# The Possibilities of Virtual Learning Environment Tool Usability for Programming Training

Violeta Jadzgevičienė [1,2] [a] , Jūratė Urbonienė [3]
[1] Lithuanian University of Educational Sciences, Studentų g. 39, LT-08106 Vilnius, Lithuania
[2] Vilnius Business College, Kalvarijų g. 125, LT-08221, Vilnius, Lithuania
[3] Informatics Methodology Department, Institute of Mathematics and Informatics,
Vilnius University, Akademijos g. 4, LT-08663 Vilnius, Lithuania

*Received 27 September 2013, accepted 25 October 2013*

**Abstract.** Learning (teaching) of programming is a sensitive and hot topic for all scientists all over the world. When we learn to program something we have not only to listen to lectures, to analyze supplied and recommended learning material, to listen to teachers advice but to take an active part in learning process too. Virtual learning environment has been more and more integrated into full time study.
The majority of virtual learning environments provide learning material, organization of learning activities, communication and cooperation means to the students but the standard kit does not meet specific learning programming needs. Active researchers, developers and *Moodle* users community has developed and constantly creates new and additional instruments. This paper analyses virtual learning environment tools that are suitable for teaching programming and corresponds effectively to a student's individual learning style. We have used several learning style methods (formulated by Kolb, Honey and Mumford) that are closely interrelated.
Main aims of this work could be formulated as follow: a) teaching (learning) methods of programming that are tailored to student's learning style are defined; b) virtual learning environment tools that help to implement those learning methods are identified. The learning (teaching) methods by using virtual learning environment tools, mentioned above, are experimentally tested on mathematics and informatics students of Lithuanian university of Educational sciences (LUES).

**Citations:** Violeta Jadzgevičienė, Jūratė Urbonienė. The Possibilities of Virtual Learning Environment Tool Usability for Programming Training – *Innovative Infotechnologies for Science, Business and Education,* ISSN 2029-1035 – **1(14)** 2013 – Pp. 3-9.

**Keywords:** Programming training; Virtual learning environments; Methods for teaching of programming.
**Short title:** Virtual Learning Environment.

## Introduction

The majority of Lithuanian universities and colleges apply blended learning in educational process: virtual environment tools (the most popular is *Moodle*) are implemented parallely with traditional lectures and workshops. Such platforms are commonly used to present the learning material, to test students knowledge (self-control and knowledge control for evaluation tests), and for communication with students when they are outside classes. According to various studies blended learning is acceptable for both students and teachers. The experience of using blended learning has shown its strengths and weaknesses. We have used blended learning for teaching basics of programming methods since 2004. Increasingly, it comes about the individualization of learning, tutorial adaptation to a particular learner's needs, abilities and learning goals. It becomes relevant to organize teaching process applying it to the individual style of a learner.

Nowadays, programming training is an urgent and serious challenge both in Lithuania and abroad. This issue has been analysed in scientific articles [1-4] as well as in a lot of dissertations that were defended on this subject. In Lithuania the principles of programming are usually introduced in the fifth grade of a secondary school were teachers introduce the design of programs by using visual programming environments (e.g. *Scratch* etc.). Later on a part of the young people start programming while studying at school, i.e. by choosing a programming module in 10-12th grade, the others start programming training after joining a higher school (university or college), whereas others begin to learn later in their lives, i.e. after starting the career. Usually the learning process is accompanied with a teacher providing theoretical learning material and tasks which have to be programmed by using programming environment. School-age children can start individual programming studies at different extramural schools. In this case learning takes place at weekends with learners ar-

[a]Corresponding author, e-mail: violeta.jadzg@gmail.com

riving to a training location. Teachers provide the students with learning material and assignment, in an electronic version, with an electronic mail used for communication.

In this case the learning results could be improved by using the *Virtual Learning Environments* (VLE) and their tools. Since 2004 programming training in Lithuanian University of Educational Sciences (LEUS) is provided by applying an integrated mode of learning. The learning material provided in virtual learning environment and part of assignments, are transferred to VLE. However, learning to program for a beginner appears to be quite complicated issue at any age level. One must first of all understand program implementation process, be able to solve problems i.e. should change one's way of thinking and develop algorithmic thinking. The main indication of such thinking is capability to put the problem solution into algorithm and to solve the problem in accordance with it (i.e. to get programmed). Therefore, one needs to learn a programming language, i.e. syntax and semantics and to use its structures while writing a program; furthermore, essence and principles of a programming paradigm to be used should also be understood.

