VISUALISATION OF OOP ELEMENTS

UDVAROS József, SK

Abstract: Visualization and active learning belong to the main characteristics of didactic teaching principles. Modern interactive digital technologies make it possible to combine all didactic principles into teaching software applications that can significantly increase the effectiveness and efficiency of education. The author of the paper has been teaching programming for many years. He constantly has been creating and using variety of teaching aids to make the teaching-learning process more effective. One of his effective teaching aids is a software application developed for supporting object-oriented programming teaching, which is described in this paper. The author focuses on the visual explanation of the concepts of object-oriented programming such as object, class, method and inheritance.

Key words: Object oriented programming (OOP), visualization, interactive learning, computer aided teaching, graphic presentation of program elements

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 the 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 only 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.

2 Visualization as a didactic principle in learning

The rapidly developing computer science gives the teachers great opportunity to use all the benefits of information and communication technology (ICT) in education process. They can use teaching materials in electronic form, showing animations, simulations and different visualizations. These teaching aids do not only increase the importance of ICT but also mean better and higher quality and efficiency in education.

Goals of visualization:

- It facilitates a particular phenomenon and shortens the educational processes.
- It simulates different processes.
- It enables better and easier interaction between user and computer, which is an important feature of multimedia.

3 Visualization in teaching/learning programming

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 fact, students have problems with understanding programming terminology, especially OOP terminology. Most likely students see only program codes during teachers´ presentations. To make them understand the logic behind it, teachers need to use visual aids. Visual representations can make understanding the terminology of OOP clearer and easier. They will be able to think and speak about classes and its relations. Nowadays only a few visualization programs exist to help understand OOP terminology. Unfortunately these programs use UML representation, which is not taught at secondary schools.

All IT teachers´ goal should be to endear programming to the students 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. A new application need to be developed that supports OOP teaching.

4 Visualization in teaching OOP

When preparing the OOP sample application we analyze the main key concepts (class, object, method, object-oriented program, message/request, inheritance). These concepts are essential for students to understand and to be able to distinguish.

We have already developed a simple application that supports OOP teaching. The aim of developing such software is to create a visual environment that easily and playfully introduces the key concepts of OOP programming.

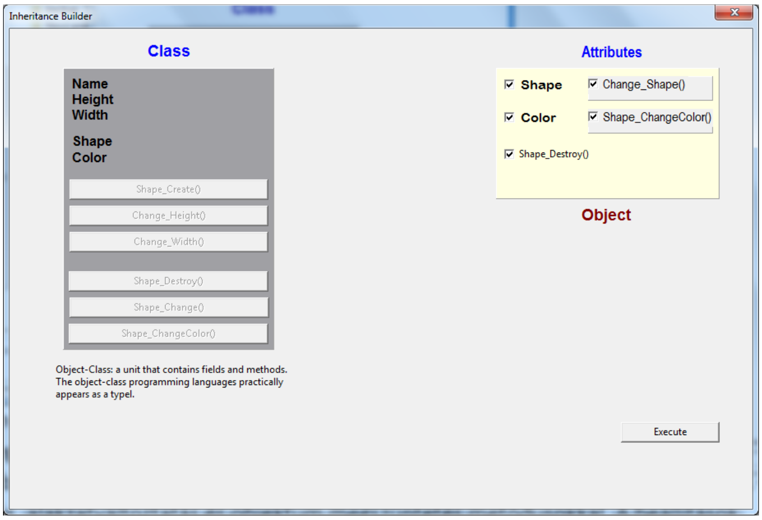
Key features of the program:

- 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 OOP can be illustrated here, the combination of data and behaviour.



☑ Shape
☑ Change_Shape()
☑ Color
☑ Shape_ChangeColor()
☑ Shape_Destroy()

Object-Class: a unit that contains fields and methods. The object-class programming languages practically appears as a typel.

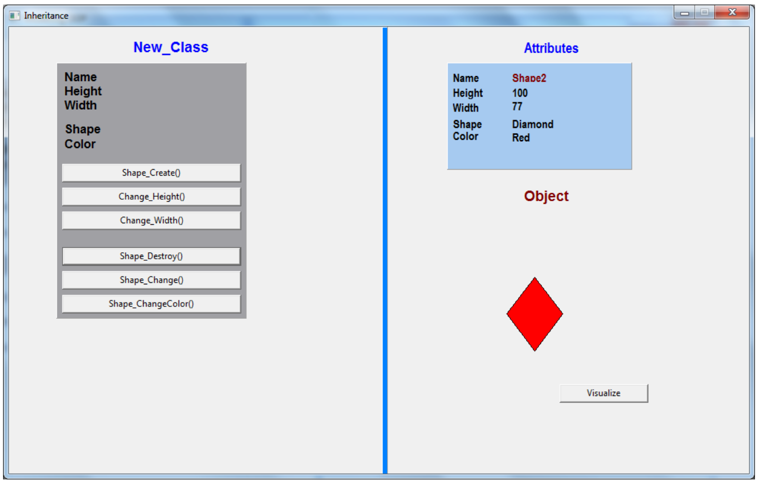
Figure 1: Demonstration attributes and methods of the class

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.

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 colour 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.



Name	Shape2
Height	100
Width	77
Shape	Diamond
Color	Red

Figure 2: 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.

5 Conclusion

The students of secondary schools need simple visual presentations to understanding logic of OOP programming.

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.

Bibliography

[1] STOFFOVÁ, V.:Počítač – univerzálny didaktický prostriedok 1. vyd. Nitra : Fakulta prírodných vied UKF v Nitre, 2003. 172 s. ISBN 80-8050-450-4

[2] STOFFA, V.: Modelling and simulation as a recognizing method in the education, Educational Media International 40 (2), 2004. Taylor and Francis, London.

[3] STOFFA, V. - VÉGH, L.: A programozás tanításának és tanulásának elektronikus támogatása. Komárno :Selye János Egyetem, Eruditio-Educatio, I. évf., 3. szám, 2006/3, 105-113. ISSN 1336-8893.

[4] STOFFA, V.: Azanimációszerepeazelektronikustankönyvekben. (Úlohaanimácií v elektronickýchučebniciach). Információstársadalom 2008, VIII. évfolyam 3. szám, s. 113-125 ISSN 1587-8694

[5] STOFFA, V. - VÉGH, L.: Guided animation of dynamic data structures. In *Third Central European Multimedia and Virtual Reality Conference*. Eger, Hungary, 2006, s. 175-179. ISBN 963-9495-89-1.

[6] STOFFOVÁ, V. – VÉGH, L.: Szemléltetőanimációk a programozásban. In *INFODIDACT 2010, 3. InformatikaSzakmódszertaniKonferencia.* Editor Zsakó,Szombathely, 2010. (príspevokna CD 6 s.)

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

Reviewer:. PaedDr. Krisztina Czakóová, PhD.

Contact address:

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