

Teaching methods of visual programming technology

Beibytgul Koishievna Shayakhmetova¹, Sholpan Embergenovna Omarova², Gali Tulegenovich Omarov²

¹Karaganda State University of E.A. Buketov, Kazakhstan

²Karaganda Economic University of Kazpotrebsoyuz, Universitetskaya 28 street, Karaganda, 100028, Kazakhstan

Abstract. In this article considered the modern direction in programming - it is a visual programming and also here considered the teaching methodology. Also it should be noted that visual programming based on objective-oriented ideas, it is the automated processing programs using special on-line processing. The most common visual programming interface is used to create a software and database management systems. This information technology in the education sphere based on the educational software.

[Shayakhmetova B.K., Omarova S.E., Omarov G.T. **Teaching methods of visual programming technology.** *Life Sci J* 2014;11(5s):359-361] (ISSN:1097-8135). <http://www.lifesciencesite.com>. 72

Keywords: visual programming technology, technology, method, interface, database.

Introduction

Contemporary courses taught at higher education institutions either exclude visual programming or mention it in terms of teaching Delphi environment. We would like to fill in this gap and outline teaching methods in such environments as Delphi, C++Builder and etc. Visual programming has recently become one of the most popular programs for software design [1].

We suppose that it is best to start with presenting the basics of Object Pascal (C⁺⁺). Here it is necessary to mention that visual programming is based on objective oriented ideas, that is why, when talking about unbiased approach, it is important to remember about basic concepts of this new software. Lecturers, who read this course, should remind their students a philosophy of concept. In our case it is encapsulation, polymorphism, and inheritance. Experience has shown that using Object Pascal significantly facilitates formation and development of students' intellectual potential, improving forms and contents of study process, and also introduction of innovative methods in education and allows solving existing problems on a new level. Thus, visual programming involves automated processing of software using special interactive processing [2].

Main part

When examining visual programming systems it is easy to notice that they are all based on objective oriented programming and represent its logical extension. Visual programming is most frequently used to create software interfaces and data base control systems. Such educational information technology relies on educational software, electronic didactical textbooks, teaching materials, and educational simulators which allow shaping educational process effectively and improving it further. In connection with everything stated above it

is necessary to point out new software, which is associated with objective oriented systems i.e. Browser. This software along with a system of clues on the screen enables a programmer to look through some parts of software environment at will and see a completed software project. Here the project means the structure of software: file content, objects and generating classes which make up software as a whole [3].

A lecturer has to be familiar with peculiarities of the software and mention one of its key features and imagine a window with a list of all classes of the system. A lecturer should also remind the students that the idea of objective oriented programming consists in encapsulation (joining) data and processing means (methods) into a type which is called a class. Students frequently ask their lecturer a question: What does a term *object* mean? When answering this question a lecturer should direct students' attention to the fact that an object is a particular variable of a specific class. A lecturer might show directly on a computer that if a class is chosen, special windows display its local functions and variables. Then if a method is chosen, its code is displayed on a separate panel. Also, it is necessary to stress that a system usually includes tools of adding and removing classes from a project. A lecturer should direct students' attention particularly to the fact that Browser software is a rendering tool which is the main integrating tool helping to view the existing system and develop software project documents simultaneously. Lecturers should explain students that Object Pascal and (C⁺⁺) languages have data and commands as structural units but here a visual object, which is called a component, is a structural unit. A component is a type of object which can be transferred (aggregated) into an application from on a special component palette. A component has a set of properties that can be changed without

changing the source program. Analysis has shown that automation of programming is achieved thanks to the option of component transferring onto form (into program) from the component palette and changing its properties without changing source code manually [4].

Then a lecturer should list main characteristics of components: visual and non-visual. Visual components are used to organize interface with a user. They include different buttons, lists, statistical and edited text, pictures and etc. These components are displayed when an application is being run. Non-visual components are responsible for access to system resources: drivers without data, timers and etc. During the process of development they are displayed by their pentagram, but, as a rule, they are invisible when an application is being run. A component can belong to any other component or form. Let us examine the form now: a form is a visual component with the features of a Windows window. During the process of development, necessary components (for example, dialog items) are placed onto forms. An application can have several forms depending on a number of dialog windows opened; also they can be added or removed. Qualification Bachelor of Computer Science and Software is developing in direction of mastering complicated software tools. In this connection software development methods related to a new generation offer interactive tools for solving routine problems allowing creating and inserting completed source code fragments into software in dialog mode [5].