Despite the complex nature of programming, which at the same time causes a lot of trouble while learning to program, still the significant influence for the learning success have training tools. A good tool itself should engage a student into a learning process, encourage a learner to achieve his learning objectives and to gain specific knowledge and skills. VLE also contains different tools that can facilitate the programming learning. Therefore, the focus of this article is to analyse what VLE tools can be used for programming training in respect of techniques suitable for a specific learning style of a learner.

Thw object of the study is focused on VLE tools. The goal of the study contains two aspects: a) analysis of programming training characteristics and selection of the most appropriate training (learning) methods for a learning style; b) analysis of the usability of VLE tools currently used to teach programming.

To achieve the goal the following objectives of the study have been raised: 1) an overview of the programming training problems and difficulties encountered in a training process of programming, 2) an overview of classifications of student learning styles and selection of the most suitable training (learning) methods for a student learning style, 3) discovery of methods effective for implementing programming training with usage of VLE tools.

The methods of study that have been used in this research are presented below: research analysis, analysis of virtual learning environment tools, literature analysis on the issue of programming training, a pilot study, a survey.

## 1. The Peculiarity of Programming Training

Programming is a challenging cognitive process. For programming learning first of all it is necessary to learn syntax and semantics of a programming language. The beginners need this in order to start solving tasks. Moreover, apart from learning syntax structures it is more important to be able to apply them to solve a real problem. It is also important to master the techniques and methods of a programming language. While learning the programming techniques, the language is only a mean of expression and application of common programming concepts. Learning programming language also promotes student's thinking skills [5].

Actually, the case of false conception of programming learning is quite frequent, as it is believed that it means learning to put down the task solution in a form of a program text by using structures of a programming language. However, program writing is just one of the programming skills. The ability to read and understand the program text is equally important. Thus a programmer spends some considerable time examining the patterns, i.e. programs written by others [6], and adapting them to the task solving. One might think that while learning to write programs you automatically learn to read them, and to keep track of the program implementation. However, studies have shown that the ability to write a program and the ability to read it has a low correlation [7]. Therefore, during programming learning it should be always kept in mind the importance of developing the reading skills and understanding of programs written by others.

Due to its complexity programming seems to be not very attractive, that is why, in order to engage the learners, it is necessary to present it as easy as possible, in a clear and attractive way. However, no matter how attractively it would be presented, it is not enough to have only knowledge or good patterns, for it is necessary to actively engage oneself in this process, to develop the skills, to think logically and algorithmically. Thus, often skills in creating algorithms (which is an integral part of programming) are implicitly developed in the junior school grades already - by analysing real-life problems, splitting them into smaller tasks, reasoning solution options as well as making synthesis of the results to obtain a general solution to the problem. The acquired thinking skills help to understand the essence of programming and make programming learning more productive [5].

According to the several References [1,6,8-9], the causes that determine programming learning problems have been set out:

i) it is difficult to understand program's objectives and their relationship with the computer;
ii) it is difficult to understand the specific programming language's syntax and semantics;
iii) incorrect understanding of programming constructs;
iv) inability to resolve the problems;
v) inability to read and understand the code of the program.

There are five components leading to difficulties in programming learning: methods of training, learning techniques, learning skills and attitudes, the nature of programming, and psychological reasons - see Ref. [8, 10]. Table 1 represents causes of programming training difficulties.

To find out learners' attitude towards programming training a survey has been carried out, which involved different groups of respondents who have already completed the programming course.

