ACTA DIDACTICA NAPOCENSIA

Volume 9, Number 1, 2016

DEMONSTRATION THE CLASS, OBJECT AND INHERITANCE CONCEPTS BY SOFTWARE

József Udvaros, Miklós Gubán

Abstract: 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. Today's operating systems and development environments apply the principle of OOP; therefore today's developments are inconceivable without the usage of OOP. The knowledge and usage of OOP and the related system development methods are essential mainly for experienced programmers and developers. Consequently, teaching computer science and programming needs to be based on OOP. OOP can be avoided in case of teaching basic computer skills; however, it is an essential part of teaching programming. It is generally known - based on teaching experiences – that teaching traditional sequential programming is facing difficulties, moreover, teaching OOP, that is closer to reality, is even more difficult. The reason for its difficulty is the fact that further theoretical and modelling skills would be required, but the traditional sequential programming principles should be taught as well. So we can ask when it is appropriate to start the OOP education at schools. Let´s take into consideration that to be able to learn programming, algorithmic thinking is essential. It raises further question such as: what method to use for teaching, how to teach students the most effectively and in the shortest period to be able to use this technique.

Key words: OOP, class, object, inheritance, gamification

1. 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. Today's operating systems and development environments apply the principle of OOP; therefore today's developments are inconceivable without the usage of OOP. The knowledge and usage of OOP and the related system development methods are essential mainly for experienced programmers and developers. Consequently, teaching computer science and programming needs to be based on OOP. OOP can be avoided in case of teaching basic computer skills; however, it is an essential part of teaching programming. It is generally known - based on teaching experiences – that teaching traditional sequential programming is facing difficulties, moreover, teaching OOP, that is closer to reality, is even more difficult. The reason for its difficulty is the fact that further theoretical and modelling skills would be required, but the traditional sequential programming principles should be taught as well. So we can ask when it is appropriate to start the OOP education at schools. Let´s take into consideration that to be able to learn programming, algorithmic thinking is essential. It raises further question such as: what method to use for teaching, how to teach students the most effectively and in the shortest period to be able to use this technique.

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. In our research, the level of IT knowledge has been measured at Hungarian secondary vocational schools in Slovakia, focusing mainly on the concepts of programming and OOP knowledge. The result of the survey shows very well that one IT lesson a week for the first year of secondary schools, provided by the national education curriculum for secondary schools in Slovakia, is not enough. The students of vocational schools have 6-8 IT lessons a week

Received May 2016.

from their second grade, which is a substantial difference from the secondary grammar schools. While at secondary grammar schools students have 3-9 IT lessons during their 4years studies, at secondary vocational schools the number of IT lessons are between 20 and 22. We can see from the responses that according to the very few programming lessons at schools the results were as expected, but unfortunately this low level is inadequate.

Our survey focuses on how familiar students are with basic terms of OOP such as: class, object, object-oriented programming, method and inheritance. The results are rather disappointing. Based on the results of our survey we can claim that teaching OOP at Hungarian secondary schools in Slovakia is at very low level. When examining the causes of the low results we have realized that to increase the number of lessons per week is not enough, but also new strategy (methodology), respectively teaching methods need to be developed, focusing on two main elements: gamification and e-learning. Therefore a well-designed application should be invented that is the main purpose of our research. As a further step, we would like a simple application to be used to measure the progress of understanding OOP among the students.

This paper wants to give answers, solutions and methods to the above mentioned two questions. Of course, in this paper it can be presented only theoretical answers, as the proper tool needs to be developed, applied in education and later on needs to be evaluated. Before presenting our ideas we need to examine the preparedness of secondary schools.

2. The preparation of IT survey for secondary schools in Slovakia

The survey was conducted with on-line method, as we wanted to get as many students´ opinions as we can. The most effective way to achieve our goal was an on-line questionnaire. Our request was sent to participate in our research to almost all Hungarian schools in Slovakia. When compiling the survey the issues we weren't only focused on relating to programming, but we wanted to know the students' general computer using habits as well.