Inference

Students can improve their skills in subject area thanks to their interest in program analysis and analysis of new programming technology. If a large number of software variants is available, such analysis allows bringing to light main patterns of software creation from varied source data and facilitates quick and thorough examination of programming object properties. Technology of application creation in Delphi, (C++Builder) environments has reached a new level. In Delphi, (C++Builder) environments a developer can choose a necessary component, a button for example, from the menu in a component palette and transfer it with the help of the mouse to a designated place on a window of a form in development. During this process the button automatically receives a name (or reference) and it is described in the form module. If a picture of a component is clicked, it becomes active. Then, the boundaries of a button can be moved and a legend (for example OK) and/or a graphical icon on a button, colors and other inferior parameters of a

button can be set in the Inspector window. A double click on a button opens a subprogram template method of a necessary click response type in the source text. In the text editor window a subprogram of click response can be set. Inspector software enables users to enter source texts of methods (subprograms of event processing, named Events), for example by pressing Enter, and also set initial values to data fields, named Properties [6].

Formal use of didactic principles of computerized training without changing pedagogic methods does not produce desired effect, but leads to formal automated training leaving behind system and integral education. Speaking about visual programming technology it is necessary to give it at least a short overview [7].

First steps of visual programming technology are defined by the structure of visual programming environment itself. At first screen forms are created simply by moving the mouse. In the object inspector their properties are set by filling in separate fields. Visual and non-visual components are placed onto the main form. The forms are joined together into single project. Next in accordance with the dialog script the methods of events of the main and subordinate forms are programmed. Programs of empty event methods appear in editor window after pressing corresponding buttons or action of the mouse. Empty methods are supplemented with certain form activation and deactivation mechanisms. After the first steps, a working skeleton of software is formed with data sources form files without data and with generated document forms printed on a printer. The Browser provides visualization of a class hierarchy arrangement of the software skeleton. In other words, technical projects of an implemented part of the software are formed automatically [8].

Further development of the software is carried out by objective oriented programming method. It is possible to implement a part of the software by structural programming method. Some missing visual and non-visual components are formed by modifying the source texts of the closest prototypes of available components. It is recommended to place new components into a component palette. This simplifies their repeated use in current or further developments. A code, related only to current development, is written in program text [9].

Conclusion

Thus, to summarize the above said, we would like to say that as a variant of software creation we propose a new pedagogical educational method, which can be summarized as follows:

Visual programming significantly increases automation of a programmer's work in writing software.

Visual programming is currently one of the most popular programming paradigms. It is based on objective oriented programming approach.

Visual programming environment supports browsers, which help to receive software structure documentation automatically.

The main element in visual programming environment is a component. Components can be visual or non-visual.

Visual programming technology consists in the following: creation of screen forms, placement of visual and non-visual components, programming of events and methods of window forms [10].

Corresponding Author:

Dr. Shayakhmetova Beibytgul Koishievna
Karaganda State University of E.A. Buketov,
Kazakhstan

References

1. Darakhnlidze, P.G. and E.L. Markov, 1996. Visual Programming Environment Borland Delphi-Spb.: BHV-Saint Petersburg, pp. 352
2. Rogerson, D., 1997. Inside COM/Translation from English. Moscow: Department of Russian Reaction, LLC Channel Trading LTD, pp. 376
3. Elistratov, N.N., 2012. Hardware and software technology Multimedia in the educational process. Bulletin of the Ryazan State University, 34: 14-23
4. Rudakov, I.E., 2012. Study of the hierarchical Methods complex discrete structures. Science and education: electronic scientific and technical publication, 6: 17
5. Barzigiarov, P., 2001. Programming model algorithmic structures with massive concurrency. Numerical methods and programming: new computing technology, 2(3): 1-16
6. Kurganskiy, V.M., 2003. Visual Design and debugging of procedural code in the object. Computer applications, 8(1): 52-58
7. Basov, D.V., 2011. Visualization design approaches of complex control systems. Vestnik of Moscow State University of instrument engineering and informatics. Series: Instrumentation and information technology, 32: 96-105
8. Daleka, V.D., A.S. Derevyanko, O.G. Kravec and E.L. Timanovskaya, 2000. Models and data structures. Tutorial. Kharkiv: APRIL, pp. 212
9. Odintsov, I.O., 2002. Professional programming. The system approach. St. Petersburg, pp. 512
10. Trofimova, I., 1999. Treatment and Storage System information. Moscow, pp. 191.

4//1/2014