There were 76 students from different study programmes who participated in this research: 29 respondents from *Technologies of Information Systems* study programme, 13 - *Programming Engineer*, 13 - *Informatics*, 12 - *Mathematics and*

Table 1. Causes of programming training difficulties. Adapted according to Ref. [8].

| Component | N | Cause of difficulties |
|---|---|---|
| Methods of training: | 1) | programming training is still not personalized; |
| | 2) | teacher used training methods are not consistent with learning styles of the students; |
| | 3) | dynamic concepts are often taught through static content; |
| | 4) | a teacher is more focused on teaching a programming language and its syntax rather than dealing with task solving through a programming language and environment. |
| The use of learning techniques: | 1) | learners use irrelevant learning techniques or methodology; |
| | 2) | learners work not enough independently to acquire programming expertise. |
| Skills and attitudes of learners: | 1) | learners must have acquired or wish to acquire a wide range of skills related to program development: understanding of problems, knowledge linking to a problem, reflection of a task and its solution, persistence in task solving, application of basic mathematical and logical knowledge, specific knowledge of programming [8]; |
| | 2) | it has been observed that the main difficulty for learners is not to get the result itself, i.e. to write a program, but to go through the development process; |
| | 3) | a lot of beginners improperly use their skills of writing a stepwise specification in a natural language, i.e. they incorrectly transform natural language semantics into a programming language [1]. |
| Programming nature: | 1) | programming requires a high level of abstraction; |
| | 2) | programming language syntax is very complex. |
| Psychological reasons: | 1) | learners are not motivated; |
| | 2) | generally they begin programming learning in a complicated period of their life, e.g. adolescence [9]. |

*Informatics*, and 9 - *Programming and Internet Technologies*. Almost half of respondents i.e. 48.7 % said that programming is difficult to learn, 39.5 % stated that programming is not difficult to learn, while the remaining 11.8 % could not decide. However, opinions differed depending on a future field of work. Those students who will have to program in the future for their career were more opting for the response that programming was not difficult to learn (88.5 %), while those who will not need to program in the future stated that programming was difficult to learn (56 %). Upon request to identify the reasons why they thought it was difficult to learn to program, the respondents were quite self-critical and pointed out that the difficulty was caused by working not enough independently (as stated by 44.7% of respondents) and also the lack of motivation (47.4%). Such reasons as complexity of the syntax of programming languages (30.3 %), difficulty for learners not to get the result (to write a program) but to go through the full development process (36.8 %) also were identified. The survey also determines student needs: learning when the teacher explains individually; communication and cooperation; learning through technology; fast feedback; the need for imagination (visualization).

No matter how the complex programming learning appears to be it is undoubtedly beneficial to a learner himself. Learning to program develops logical thinking and, consequently, this changes the learning style of the students. Scientific research has shown that programming develops the ability to distinguish between the key stages of task or problem solving: analysis, i.e. splitting a task into separate parts; making a plan, i.e. separating each part of a solution; and synthesis, i.e. making separate parts integrated again. The next chapter examines criteria of learning efficiency.

In order to make programming learning easy and attractive, the efforts should be made by both a teacher and a student, as well as right attitude, favourable disposition and close collaboration between them should be established [11]. It is also important to choose the right, i.e. learning-effective, programming tools, a language and environment. Lately, there has been much talk about personal creativity and its development. Often this is being achieved at an early age already. However, the conventional training model used in Lithuania often acts as an inhibitor of human creativity. Programming training is no exception. Here creativity is particularly important because you have to find a solution to a given problem on your own, as well as to implement this solution in a relevant programming environment by means of appropriate tools. Therefore, a combined method of training has being applied, with the help of virtual learning environment enabling the diversification of training tools.

## 2. Correlation of Learning Styles and Training Methods

This chapter provides an overview of methods suitable for programming training and their correlation with a learner's *learning style* (LS).

Programming training in Lithuanian schools was initiated around 1986 mainly focusing on algorithms. Due to the lack of computers usually programming was taught just on a theoretical level and along with mathematics. For a long time reproductive training methods aimed to impart the knowledge have been applied not only to programming training but

