СЕКЦІЯ «ІНФОРМАТИКА»

УДК 378.147:004.25(07) **L.V.Grishko**

BASIC PROGRAMMING: COMPETENCE APPROACH

This article focuses on miscellaneous approaches in teaching programmers-to-be with the aim to help the latter form key competences. Basic programming competences which students ought to develop while mastering the subject receive their definition. The explanation is given to the choice of structural paradigm of programming and programming language for teaching.

Keywords: *teaching, bssic programming, competence, paradigm of programming and programming languages.*

Introduction

The competition at the modern labor market demands from a professional to quickly learn new technologies and also to easily adapt to new working conditions. Therefore the concept of the education system is connected with the attempt to approximate the final results to the planned ones. This can be realized due to the implementation of the competence approach in the process of training.

The competence approach in its modern shape is specified by ideas of general and personal development characterized by psychological and pedagogical concepts according to which education must be developing and personality-centered. Thus the educational content is defined by a 4-component model: knowledge, skills, creative experience and values [1].

Aim

The aim of the article is to study conditions under which students develop general and special competences of a programmer while mastering the basic programming.

The body of the article

The competence approach includes two basic concepts. The first one comprises a set of interdependent personal values connected with certain objects and processes. The second one is the capacity, i.e. a person has a certain competence including personal attitude to the latter and the object of their activity [2]. Educational competence is understood as a set of knowledge, skills, abilities and a student's experience connected with certain objects of reality, necessary to carry out personal and socially important fruitful activity.

- Educational competences together with the educational content are differentiated by the following levels [1]:
- key competences (realized on the general level, common for all subjects);
- general subject competences (formed on the level of a set of subjects, or an educational branch);
- subject competences (realized within one subject).

There are various ways to form key competences and their classifications. For instance, Russian typology includes general cultural, cognitive, informational, communicational, and social competences of self-improvement. According to OCR certification (developed by

Oxford and Cambridge Certification Boards) there are key competences of several levels necessary for training a high-quality specialist of any professional activity: 1 – communication skills; 2 – numeracy skills; 3 – IT skills; 4 – interaction with other people; 5 – raise of training skills and professional efficiency; 6 – problem-solving skills; 7 – self-development skills [3].

Accordingly the competence approach to basic programming training focuses on the training results. The latter are regarded not as the sum of knowledge received by a student during their training but the ability to employ this knowledge in the process of studies and future professional activity. On the whole the success of a specialist in the future professional and social activity is determined by the level of development of their key competences [4, 5].

Basic programming competences are the following:

General:

- the capacity to practically employ basic programming knowledge, skills and abilities in the process of computer science training, other subjects studying and professional activity in future;
- plan and arrangement of the studying activity;
- information search and analysis of data received from various sources;
- formalization of the knowledge, its accumulation;
- team-work;
- usage of foreign languages in the professional activity;

Special:

- the algorithm making and program writing;
- application and combination of typical algorithms and familiar programming languages;
- fixing and texting of educational programs in the environment of a programming language developing;
- understanding the application areas of basic programming, its employment on other profession-oriented and special subjects, and in future professional activity;
- mastering of new programming languages.

The choice of programming paradigm and the programming language are the problems in basic programming training. There are different programming paradigms representing independent approaches: functional, imperative, object-oriented, declarative etc. These days there are used such computer systems that make it possible to use parallel programming: a program is being made on one computer and is split into tasks in parallel programming. These tasks are done on the virtual multiprocessor computer simultaneously, though in reality only one task can be used at a time. Information torrents are regulated by new technologies – net programming – providing the Internet navigation.

The analysis of higher education teaching plans and programs, educational standards for computer science in the majority of Ukraine's higher learning institutions demonstrates that structural programming is studied as the basic programming. This develops imperative programming.

Structural principles of programming were formulated in the fundamental works of V.M. Hlushkov on the theory of discrete transformations in which the researcher defined the main structures and rules of their compositions and proved fundamental theorem about the possibility to transform the program into a structural form without altering the database the program works with [6]. According to works by E. Deikstra [7] program structural organization meets the demands of program correctness proof and allows changes without complications.

We also view basic programming training on the ground of structural programming due to the following:

- programming is one of the human activities requiring high accuracy and diligence. Instilling these qualities to students who create their own programs is the most complicated task in the process of programming training. It is possible to form these qualities using the structural approach. Implementation of structural organization helps to form and develop algorithmic thinking ([8], [9], [10]). The latter means the sequence of actions and together with logical and artistic thinking determines one's intellectual level and creative potential. Planning skills with the habit to accurately and fully describe one's actions help professional programmers develop miscellaneous problem-solving algorithms.
- the necessity of teaching continuity. Structural programming is traditionally studied in Ukraine in the school programming course, chapter "Algorithm and Programming", in Pascal.

Regarding the choice of programming language for basic programming it is necessary to note that the programming language is the main tool of "communication" of the programmer and the PC. The modern level of IT development demands from the programmer to be able to create software in different languages and even different styles simultaneously. The knowledge of different programming paradigms, and several programming languages, and understanding of main differential features of programming languages enables the programmer to easily master new programming languages and develop together with the programming industry.

The high quality of basic programming training of the programmers-to-be within the course is provided by the opportunity to master two programming languages: Pascal and C.