The questionnaire was compiled based on several conversations with experts and specialists. The questions were divided into several groups. The first part of the questionnaire contained personal characteristics of the respondents. The second part dealt with the attitude to computer studies, such as how many hours per day they spent with information technology. The third part dealt with the students – how they grade themselves on the basis of their knowledge. The next group of questions focused on the general computer intelligence. The last part of the questionnaire tested the students´ programming and OOP knowledge through specified concepts.

The questions we asked were the following:

What type of school are you visiting?

What is your gender?

What classes are you at?

How long have you been studying IT?

How would you grade yourself based on your overall IT skills?

How many hours a week do you spend in front of the computer at school?

How many hours a week do you spend in front of the computer outside your school?

Have you learnt programming before (at school, by yourself)?

Describe your knowledge of information technology!

If you have studied programming, describe the following terms: program, algorithm, selection, iteration, object-oriented programming, classes, methods, objects and inheritance!

3. Results of the survey

In Slovakia, in the 2015/2016 school year, a total of 10411 students attend secondary schools with Hungarian teaching language (4 years school visiting with final exam at the end). There are two types of schools: 3545 (34.05% of the students) students attend high school (secondary grammar school) and 6866 (65.95% of the students) attend secondary vocational school. From the high school students' 40.85% are boys and 59.15% are girls, while at the vocational high school 57.11% of the students are boys and 42.89% are girls.The respondents consisted of 19 classes of secondary grammar school with Hungarian teaching language, 11 classes of secondary vocational school with Hungarian teaching language, 7 hungarian classes in Slovak high schools and 32 Hungarian classes in Slovak secondary vocational school [6]. The questionnaire was completed by 405 students.

Table 1. *Respondents who have filled the questionnaire*

	High school		Secondary vocational school		
	Boys	Girls	Boys	Girls	Total
1st grade	40	42	0	1	83
2nd grade	34	62	35	5	136
3rd grade	38	56	22	0	116
4th grade	14	18	37	1	70
Total	**126**	**178**	**94**	**7**	**405**

The tables clearly show that the questionnaire was filled out by 83 first-grade students, 136 second-grades, 116 third grades and 70 fourth grade students, including 220 boys and 185 girls.

Based on the responses the questionnaires were filled out by much more students of high schools than students of secondary vocational schools. Moreover, there were only a few girls respondents. One of the reasons could be the proportion of girls attending secondary vocational schools or high schools. Other reason was IT teachers´ low willingness to help. To sum up, according to the number of respondents we could draw general conclusions about the students´ IT knowledge at secondary schools in Slovakia.

The self-evaluation of the students were the first thing examined . The results seemed rather surprising. The 405 students claimed only a very little progress of their own IT knowledge during their four years study. Only a few per cent improvement could be seen from the first to the fourth grade, so unfortunately, there was no linear increase in their knowledge.

Average time spent in front of the computer outside schools was 15.72 hours per week, where boys spent 16.94 hours per week and girls spent 14.28 hours sitting by their PCs at home.

The following graphs represent the general knowledge of the respondents. Data can be associated with the time spent in front of the computer outside school. It can be seen that a very large proportion of students know and use social networks, chat programs and games. They are generally used at home, as at schools these devices are very limited. The Office package and graphic tools are thought at school. The data show that only a quarter of the respondents are familiar with the virtual world or it is not clear to them what virtual world means.

