15/1-2. (2017), 65–75
DOI: 10.5485/TMCS.2017.0432
tmcs@science.unideb.hu
http://tmcs.math.unideb.hu

Teaching Mathematics and Computer Science

Analysing the effects of OOP helper application

József Udvaros

Abstract. Nowadays students of secondary schools are familiar with the usage of computer very soon, lot of them are even capable of handling user applications very cleverly. This is satisfying for most of them. Those who imagine their future in programming or system developing, need to have deeper knowledge about object oriented programming, however, students do have it at very low level or not at all. We want to make sure whether this suppose is true, so different examinations have recently been made at Slovakian secondary schools with Hungarian teaching language. We have reached a conclusion that the students' knowledge of object oriented programming is deficient. We could achieve better results by using proper applications as a visual aid. In this paper we examine the efficiency of an application made by us.

Key words and phrases: OOP, teaching object-oriented programming, helper application.

ZDM Subject Classification: U60.

Introduction

The world all around us is rapidly developing. We are witnessing the rapid evolution of technology and communication. This means new challenges and responsibilities to future strategies and attitudes. Solving practical problems today is inconceivable without modelling and IT skills. The [7] is also showing an example where an IT model is needed to solve a logistic problem. We can expect such knowledge only from the best students; however, the current IT level of the secondary schools' students is not efficient. The programming is difficult task [2]. At the beginning of the 21st century, it is unacceptable for students not to master the basics of algorithm during their secondary school studies, moreover, not to understand the basic concepts of programming and program development. Day by day we are facing the fact that our students' computer skills are at very low level, while the market increasingly requires IT professionals. We can solve this problem with modifying IT teaching in accordance with the requirements of young students. First of all, we need to prove that the situation is really so bad. As the first step, we have to undertake a research where the level of IT knowledge was measured at Hungarian secondary vocational schools in Slovakia, focusing mainly on the concepts of programming and OOP knowledge. The [8] is showing the results of this research, and defines the solution in gamifications and e-learning. Visual aids are very important by the perspective of effective education [6]. OOP visualization tools like BlueJ (Java) do already exist. According to me, this tool requires a much deeper knowledge of the basic terminology and doesn't contain elements of gamification, but their introduction at secondary schools is improper. Because of that, we made an own application, which is based on gamification. The program helps to understand the terms as class, object, and inheritance in DELPHI, so students with minimal programming skills will be able to deepen their knowledge, thanks to this visualization. The use of this program does not require knowledge of UML, respectively knowledge of other abstraction presentation systems. The [3] is showing this application.

We also need to prove the following hypothesis by our application:

(1) Teaching OOP via visualization application allows students better understanding both the concept and the application itself, too.

(2) The students' modelling skills can be improved by the new visualization and gamification method.

(3) The students can significantly better understand the object-oriented concepts thanks to the newly developed simple application.

(4) The application and the whole method facilitate the practical education of programming and increases students' programming skills.

The program's effectiveness and applicability in the educational process

Before undertaking our research, the students' knowledge about OOP was equal to zero as they had never been learning OOP before. We organized a programming course for students from 2^{nd} to 4^{th} classes (the students were 15-19 years old). We 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. We chose students with different skills to prove the efficiency of the software. Both groups included very skilled and not so talented students as well. We chose two schools–The Kodály Zoltán secondary grammar school in 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. The survey was conducted with on-line method.

The questions we asked were the following:

(1) What does an object mean?

(2) Can you say some examples for objects?

(3) Why do we need modelling?

(4) What is the correlation between model and reality?

(5) Can you say an example for a model?

(6) What does generalisation mean?

(7) Can you say some examples for generalisation?

(8) What is a class?

(9) What is the difference between an object and a class?

(10) Which attribute can be that when a specific element gets an attribute of a more general element?

(11) Can you say an example?

(12) Can two objects exist from one class with different parameters?

(13) What does it mean in practice?

(14) Program recognition.

(15) What is an abstraction?

After analysing the results of the survey, we could clearly state that the results of the group which used OOP application were better than the other group's results. Next we show results of survey, which was published in [1].

