

Introductory Programming Training of First Year Students at Riga Technical University

Natalya Prokofyeva, Marina Uhanova, Sabina Katalnikova, Oksana Zavyalova and Aleksejs Jurenoks

Faculty of Computer Science and Information Technology, Riga Technical University, Riga, Latvia

{natalija.prokofjeva, marina.uhanova, sabina.katalnikova, oksana.zavjalova, aleksejs.jurenoks}@rtu.lv

Abstract. In this paper, solution to the problem of introductory programming education at Riga Technical University is considered. At the Faculty of Computer Science and Information Technology, the main courses of student training are courses in programming languages and environments. Their contents, forms and methods of teaching must correspond to the contemporary state of programming languages, methods and technologies, as well as to the perspectives of their development. The paper provides a description of changes in content and method of teaching the course “Algorithmization and Programming of Solutions” offered to first year students at the Faculty of Computer Science and Information Technology.

Keywords: introductory programming training, Pascal, Java, example-based method

1 Introduction

Along with the ever-accelerating scientific and technical progress, continuous growth of knowledge and change in its content occurs. This obliges the specialists constantly to master new knowledge, skills and abilities and imposes new demands on their training. At present, it is very important to implement a technique of teaching that would assure informational and methodical support of the learning process. As is well known, every study course includes three components: basic (theoretical) part (i. e. ideas, knowledge); laboratory practical part (abilities, experience); examination part (control of abilities and skills). All these parts are obligatory and traditionally employed by teachers in all higher education establishments. Thus, in creating (projecting) a course the following tasks are to be solved:

- Elaboration of a detailed module structure of the course and scenarios of learning;
- Creation of quality methodical materials;
- Choice of forms and techniques of knowledge control, development of knowledge testing materials;
- Selection and structuring of accessory materials.

The problem of increasing the quality of study courses has been considered by the authors of this paper elsewhere [1]. It was established that balanced supplementation of study courses with new content elements not only increases performance of the students, but also to a great extent increases the quality of teaching.

At the Faculty of Computer Science and Information Technology the main courses of student training are courses in programming languages and environments. Their contents, forms and methods of teaching must correspond to the contemporary state of programming languages, methods and technologies and the perspectives of their development. However, the staff teaching these courses faces a host of problems to be solved in order to ensure quality training of future specialists in information technologies.

In this paper, solution to the problem of introductory programming education at Riga Technical University is considered. Changes in content and method of teaching of the course “Algorithmization and Programming of Solutions” taught to first year students at the Faculty of Computer Science and Information Technology are described.

2 The Problem of Choice of the Initial Programming Language

When speaking of first year students, one of the main peculiarities of this audience is almost complete absence of basic training in programming. Undeniably, among the students enrolling into the university there is a small percentage of those having basic programming skills. But the inevitable conclusion imposes itself: one should count on the average level of the students, which means that our target audience has no programming skills whatsoever and teaching has to be started really from zero.

Every year in September a testing of first year students is performed with the aim of determining the level of initial training of the students. The results show that in the academic year 2015/2016 only 18% of first year students enrolled at the Faculty of Computer Science and Information Technology are familiar with the basics of programming. In order to even out the level of training of the first year students, those students who have not completed the test successfully are given the opportunity to attend the course “Introduction to computers and algorithms”.

Thus, the level of programming training of first year students is extremely low. This is what we have on the “entrance”. “On the exit”, or on completing their higher education, the student must have mastered knowledge, abilities and practical experience in object-oriented, logical and WEB programming. Disregarding the latter two, we shall dwell in more detail on teaching “classical” programming to first year students. It is precisely the subject matter taught during the first semester that lays the basis for students’ programming style and programmer’s thinking.

It is possible to single out the following problems:

- Choice of initial programming language and languages for subsequent training;
- Selection of study content;
- Selection of teaching methods, forms and means, including those based on information technologies;
- Realization of inter-subject and intra-subject connections etc.

Traditionally, programming training presupposes first mastering structural programming (Pascal), and then object-oriented programming (Java, C++). Analysis of papers devoted to this problem shows that most educational establishments hold to this approach [2, 3].

At the Faculty of Computer Science and Information Technology of Riga Technical University the basic course in programming training is "Algorithmization and Programming of Solutions". This course presupposes familiarization with the principles of algorithmization of computational processes and program creation.