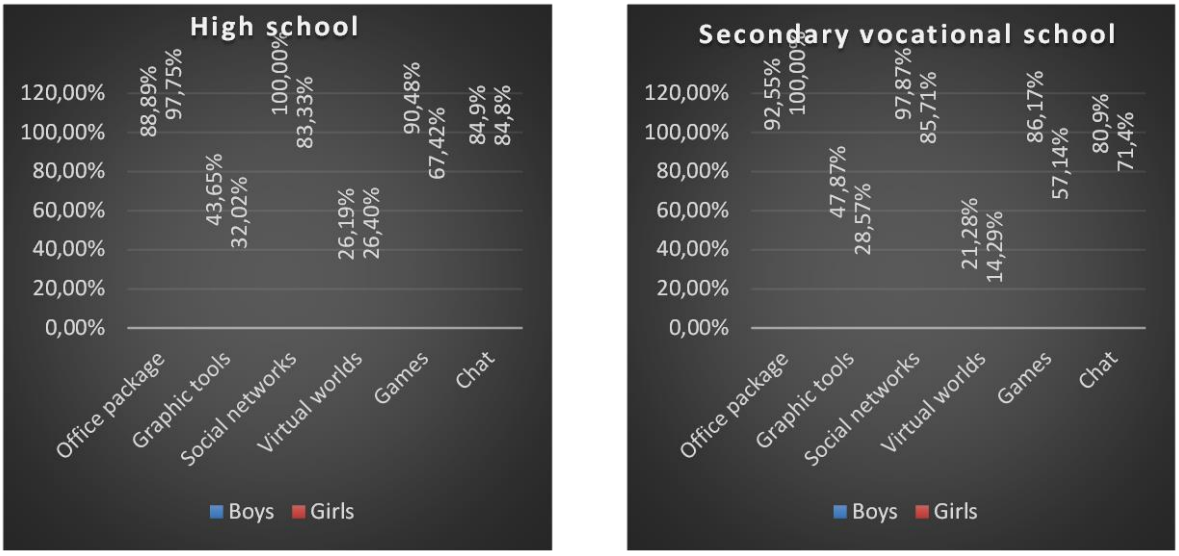

Figure 1. ***Comparison of IT skills by gender***

The last part of the survey was testing respondents´ programming and OOP knowledge through various concepts. We investigated whether the respondents were familiar with the basic concepts of programming and OOP. The survey focused on basic terms, not algorithmic thinking. These questions were answered only by those respondents who had studied programming before. From the 405 respondents, 160 secondary school students learnt programming. The following table reflected the real situation of the secondary schools. In the first grade there was no programming thought. As expected, in the second and the third grade (plus the fourth grade for secondary vocational schools) the number of respondents grew linearly. Surprisingly, there was a significant drop in the numbers of the fourth grade students at high school. The reason for this fact was very difficult to justify.

Table 2. *Respondents who have studied programming*

	High school		Secondary vocational school		
	Boys	Girls	Boys	Boys	Total
1st grade	3	0	0	0	3
2nd grade	8	13	17	4	42
3rd grade	20	23	18	0	61
4th grade	12	14	27	1	54
Total	**43**	**50**	**62**	**5**	**160**

The following table represents the proportion of correct answers by school grades and types.

Table 3. *Respondents who have added correct answers*

High school	Algorithm	Program	Selection	Iteration
1st grade	0,00%	33,33%	0,00%	33,33%
2nd grade	52,38%	42,86%	9,52%	42,86%
3rd grade	51,16%	44,19%	6,98%	41,86%
4th grade	76,92%	73,08%	73,08%	65,38%
Total at high school	56,99%	51,61%	25,81%	48,39%

Secondary vocational school				
1st grade	-	-	-	-
2nd grade	38,10%	28,57%	4,76%	42,86%
3rd grade	22,22%	16,67%	33,33%	44,44%
4th grade	57,14%	64,29%	17,86%	67,86%
Total at secondary vocational school	41,79%	40,30%	17,91%	53,73%
Total	50,63%	46,88%	22,50%	50,63%

When testing some basic programming terms, the first question was to define what algorithm was. No one from the first grade at high school could define this term. Consequently, it was assumed that they had just begun learning programming. The second and third grade performed around 50%. This was acceptable because during the two-year study, they were dealing with programming only for half a school year. The fourth grade showed quite a strong growth. It could be seen that the fourth grade students who answered the question were those who were interested in IT and wanted to graduate from this subject. The vocational high school – where we received much less completed questionnaire – showed stable performance of the students. In the first grade there were 38% correct answers. The rest of the respondents showed variable results.