The experience demonstrates that students should be taught to Pascal programming language first, which was developed by N.Virt "for beginners, how to write programs". Also "…structural programming shaped the descending principle of development (step-by-step decomposition), structuring of logics and data due to simplicity and mathematical ground, and raised the software reliability. Laconic Pascal makes this all possible" [11].

C programming language is chosen as a perspective one for the main practical programming industries:

- C is a universal language, closely connected with the systemic programming which is significant for future software developers;
- The number of programmers using C program language is increasing nowadays;
- syntax and semantics of new programming languages (Java, JavaScript, C#, PHP, etc.) are partially based on the concepts characteristic for C language.
- the knowledge of syntax and semantics of C programming language will become the ground for mastering C++. Its author, B. Straustrup, states that not considering some details C++ is the next one after C" [12];
- C programming language together with C++ and Java, developed on its basis, influence the development of programming in the greatest degree;
- C programming language knowledge can be applied in different practical areas because it has enough operators and the possibility to structure data.

It is necessary to mention that C programming language knowledge will help students to master "object Oriented Programming", "System Programming", "Operation System" etc.

The sequence of (Pascal -> C) training enables:

- to provide similar starting level of knowledge and skills on algorithms among students for studying the main chapters in basic programming;
- improve the knowledge of those students who have already mastered Pascal programming language earlier;
- starting from the first semester solve algorithmically difficult problems.

By using common features in the ruling constructions of Pascal and C programming languages, provide comparative training to these languages. This enables profound mastering of them.

Conclusion

Teaching experience demonstrates that competence approach in basic programming training of future programmers as a part of basic programming methodology facilitates the formation of programmer's general and key competences among students and form professional qualities [13] of a specialist in the sphere of computer science.

Literature

1. Проблемы и перспективы реализации компетентностного подхода в образовании : (Проблемы компетентностного подхода) [Електронний ресурс] / А.Г. Бермус // Интернет-журнал «Эйдос» – Научные исследования. – Режим доступу до журн. : http://www.auditorium.ru
2. Хуторской А.В. Ключевые компетенции и образовательные стандарты // Интернет-журнал "Эйдос". – 2002. – 23 апреля. http://www.eidos.ru/journal/2002/0423.htm
3. Ключевые компетенции 2000. Программа. Уровни 1, 2, 3, 4, 5. KEY SKILLS 87 0005 (OCR. RECOGNISING ACHIEVMENT. – Oxford Cambridge and RSA Examinations).
4. Мозолин В. В. Компетентностный подход в обучении информатике: возможности и проблемы/В. В. Мозолин//Информатика и образование. – 2005. – №2. – С. 115 – 116.
5. Введенский В.Н. Регулятивная компетенция специалиста как проявление регулятивной функции психики личности // Инновации в образовании.- 2014.- №1.- С.127-136.
6. Глушков В.М. Теория автоматов и формальные преобразования микропрограмм / **В.**М. Глушков // Кибернетика, 1965. – № 5. – С. 1–10.
7. Дейкстра Э. Заметки по структурному программированию // Дал У., Дейкстра Э., Хоор К. Структурное программирование. – М.: Мир, 1975. – 247 с.
8. Ершов А.П. Программирование – вторая грамотность http://ershov.iis.nsk.su/russian/second_literacy/article.html
9. Кнут Д. Э. Алгоритмическое мышление и математическое мышление http://www.philosophy.ru/library/math/knut.html
10. Копаєв О. В. Алгоритмічна діяльність та рівень розвитку теоретичного мислення // Вісник Черкаського Національного університету. Серія «Педагогічні науки». Випуск 55. – Черкаси, 2004. – С. 77-81.
11. http://www.osp.ru/pcworld/2001/04/161427 Летопись языков Паскаль // Интернет-журнал «Мир ПК».
12. Страуструп Б. Язык программирования C++, 3-е изд.: Пер. с англ. – СПб.; М.: "Невский диалект" – "Издательство БИНОМ", 1999. – 991 с.
13. Гришко Л. В. Методична система навчання основ програмування майбутніх інженерів-програмістів : дис. на соискание науч. степени канд. пед. наук : 13.00.02 / Л. В. Гришко. – Київ, 2009. – 276 с.

Стаття надійшла _11_._11_._2014___
Прийнято до друку _25_._11_._2014___

Анотація

Л.В. Гришко

Про компетентнісний підхід в процесі навчання основ програмування майбутніх програмістів

У роботі розглядаються різні підходи до формування ключових компетенцій майбутніх інженерів-програмістів. Також визначаються компетентності з основ програмування, які потрібно і можна сформувати у студентів у процесі навчання основ програмування. Обгрунтовується вибір структурної парадигми програмування і вибір відповідної мови програмування для навчання.

Ключові слова: *навчання, основи програмування, компетентність, парадигма програмування, мова програмування*.

Аннотация

Л.В. Гришко

О компетентностном подходе в процессе обучения основам программирования будущих программистов

В работе рассматриваются различные подходы к формированию ключевых компетенций будущих инженеров-программистов. Также определяются компетентности по основам программирования, которые нужно и можно сформировать у студентов в процессе обучения основам программирования. Обосновывается выбор структурной парадигмы программирования и выбор соответствующего языка программирования для обучения.

Ключевые слова: *обучение, основы программирования, компетентность, парадигма программирования, язык программирования.*