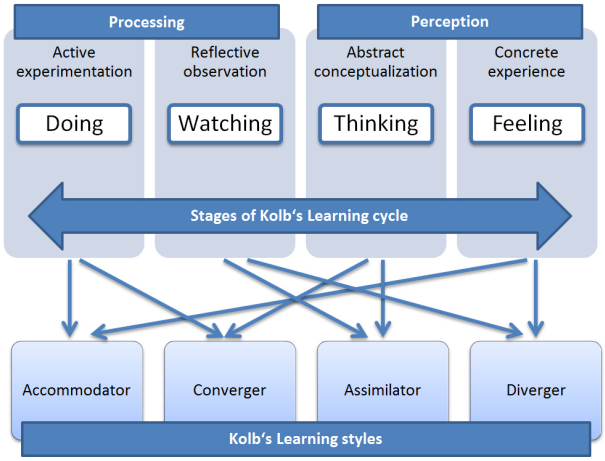


Fig. 1. Cycle of Kolb Learning Styles.
Adapted according to Ref. [15-16].

other subjects as well. In these cases a teacher leads and a student just obeys and performs. A modern teacher should not act as a leader but should facilitate learning by encouraging the learners to work actively and thus acquire new knowledge.

Mayes and Fowler as the E-learning researchers refer to a learning cycle with a feedback, in which three stages are distinguished: *Conceptualization - Design - Dialogue* [12]. Each of these stages is related by the authors to a corresponding training theory - *cognitivism*, *constructivism* and *social constructivism*.

First year students enrolled in *Programming basics and techniques* course (*Mathematics and Informatics* study programme at LUES) are taught using combined method of training, i.e. lectures, practical work with some part of learning activities being incorporated into a virtual learning environment. Training is built on the basis of the above mentioned cycle, by selecting the appropriate training methods for each stage.

At a conceptualization stage a new learning material is interpreted by relating it to an existing experience of the learners. New concepts are introduced in the following ways.

1. Descriptions of new concepts and definitions are provided by a teacher. In this case, the most commonly used informative training methods are the following: a lecture, demonstration, visualization of algorithms, pattern analysis (program code reading and interpretation).
2. Learners independently ascertain the value of the concept from the training material content provided in a VLE. Then, such methods of training as an involving lecture, brainstorming, a concept map, structured notes, visualization of algorithms, discussions, pattern analysis (program code reading and interpretation) are used.

In a design stage, the new concepts are established and customized to help solving a real problems. At this stage the following practical creative work (exercises) is carried out: writing the program code - individually or in groups, by using intuitive, trial-and-error, step-by-step and other methods, also algorithm block diagrams are drawn.

At a dialogue stage knowledge is solidified by experience sharing, knowledge testing and evaluation. Group work methods and discussions are used with evaluation of individual and group work, together with testing and individual practical work performed.

Learning success depends on how the maximum learning goals are being achieved, i.e. whether the necessary knowledge and skills are being acquired, and what emotions are experienced by learners during the learning process. Learning success is to a large extent determined by learning efficiency, which depends on the willingness to learn and knowledge how to learn. It is also influenced by an attractive learning environment.

However, regardless of attractive learning environment good learning results are still determined by learner's personal qualities and his LS. Big number of LS classifications is presented in Ref. [13]. According to Ref. [14], the aspect of student activity is very important in the process of programming learning. Kolb distinguishes four phases of learning through experience process. Fig. 1 represents cycles of learning style:
*Concrete Experience* (for example, using the cycle to calculate the sum of the sequence);
*Reflective Observation* (for example, discussing the experience, i.e. what is easy, what is difficult, what effects the results);
*Abstract Conceptualization* (after discussion of experience an action plan on how to continue the work is developed);
*Active Experimentation* (testing theories in practice).

In Kolb theory, learning cycles representatives of four styles can be met, which are identified by Honey and Mumford using several definitions as presented in Fig. 2:
*Activist* - prefers doing and experiencing;
*Reflector* - observes and reflects;
*Theorist* - wants to understand underlying reasons, relations;
*Pragmatist* - likes hands on.
The LS affects the learner's inherent qualities and his learning activities, learning techniques, the issues raised, his communication and ability to generalize.

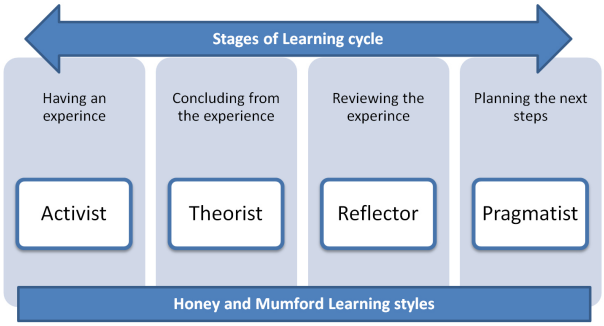


Fig. 2. Honey and Mumford Learning Styles.
Adapted according to Ref. [14-16].