The percentage of the correct answers for the rest of the technical terms was similar to the answers given to "algorithm". The only exception and the worst results were for explaining the term "selection".

The result was very low especially in the first three grades of high school students. Many students mixed it up with biological selection. The reason could be that the teacher did not use "selection" as a technical term when teaching programming but prefered the word "branching". In addition, it was evidence that students did not use textbooks, or other technical literature.

Based on the responses for defining technical terms it could be drawn a conclusion that the level of OOP knowledge was very low as there were only very few correct answers. The fourth grade students showed outstandingly good results, we believed they were luckily the group of students preparing for final exam. It would have been interesting to see the answers of the students who didn't want to take their final exam from IT. At the secondary vocational school the results were acceptable, and even better than expected because there were not only schools with IT specialization. There would have been interesting to evaluate separately the IT and electrotechnical vocational schools.

For measuring the level of OOP knowledge we examined the following key technical terms: object-oriented program, class, method, object and inheritance. These were the basic terms that students had to understand and differentiate.

Table 4. *Respondents who have added correct answers*

High school	Object-oriented program	Class	Method	Object	Inheritance
1st grade	0,00%	0,00%	0,00%	0,00%	0,00%
2nd grade	9,52%	0,00%	4,76%	9,52%	0,00%
3rd grade	4,65%	2,33%	6,98%	4,65%	4,65%
4th grade	30,77%	26,92%	30,77%	30,77%	23,08%
Total in high school	12,90%	8,60%	12,90%	12,90%	8,60%
Secondary vocational school					
1st grade	-	-	-	-	-
2nd grade	0,00%	19,05%	0,00%	0,00%	0,00%
3rd grade	11,11%	0,00%	5,56%	11,11%	5,56%

4th grade	17,86%	17,86%	32,14%	17,86%	14,29%
Total in secondary vocational school	10,45%	13,43%	14,93%	10,45%	7,46%
Total	11,88%	10,63%	13,75%	11,88%	8,13%

It could be seen from the results that the first grade students of secondary schools did not learn OOP. The second and third grade students had already met with the basics of OOP to a small extent. The terms “inheritance” and “method” was known for about 5% of the respondents. The other two terms:”object oriented program” and “object” were well defined by 5% of high school respondents, and 11% of secondary vocational school students. Surprisingly the term “class” was known by 19% of second grade vocational school students while on the other hand, no one gave correct answer from the high school students. The fourth grade students had extremely good results again, the results were much better than in the whole previous grades. The fourth grade high school students had achieved much better results than their specialized vocational school classmates, except defining the term “method”.

Based on the results of our survey It could be claimed that teaching OOP at Hungarian secondary schools in Slovakia was at very low level. When examining the causes of the poor results we realized that to increase the number of lessons per week was not enough, but also new strategy (methodology), respectively teaching methods needed to be developed, focusing on two main elements: gamification and e-learning.

4. Gamification

A baby has been playing since its birth. Children learn and develop their skills by playing. We can also see that kids acquire the use of computers and any "widgets" very quickly, because they need them for their games. This can also be applicable to high school age group, where other serious strategy games are also played [11, 12].

This has already been confirmed by the survey, as a quarter of the respondents were familiar with and they used different virtual worlds where avatars collected redeemable rewards [10]. These games were more popular among the boys, because nearly 90% of them played such games, while the girl players were only 60%.

This was recognized by scientists, who emphasized using some elements of the games in other areas of life. Education had always been more effective in a playful way. This point of view was expressed in method a so called “gamification”. The concept of gamification was created by a British programmer, Nick Pelling in 2002, while the gamification became popular only around in 2010 [3]. In recent years, several successful applications were known in the fields of economy and education.

Based on the above mentioned facts, the OOP and planning should be carried out using playful elements – by various games. It does not mean that we have to play, but to integrate the well known mechanisms from games to the educational process [7, 8]. Motivation is particularly important. Many people today think that “playful education” means application that can be achieved by different devices (smartphone, computer, etc.) and which support learning and preparation of students. Based on our experience, this method is especially effective in language teaching, but in our view it can be applicable for teaching concepts and techniques of OOP as well.