Students' knowledge about the object

There were 4 possibilities with one correct answer when examining the students' knowledge about the object. The students in "control group ", who didn't use our application, gave 63.64% correct answer, while the students in other group, who used our application, gave 79.75% correct answers. In the practical part of the survey, students had to give an example for an object. The better results (96.20%) were reached by the students who used our application, while the other group had 87.50% correct answer. We could clearly see which examples were used by the teacher.

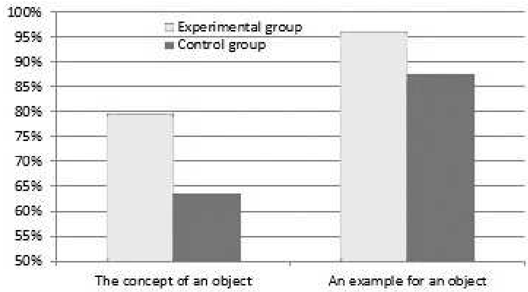

Figure 1. Examining the concept of an object

Students' knowledge about modelling

94.94% of the students in experimental group, who used our application, gave the correct answer for the question: "Why do we need modelling?"while the other group's success was 90.91%. 93.67% of the students using the application could define the correlation between model and reality, while the other group's success was 90.91%. Fewer students could give a practical example than to answer our theoretical question. The ratio was 88.61% vs. 80.68%. Students' knowledge about practical usage of modelling could be improved by using visualisation, which contained elements of gamification, too. The gamification is new teaching method [4], [5]

The students' results related to generalisation

When examining the concept of generalisation, the students could choose from 3 options. The group which used our application gave 96.2% correct answers

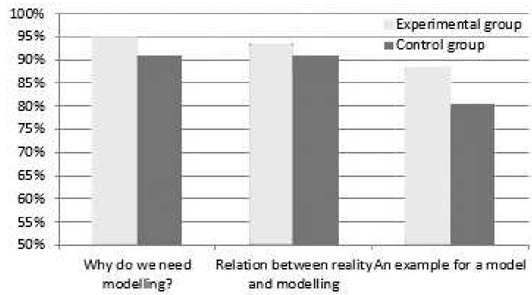

Figure 2. Knowledge about modelling

while the control group's success was 90.91%. When we asked about a concrete example of generalisation both groups were weaker by nearly 10%. The result in percentage was: 88.61% vs. 80.68%.

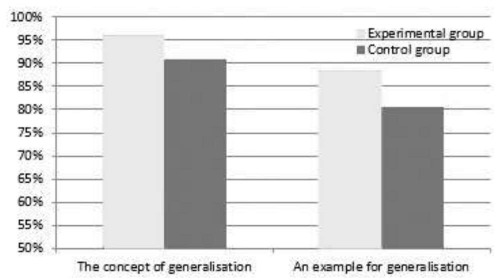

Figure 3. The students' knowledge related to generalisation

The students' knowledge about the concept of the class

The concept of the class was defined correctly by 88.61% of the students in the experimental group and by 84.09% of the students in the control group. The next question required an example for the difference between the class and the object. The results between the two groups showed big difference, as the students using the application gave 87.34% correct answers compared to second group with 75.86% correct answers.

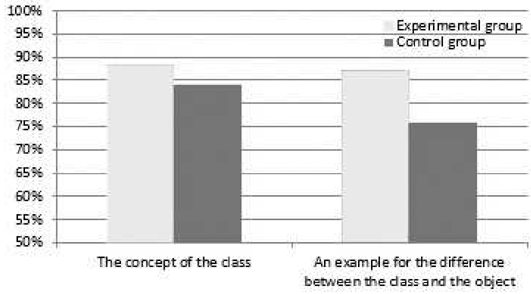

Figure 4. The concept of the class

The students' knowledge about the inheritance

"What kind of attribute is it when a specific element gets an attribute of a more general element?"92.41% and 90.91% of the students gave the correct answer, which was "the inheritance ". Then they had to give an example of inheritance. "Can two objects exist from one class with different parameters?"It was a yes and no answers but the practical part was justified much better by the experimental group who used our application.