Table 2. Teaching (learning) methods appropriate for the LS. Adapted according to Ref. [14].

| Learning style by Kolb - *Diverger*<br>Learning style by Honey and Mumford - *Reflector* | | |
|---|---|---|
| Features | Most appropriate methods | Less appropriate methods |
| prefer concrete experience and reflexive observation; have the ability to imagine, to look at specific situations from various perspectives; broad imagination, vision anticipation, improvisation situations requiring find ideas; willing to communicate with people interested in human relationships, there are emotional; | brainstorming; concept mapping; cooperative groups; demonstrations of examples using visualization tools; reference study; self testing; discussions; case studies. | lectures may be helpful if they provide expert explanations and analysis. |

| Learning style by Kolb - *Assimilator*<br>Learning style by Honey and Mumford - *Theorist* | | |
|---|---|---|
| Features | Most appropriate methods | Less appropriate methods |
| prefer reflexive observation and abstract conceptualization; easily creates theoretical models share the information and abstract elements of the whole situation; tend to make inductive conclusions, summarize the results; prefer to work with abstract ideas and concepts than with the people; | lectures; demonstrations; readings; case studies; different observations; individual work; working in step-by-step manner; design of analogies, systems, models, theories, and projects. | talking with experts is normally not helpful (discussions, feedback); group work with activists. |

| Learning style by Kolb - *Converger*<br>Learning style by Honey and Mumford - *Pragmatist* | | |
|---|---|---|
| Features | Most appropriate methods | Less appropriate methods |
| transform theory to prepare for a particular activity; develop ideas; tend to make hypothetical deductive conclusions interested in things and theories (like the check), the specific problems; little interest in people; are not emotional. | practical application of ideas; problem solving; feedback; self testing; example analysis; visualization of algorithms; decision-making; solving technical problems over interpersonal issues; applying new learning to actual practice to see if they work; field work; observations; coaching; simulations. | group work; not evaluated work. |

| Learning style by Kolb - *Accommodator*<br>Learning style by Honey and Mumford - *Activist* | | |
|---|---|---|
| Features | Most appropriate methods | Less appropriate methods |
| distinguishing feature - the need to act, to do, to implement; organization of the activities, coping with situations requiring rapid adaptation to changing circumstances; relies more on intuition than logic, tend to solve problems intuitively, through trial and error; use theories created by others and results of analysis, and therefore prefer to work in teams; most likely to take risks; pay to communicate with people, but sometimes it may seem impatient, demanding. | problem solving in an intuitive and trial-and-error manner; challenges of new experiences, involvement with others, assimilation; to do anything new; problem solving; small group discussions; group work; role-playing; gaming. | individual works; non-interactive lectures; working in step-by-step manner; reference study. |

In planning the training more suitable training (learning) methods can be identified with regard to the classification of LSs and characteristics of a LS. Classification of methods based on LS is presented in Table 2.

If there is a purpose to organise the learning based on LS, at the beginning of learning activities the student's LS is determined by using a special test.

However, it happens that the LS changes in the process of learning, therefore, the LS should be diagnosed not once. Both, a teacher and a learner, taking into account the inherent style characteristics can plan their own learning activities.

In terms of a certain learner the above described classification of learning methods according to LS is not absolute.

Table 3. VLE's tools suitable to teaching (learning) methods.