Figure 2. *The elements of gamification[9]*

5. The role of e-learning in education

The spread of Internet has contributed to a variety of online learning materials, which helps us gain more knowledge. The Internet is not just used for acquisition of knowledge, but can also be used as an information channel.

The results of our survey show that students spend much time sitting by their computer in their spare time and they gain lot of information from the Internet related to their subjects. I have also experienced it in my teaching practise.

Students learn not only with preparation of the needed material, but also with their cooperation, which is done through communication channels. This type of education uses e-learning framework. LMS (Learning Management System) application has been widely spread nowadays. The role of LMS is to identify users and based on their roles and entitlements provide them with the relevant teaching materials (courses) [5]. The LMS servers log the users' data, which is important in terms of learning process. These data can be evaluated later on; moreover it can serve as material for various statistics in the future. Such a widespread system is Moodle. The design and development of Moodle was guided by constructivist pedagogy principles [4]. With their frame system the designers wanted to provide an ideal virtual teaching and learning environment. The experts put an emphasis on Moodle to create wide range of teaching activities.

A "clean" e-learning is commonly used in distance education. Teachers have important role in public education; therefore so called "mixed" e-learning is frequently used. It means that in addition to traditional education, e-learning is used in parallel. This is a perfect form of teaching without any disadvantages.

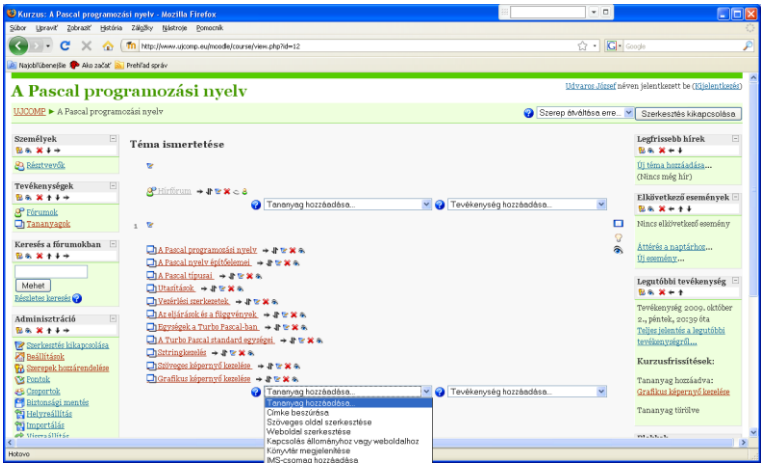

Figure 3. *Moodle in use*

From the perspective of teaching programming, the usage of mixed e-learning is one of the most effective teaching methods.

In our view, the programming cannot be learnt just on IT lessons, as by demonstrating one or two examples on the lessons are not enough to gain experience in programming. Students can get programming skills and algorithmic thinking only with practising a lot and with writing lots of available programs. Therefore it is necessary for students practise programming outside the school as well.

The teacher's task is to ensure the students the needed extra-curricular learning materials, presentations, interactive visualizations and tasks. The easiest way to provide students these materials are through different types of frameworks. They make administration easier, homework checking simplier, and they enables tracking the level of students´ knowledge as well.

We believe that with combining the above mentioned two options, as well as adding an interactive application, an effective teaching method can be created that could be used for teaching OOP. In the next chapter we would like to present a new application that visually illustrates the basic concepts of OOP and their relations. The application is our own development.

6. Presentation of the most important elements of OOP and expectations from the training program

As we can see from the results of the survey that secondary school students´ programming skill level is inadequate. As IT teachers, our goal is to endear the students to programming and get them understand the logic of programming, in order to get more and more information technology experts out of schools. To achieve this goal, different methods are required than in conventional teaching. Using the previous two options, a third device seems to be applicable and it is an OOP teaching support application.

When preparing the OOP sample application we analyzed the main key concepts. These concepts are essential for students to understand and to be able to distinguish.