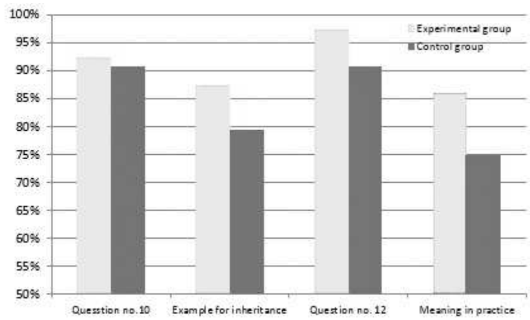

Figure 5. Knowledge about the inheritance

Recognition of a program detail

The students had to decide whether the following part of the programme defines a class or an object in DELPHI.

We can see the results in the following graph:

```
Type
TShape = Class
Name      : String;
Height    : Integer;
Width     : Integer;
Procedure Shape_Create();
Procedure Change_Height();
Procedure Change_Width();
End;
```

Figure 6. The examined program detail

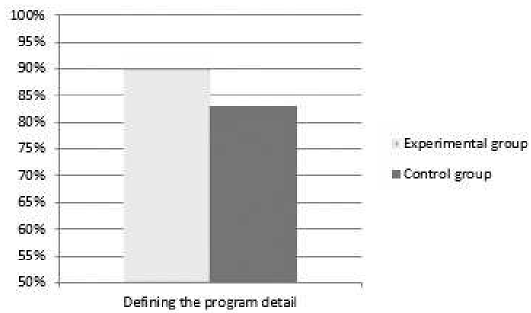

Figure 7. Recognition of a program detail

The students' knowledge about the abstraction

When examining the concept of the abstraction, the students had to choose the correct answer from 4 options. 93.67% of the students using our application answered correctly, while the control group's success was 88.64%.

Comparing the results of the two groups

We produced a graph comparing the results of the two groups. The differences between the two groups could be clearly seen.

Verifying the hypothesis

Verification of the hypothesis no. 1:

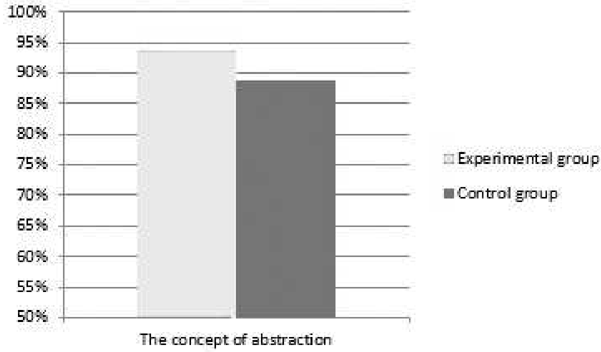

Figure 8. Students' knowledge about the abstraction

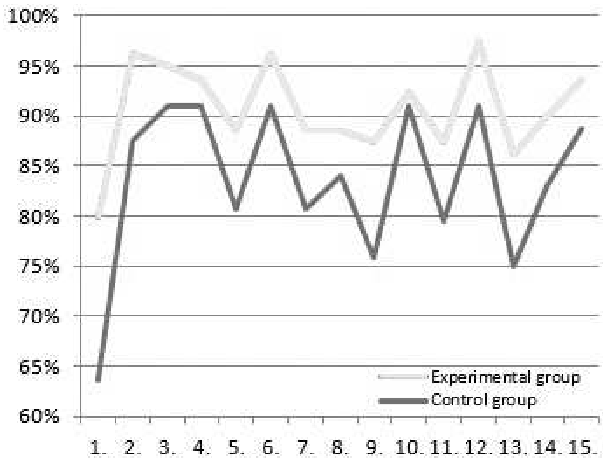

Figure 9. Comparing the results of the two groups

Questions no. 1, 3, 4, 6, 8, 10, 12 and 15 focused on theoretical knowledge of object-oriented programming, while the other questions examined the practical applicability of OOP. The Figure no. 9 above clearly revealed that the experimental group using our application performed much better for each question compared to the control group. The percentage differences between the answers of the two groups were as follows: knowledge about the concept of the object: 16.11%, the importance of modelling: 4.03%, the relationship between reality and modelling: 2.76%, the students' knowledge about generalization: 5.29%, their knowledge about the concept of the class: 4.52%, students' knowledge about the inheritance: 1.50%, making differences between the class and object: 6.56% and the students' knowledge about abstraction: 5.03%. Thus we confirmed our first hypothesis, according to that: "Teaching OOP via visualization application allows students better understanding both the concept and the application itself, too."