| Teaching (learning) method | VLE's tools |
|---|---|
| brainstorming | blog, chat, questionnaire, feedback, forum, glossary, mindmap, messages, wiki |
| cooperative groups | assignments, blog, calendar, chat, database, feedback, forum, glossary, IMS content package, lesson, messages, mindmap, url, wiki, workshop, VPL |
| demonstrations | assignments, file, page, lesson; workshop, url, VPL |
| Discussions, small group discussion | chat, forum, messages, blog, glossary, choice, workshop |
| example analysis | assignments, blog, database, file, forum, glossary, hotpot, lesson, page, quiz, survey, url, wiki, VPL |
| group work | blog, calendar, chat, database, folder, forum, glossary, lesson, messages, page, url, wiki, workshop, VPL |
| individual work | assignments, feedback, file, glossary, label, lesson, mindmap, page, quiz, url, VPL |
| lectures | blog, questionnaire, feedback, file, folder, glossary, hotpot, label, lesson, page, url, wiki |
| problem solving | assignments, blog, calendar, chat, choice, database, feedback, file, forum, glossary, lesson, messages, url, wiki, workshop, VPL |
| reference study | assignments, blog, database, file, lesson, page, url |
| self testing | questionnaire, hotpot, mindmap, quiz, survey, VPL |
| visualization of algorithms | assignments, blog, database, file, forum, glossary, lesson, url, wiki |

Taking into account personal qualities of a certain learner, he might like to use those methods which, in accordance with his identified LS, are not appropriate for him. Also, the learner is usually characterized by several LS, with one of them to be more expressed. The attractiveness of the method can be determined by an ingenuity of the teacher.

In the process of studying while using a combined learning techniques there appear more possibilities to personalize the training according to a LS. Methods the most acceptable for a learner can be selected by a teacher or they can be selected by a learner himself. The next chapter will explore VLE tools to help you adapt the above mentioned methods of training.

## 3. Study of the VLE Tools Usability for Programming

The most popular VLE in Lithuania is *Moodle*. It is used both in many institutions and secondary schools.

In *Moodle*, a training course developer is provided with a large set of standard tools allowing to diversify the methods used. Active researchers, developers and *Moodle* users community has developed and constantly creates new and additional instrumentality. A team of teachers at the University of Las Palmas de Gran Canaria, Spain in collaboration with Innovation Center for Information Society has developed *Virtual Programming Lab* (VPL). VPL is an activity module for *Moodle* that manage programming assignments.

The VLE tools can be used at all stages of learning cycle described by Mayes and Fowler in Ref. [12], i.e. in conceptualism, introducing concepts and their relations, as well as at a design stage, where practical tasks are solved, and at a dialogue stage, with using the tools of communication and collaboration between a teacher and students, as well as between peers. Teaching (learning) methods appropriate *Moodle* tools can be seen in Table 3.

For interactive training material presentation such *Moodle* VLE tools as pages, a file, a glossary were used in teaching programming basics and techniques at the LUES. In the process of training (learning) active training methods (an involving lecture, brainstorming, concept maps, etc.) were used. For implementation of these methods such VLE tools as a blog, a chat, a feedback, a forum, a glossary, a mindmap, wiki were used.

Individual assignments were given in VLE. Students performed their programming assignments in an appropriate programming environment and submitted the result of their work to a teacher in VLE. For this an assignment tool was used. VLE tool VPL for programming training was also tested, however in order to evaluate its efficiency it would require more scientific investigation.

During 2011-2012 academic year a pilot study on usability of group work methods in programming training was carried out in the LUES. The study enrolled 17 students (study programme of *Mathematics and Informatics* at LUES). All students who took part in the study have not yet tested the method of group work in programming learning. The use of this method was evaluated positively. Fig. 3 represents results of student's survey.

For implementation of the group work method the VLE tools, such as a glossary, a mindmap, wiki, were also used. During task performing such VLE communication tools as a forum, a chat, messaging were used for discussions. When assessing tasks performed in groups it was observed that VLE communication tools were rarely used by students.

Participants involved in the pilot study were asked to evaluate VLE communication tools. In comparing virtual and *face to face* communication the students indicated that while performing the task they preferred *face to face* communication, which was "faster, easier, take less time than while you write what you would like to say."

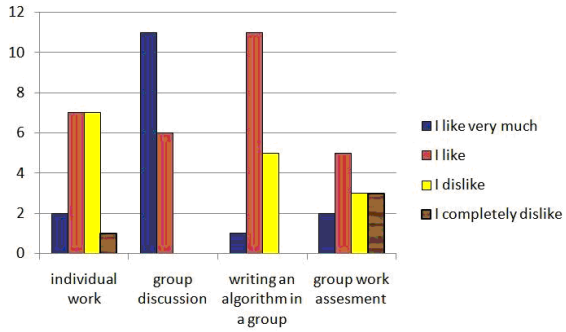


Fig. 3. Results of student's survey.