The first important concept is the term: “class”. From the programming point of view the class appears as a type containing data and methods. When presenting a class it is important to emphasize its abstraction because in reality it does not appear [1].

Except “class” other related technical terms will be introduced such as encapsulation, inheritance, generalization, and polimorphism.

The other important concept is called “object”. An object is a unique individual of a class and each object is unique. It is important for the tuitional program to emphasize this fact and to visualise the difference between the class and the object. This may be well represented in a playful way.

Object-oriented program. Under the OO program we can understand a program, which is actually the sum of objects that communicate with each other, and where each object has its own well-defined task [2]. Our own developed program also presents this fact, as our goal is to provide students with OO programs that they would be able to develop themselves.

Message / request. The objects are told to perform various tasks through messages. It should also be a part of the application.

7. Presentation of the software

A simple application have already been developed for the primary hypotheses. The aim of developing such software is to create a visual environment that easily and playfully introduces the key concepts of OOP programming.

Key features of the program:

- Multilingualism

- Simplicity

- Gamification

- Clarity

- Educational

The application is going to be multilingual for broader testing. The initial window offers a choice of default language environment (English, Hungarian and Slovak).

Introducing the concept of the class using the developed application

After selecting the language environment, TShape class appears in the left side of the window, which is the basic class of a rectangle. It contains name, height and width attributes, moreover methods for shape creation, changing width or height. They are important because one of the characteristic features of the OO can be illustrated here, the combination of data and behavior.

Introducing the concept of the object

On the right side a rectangular object offers testing possibility of each method and the values of the attributes are shown as well.

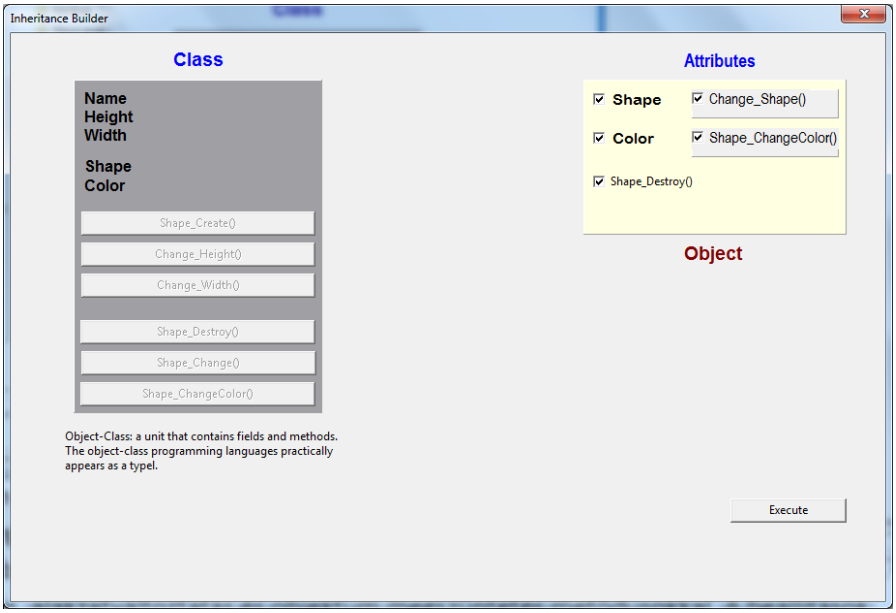

Figure 4. *Demonstration attributes and methods of the class*

Inheritance

Let us introduce the term "inheritance". From the previous TShape basic class we create a different shape (New_Shape) class by inheritance, where there is potential to develop further the derived class. We can expand it with color and shape attributes, respectively with methods of changing colours, changing shapes and destroying objects. After setting the parameters, the object created from derived class can be tested.