Proof of the hypothesis no. 2:

The questions no. 4, 5, 6, 7 were connected with modelling. Using the visualization application, that included elements of gamification, significantly increased the students' knowledge about practical usage of modelling. While evaluating the questions some significant differences were discovered between the two groups' correct answers. The differences in percentage were as follows: 2.76% was the difference for the question about relationship between realities and modelling, 7.93% was the difference between the two groups when we asked them to give an example for a model. The question about the concept of generalization turned out with 5.29% difference and there was 7.93% difference when explaining the generalization on the concrete examples. Thus we confirmed our second hypothesis, according to which "The students' modelling skills can be improved by the new visualization and gamification method."

Proof of hypothesis no. 3:

The results confirmed that students' understanding of object-oriented concepts could significantly be increased with the help of the newly developed application, which was in fact our third hypothesis. Teaching OOP was more effective by using visualization tools, which the students could try immediately. While developing the visualization tools we had to be careful to get the tool as simple as we could and to include elements of gamification. Today's young people who grew up with playful mobile applications would master the concepts of OOP faster, easier and more efficiently.

Verification of the 4th hypothesis:

We could clearly see from the responses for the practical questions that "The application and the whole method facilitate the practical education of programming and increases students' programming skills ", which was also our fourth hypothesis. The increased number of the correct answers given for the practical questions was the evidence for our fourth hypothesis.

Summary

We made a research and a survey at secondary schools in Slovakia about the applicability of OOP in education with an application. The result of our survey proved the fact that this application helped quicker understanding of the terms as inheritance, class, object, and even students with only basic ideas about programming could deepen their knowledge by the help of visualization. As the result showed, the answers for the theoretical questions were not as influenced by the application as the practical questions, where the knowledge of the students improved a lot. Therefore, we could claim that the programming and modelling skills could be developed by this tool. The difference between object and class became more understandable for those students, who had opportunity to practice individually by using our application. Programming is a practical task that needs to be done individually; therefore it is so important to gain good practical programming skill. We can make a statement that objects oriented programming can be taught more effectively by using visual aids. We have also proved our following hypothesis:

(1) Teaching OOP via visualization application allows students better understanding both the concept and the application itself, too.

(2) The students' modelling skills can be improved by the new visualization and gamification method.

(3) The students can significantly better understand the object-oriented concepts thanks to the newly developed simple application.

(4) The application and the whole method facilitate the practical education of programming and increases students' programming skills.

References

[1] M. Avornicului, *Bevezetés a számítógépek programozásába*, Ábel Kiadó, Kolozsvár, Románia, 2016.

[2] M. Gubán, J. Cselényi and D. Vadász, Comparing method of mathematical programming and heuristic method to establish delayed assembly plants oriented by logistics and examination of these methods, in: *Proceedings of the 4th Workshop on European Scientific and Industrial Collaboration*, Vol. 4, 2003, 587–594.

[3] Gy. Molnár, Új módszerek a pedagógiai gyakorlatban–az ikt alapú megoldások tükrében, *Szakképzési Szemle* **3** (2011), 170–177.

[4] J. Udvaros, The investigation of oop helper application effects in slovakian secondary schools, *Journal of Logistic–informatics - management* **1** (2016).

[5] J. Udvaros and M. Gubán, Demonstration the class, objects and inheritance concepts by software, *Acta Didactica Napocensia* **9**, no. 1 (2016).

[6] J. Udvaros and M. Gubán, Szlovákiai magyar középiskolások informatika tudásszintjének elemzése az objektumorientált programozás oktatásának kialakítására, *Alkalmazott Tudományok III. Fóruma* **III** (2016), 851–879.

[7] B. Vendler, A gamifikáció vízválasztó éve - a játékosítás vállalati hasznosítása, *HVG Business*, no. 2 (2015), 23–25.

[8] L. Végh and V. Stoffa, Szemléltető animációk a programozásban, in: *INFODIDACT 2010, Informatika Szakmódszertani Konferencia*, (L. Zsakó, ed.), Szombathely, Magyarország, 2011.

JÓZSEF UDVAROS
KODÁLY ZOLTÁN SECONDARY GRAMMAR SCHOOL IN GALANTA
GALANTA, SLOVAKIA

E-mail: udvarosj@atlas.sk

(Received January, 2017)