Introductory training at RTU is also based on structural programming, i. e. on teaching Pascal. The main argument for this is the fact that teaching object-oriented programming without prior knowledge in structural programming is extremely complicated if not impossible. It should be noted that most university teachers belong to older generation. As a rule, they have gained immense experience in using the techniques of structural programming based on algorithmic decomposition of big systems. However, using out-of-date instruments dims students' enthusiasm and can form arrogant attitudes toward programming courses as waste of time. Training specialists demands new approaches able to endow it with methodological meaningfulness and contemporary fulfilment adequate to existing reality and standards of the new generation.

Already Edsger W. Dijkstra in his book *A Discipline of Programming* noted that "A most important, but also a most elusive, aspect of any tool is its influence on the habits of those who train themselves in its use. If the tool is a programming language, this influence is – whether we like it or not – an influence on our thinking habits" [4, viii]. That is why the problem of choice of programming language, selection of content and adequate teaching methods of future IT-specialists is very relevant at the present time.

Today's students will start their professional activity in 3–4 years after taking the course in programming. In this lapse of time the industry will have advanced in terms of languages, methods and technologies. That is why it is not appropriate to teach something that is already today obsolete in information technologies. Object-oriented approach should substitute the structural one, the former currently being the basic paradigm used in programming industry. It is also most widely used for internet application programming (developing web-services, applets, dynamic content etc.).

Thus, object-oriented language should be the first programming language for a student. It can be either Object Pascal (with its realization in the visual programming environment Delphi or free environment Lazarus) or Java – a wholly object-oriented language allowing to develop platform-independent application (for developing Java programs it is possible to use the integrated development environment Eclipse).

Directors of study programs at the Faculty of Computer Science and Information Technology, as well as the representatives of students' parliament have recommended Java as the initial programming language. The advantages of this choice are the possibility of comparative analysis of different programming languages and of choosing optimal set of instruments for problem solving, as well as development of algorithmic and logical thinking of students.

Thus, starting with 2015/2016 academic year Java has been selected for programming training of first year students, while sophomore students as before first learn C and then C++.

3 Method of Teaching JAVA at Riga Technical University

University teachers are facing a problem: how to teach students not having any programming experience a language as serious as Java? One of the solutions was to apply an example-based method.

The course "Algorithmization and Programming of Solutions" comprises 48 hours of lectures and 32 hours of practical tasks. The first part of the course includes topics like linear, cyclic and branched algorithms, variable and elementary data types, arrays, operators and operations, methods. During the second part of the course students master the following topics: basics of object-oriented programming, line and symbol processing, task with files (streams); task with file system.

Example based method includes the following: during the lectures a particular topic is expounded and then a practical task is offered to students. Solving the task includes working out an algorithm (a structural scheme); analysis of the tools offered by the programming language that are necessary for realizing the algorithm; example of a solution (program code). Then students are asked to solve a one-type task. If students have questions during executing the task teacher helps them by explaining where a mistake might have occurred. Then popular mistakes are considered and finally the right solution of a task is demonstrated. Thus, many practical tasks are handled during the lectures.

In this course, several practical tasks are envisaged. Organization of practical tasks takes place in the following way. Student must develop an algorithm, write a program and submit it electronically to the study portal ORTUS [5]. When the program is evaluated the student must defend his task, i. e. write a report and answer teacher's questions about the program and the task in general. The first part of the course includes five, the second one – three laboratory tasks (see Table 1).

Table 1. Laboratory tasks on *Java* programming language

First semester			
No.	Title	Definition of the task	Constructions in Java
1	Branched programs	Enter coordinates of points x and y. Determine, to which color field does this point belong.	Input-output operations. Announcement of variables. Conditional operator.
2	Development of a simple cyclical program	Calculate the trajectory of bullet flight based on given parameters. Display on the screen a table containing time and coordinates of the bullet flight.	Cycle operators. Use of mathematical functions.
3	Processing one-dimensional arrays	Create a one-dimensional array and fill it in with values. According to the option of the task modify elements of the array. Display the elements on the screen before and after modification.	Arrays, break and continue operators.