The analysis of frequency of VLE communication tool usability has shown that students tend to give the priority to a messaging tool.

## Conclusion

Programming learning is a complex process; therefore it is necessary to take into account the needs of students. The survey has clarified the following needs of the learners: learning when the teacher explains individually; communication and collaboration; learning through technology; fast feedback; need for imagination (visualization).

To achieve the objectives of his subject a teacher can choose between training methods or make their combinations, use different tools and technologies, and create an authentic style of work. However, it should be always taken into account the actual situation, i.e. the student's age, needs, capabilities, his learning style, the teacher's skills and qualities himself, the objectives of the subject. The study involved the classification of learning techniques according to learning styles of a learner. You can get the training (learning) process more personalized, when you identify the learning style and know the techniques suitable for that style. To assess the effectiveness of this classification requires an in-depth investigation, which will be carried out in the next academic year.

A number of *Moodle* VLE tools are suitable, and can be used, for enriching the programming training techniques. The pilot study carried out with students allowed to verify the appropriateness of the use of selected tools. Although the students have specified their need to communicate using technologies, however, when performing the task, to save the time, they rarely used chats, forums or messaging.

## References

1. Bennedsen, J. and Caspersen, M. E. Programming in Context: a Model-First Approach to CS1. – *SIGCSE '04*: Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education, Norfolk, Virginia, United States (2004) 477 – 481.
2. Bennedsen, J., Caspersen, M. E. Revealing the Programming Process. – *SIGCSE '05*: Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education, St. Louis, Missouri, United States (2005) 186 – 190.
3. Kinnunen, P., Simon B. Experiencing programming assignments in CS1: the emotional toll. – *ICER 2010* 77-86.
4. Kinnunen, P., Simon B. CS majors' self-efficacy perceptions in CS1: results in light of social cognitive theory. – *ICER 2011* 19-26.
5. Mayer, R. E.; Dyck, J. L. and Vilberg W. Learning to Program and Learning to Think: What's the Connection? – *Communications of the ACM* 29(7) (1986) 605-610.
6. Mannila, L. Novices - Progress in Introductory Programming Courses. – *Informatics in Education* 6(1) (2007) 139-152.
7. Winslow, L. E. Programming pedagogy, a psychological overview. – *ACM SIGCSE Bull* 28(3) (1996) 17-22.
8. Gomes, A.; Carmo, L., Bigotte, E. and Mendes, A. Mathematics and programming problem solving, – In: Proceedings of the 3rd E-Learning Conference *Computer Science Education* September 2006, Coimbra, Portugal.
9. Jenkins, T. On the Difficulty of Learning to Program. – In: Proceedings of 3rd annual conference of the LTSN-ICS, Loughborough University, United Kingdom, August 2002, pp. 53-58.
10. Dagienė, V. and Urbonienė, J. Learning Programming: Comparative Analysis of Languages and Environments. – *Information Sciences* ISSN 1392-0561, 54 (2010) 44-62.
11. Juškevičienė, A. and Urbonienė, J. The Research on the Web 2.0 Technologies Adaptation for Teaching Programming. – *Studies in Modern Society* ISSN 2029-431X, 2(1) (2011) 161-170.
12. Mayes, J. T. and Fowler, C. J. H. Learning technology and usability: a framework for understanding courseware. – *Interacting with Computers* 11 (1999) 485-497.
13. Hawk, T. F. and Shah, A. J. Using Learning Style Instruments to Enhance Student Learning. Decision Sciences – *Journal of Innovative Education* 5(1) (2007) 1-19.
14. Honey, P. and Mumford, A. (1999). Learning styles questionnaire. – <http://www.psychpress.com.au/psychometric/products/others/LSQ%20Report%20%28Sample%29.pdf>, accessed 2013 05 30.
15. Kolb, D. A. Experiential learning: Experiencies as the source of learning and development. – Englewood Cliffs, N.J.: Prentice-Hall. 1984.
16. Kolb, A. Y., & Kolb, D. A. The Kolb Learning Style Inventory- Version 3.1, technical specification. – Boston: Hay Group. 2005.