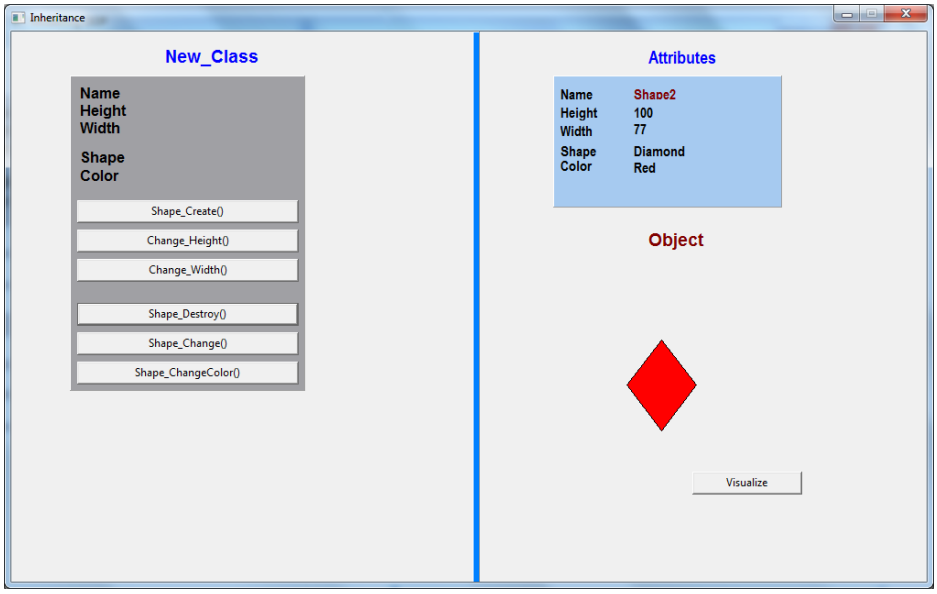

Figure 5. *Demonstration the object*

The program helps to understand the terms as class, object, and inheritance, 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 program will be developed for final testing in a way to illustrate the concepts of message / request and to correspond to the expectations of gamification.

Using the program for testing and final application formulation

The completed program is not final; it is a trial version that serves as a test to verify the hypotheses. The aim of our research is not to create a complex tool, but to show what tool should be developed and for what age group.

The program is primarily intended for secondary schools, where it will be used with a teacher guide.

The effectiveness and applicability of the program must be checked and evaluated in the teaching process. The following two methods will be used to realize and complete our research:

The first method will use a questionnaire focused on respondents´ opinion, where the respondents will be IT teachers at secondary schools, and students, who study programming. We would like to know whether our own developed tool is useful for them. We are also interested in their feelings and experiences of using visual aids for teaching and learning programming. This study examines whether the students find it comfortable according to their level of IT knowledge, therefore the questionnaire will include the students' age and their level of computer application skills. It is known that today's students are very skilled at computer programs, which is especially important in a case of this tutorial. We are eager to know how relevant this application form will be found by them.

The second method will be a knowledge assessment test that we plan to use before and after knowing the program. We are planning to give a course on this subject for students. There would be two groups. While the first group of students would study OOP concepts without visual aids; the second group would use a program as well. Students participating in the knowledge assessment tests would be from different level in order to see in what extent the program supports learning. Therefore, there would be weak, middle and very talented students in both groups. We are planning to carry out a survey about OOP before using the program and at the end of the course. Some practical programming elements will be included as well.

The questionnaire and the knowledge assessment test will be carried out on-line on the internet, electronic form will be sent to teachers and students who are participating in our research. It will be involved secondary schools in Hungary and Slovakia where object oriented programming is thaught, in addition there will be the same secondary schools who had already participated in the first survey.

8. Conclusions

A survey has been prepared for the newly designed teaching approach and to support our hypothesis. In this paper, this survey has been presented this survey, which examined why there were only very few IT experts. Based on our survey it can be seen that the computer science education level at primary and secondary schools, furthermore the number of IT classes is not adequate. In many cases, students are developing themselves with self-study as they do not receive appropriate support at schools. A survey has revealed that the programming knowledge, which is essential in creative IT training, is at very low level, moreover new approaches rarely appear in educational trainings.

