen their OOP practical knowledge. With the gamification elements I managed to get the students to look at the educational-visualization application as a simple game via they learned the given concepts as well. The application just tested the main elements and mostly served to prove my hypothesis. As a step forward I want to develop further this app. On one part to create a web based solution, as well as making the visualization more interesting and implementing more gamification elements. Moreover, I would like to show OOP’s cooperation possibilities, introduce different visibility elements as well as to visualize the virtual class attributes and methods.

My plans include implementing a code generator as well, which during the visualization, automatically generates the code. Beside the visualization I would like to develop an e-learning curriculum, with solved examples and an archive of examples for learning the basics of OOP.

Publications of the author according to the topic

UDVAROS J. – GUBÁN M. (2016). Demonstration the class, objects and inheritance concepts by software. ACTA DIDACTICA NAPOCENSIA 9:(1) Paper 3., ISSN 2065-1430


Further Publications