The survey confirmed that there is a great need to develop a new teaching method that will help and support Z-Generation with efficient programming and OOP training. We want it to be achieved by combining e-learning, gamification and newly developed application. This development will be realized in the near future.

In addition to the above mentioned goals, we want to show that our newly developed application effectively supports the teaching of OOP using gamification and e-learning. These tools are independently capable of facilitating better understanding, but with combining them together we can get much more effective educational tool that helps students understand important concepts faster and

they can get an insight into the area of modeling as well. It does not mean that the teachers would be dispensable, but it will support students learning both at schools and at home, in addition it will help to train better IT professionals.

Currently, the second main step, the testing, is yet to come, the impact of program and e-learning at secondary schools will be investigated in the near future by us.

References

[1] GradyBooch, Robert Maksimchuk, Michael Engle, Bobbi Young, JimConallen, Kelli Houston (2007):Object-Oriented Analysis and Design with Applications, 3rd Edition, Pearson

[2] Robert Lafore (2002): Object-Oriented Programming in C++, 4th Edition, Pearson

[3] Nick Pelling (2011): The (short) prehistory of gamification, Retrieved on January 10, 2016 from http:// nanodome.wordpress.com/2011/08/09/the-shortprehistory-of-gamification/

[4] György Molnár (2009): Tapasztalatok a Moodle e-learning keretrendszeralkalmazásáról – Az alkalmazott Moodle oktatási keretrendszer leírása, használata, In: Dr. BenedekAndrás (szerk.): „Oktatás-innováció és minőségfejlesztés az IKT stratégiai integrálásával – élenjáró módszerek és gyakorlatok a felsőoktatásban" Konferenciakiadvány, ISBN 978-963-420-980-5, BME, Budapest, 2009. pp. 50–64

[5] György Molnár (2011): Újmódszerek a pedagógiai gyakorlatban – az IKT alapú megoldások tükrében, In: Szakképzési Szemle ISSN 0237-2347, XXVII. évfolyam, 2011. 3. volume, pp. 170–177

[6] Website of Statistic Institute of Education in Slovakia, Retrieved on February 2, 2016 from http://www.uips.sk

[7] Rab, Á. (2012). A gamifikáció lehetőségei a nem üzleti célú felhasználások területén, különös tekintettel a közép-és felsőoktatásra. Retrieved on February 10, 2016 from http://www.oktatas-informatika.hu/2013/03/rab-arpad-a-gamifikacio-lehetosegei-anem-uzleti-celu-felhasznalasok-teruleten-kulonos-tekintettel-a-kozep-esfelsooktatasra/

[8] Kömlődi, F. (2015/02). Életjátékok; Gamifikáció tegnap, ma és holnap. HVG Business, 17-21.

[9] Jennifer Zimbrick (2013): Is Gamification a Positive Learning Trend?, Retrieved on March 11, 2016 from http://www.coetail.com/jzimbrick/2013/12/01/is-gamification-a-positive-learning-trend/

[10] Vendler, B. (2015/02). 2015 A gamifikáció vízválasztó éve - A játékosítás vállalati hasznosítása. HVG Business, 23-25.

[11] Chou, Y.-k. (2013). Yu-kai Chou & Gamification. Gamification in Education: Top 10 Gamification Case Studies that will Change our Future, Retrieved on March 11, 2016 from http://www.yukaichou.com/gamification-examples/top-10-education-gamificationexamples/#.VlB4b9MxjIV

[12] Chou, Y.-k. (2013). Yu-kai Chou & Gamification. Top 10 Marketing Gamification Cases You Won't Forget, Retrieved on March 11, 2016 from http://www.yukaichou.com/gamification-examples/top-10-marketing-gamificationcases-remember/#.VlIGotMxjIX

Authors

József Udvaros, Kodály Zoltán High School, Galanta, Slovakia, e-mail: udvarosj@atlas.sk

Miklós Gubán, college professor, Budapest Business School, CBA Zalaegerszeg, e-mail: Guban.Miklos@uni-bge.hu