First semester			
No.	Title	Definition of the task	Constructions in Java
4	Processing two-dimensional arrays	Create a two-dimensional array containing results of a sports competition. Process the results in a particular way depending on the option of the task.	Multidimensional arrays. Nested loops. Generation of random numbers.
5	Ways of organization of nested loops	Develop a program filling in a two-dimensional array with numbers based on nested loops.	Techniques of nested loop organization
Second semester			
No.	Title	Definition of the task	Constructions in Java
1	Sorting arrays	Develop a program that would sort elements of the array in the ascending order in two different methods.	Methods. Recursion.
2	Lines and text files	Develop a program modifying the content of a text document in a particular way.	Java libraries. Processing lines. Classes of input-output.
3	Creation of a file processing system	Create a simple information system.	Basics of object-oriented programming

Laboratory tasks are devoted to mastering input/output of information in Java, to organizing of branched algorithms by using a conditional operator, to using different operators of cycle organization while working with arrays, as well as lines and use of files. Descriptions of several laboratory tasks are shown below.

3.1 Laboratory Task “Branched Programs”

Task

1. Develop an algorithm and write a program that would analyze the color of the field in which a particular point is located based on the x and y coordinates of the point.
2. Write a report on the task including: definition of the task, analysis of solution, description of the algorithm, program code, testing example.

Requirements

- Output of information about the author of the program (name, surname, group number, student ID number);
- Input of point coordinates (x and y – real numbers);
- Output of messages “red”, “blue”, “green” or “white” depending on the field to which the point (x, y) belongs;
- Make provision for output of messages in response to mistaken action on the part of the user.

Task options

- Task options correspond to the three last numbers of a student’s ID (e. g. if the ID number is 123RDB456, the option is determined by the numbers 4, 5 and 6);

- The program must be uploaded to the ORTUS environment as a source file (*.java) and a compiled file (*.class).
- 500 options of this laboratory task have been developed (some of them are seen on Figure 1).

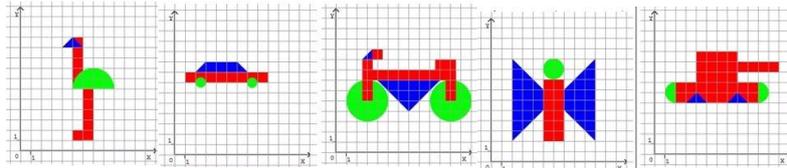


Fig. 1. Examples of laboratory task

3.2 Laboratory Task “Development of a Simple Cyclical Program”

Task

1. Develop an algorithm and write a program that would calculate the trajectory of a bullet flight based on given parameters and conditions. The trajectory must be determined prior to reaching the goal or ground.
2. Write a report on the task including: definition of the task, analysis of solution, description of the algorithm, program code, testing example.

Requirements

- Output of information about the author of the program (name, surname, group number, student ID number);
- Input of the value of the initial speed of the bullet v_0 and the angle of flight α (v_0 and α – real numbers);
- Output of the bullet’s trajectory as a table (time, coordinates of the bullet x and y);
- Output of the message “the target was destroyed” if the bullet has reached its goal and the message “shot off the target” in the opposite case;
- Make provision for output of messages in response to mistaken actions on the part of the user.

Task option

- Task option correspond to the three last numbers of a student’s ID (e. g. if the ID number is 123RDB456, the option is determined by the numbers 4, 5 and 6);
- The program must be uploaded to the ORTUS environment as a source file (*.java) and a compiled file (*.class).
- 400 options of this laboratory task have been developed (some of them are seen on Figure 2).

For each laboratory task solution examples have been worked out including descriptions of algorithms as block-diagrams, Java source code, as well as a set of tests for testing the program.

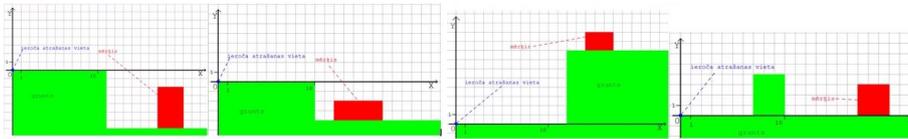


Fig. 2. Example of laboratory task

However, as the student flow is very big (16 groups 30 people each) the problem was to evaluate all these laboratory tasks. For this purpose, a testing program was developed including requirements for each laboratory tasks and special tests were created. The task submission process comprised two parts. Mark for the program itself (based on automatic check results) and defense of the laboratory task. For the defense the student had to prepare a report. During the defense the teacher could ask the student 2–3 questions about the program.

The evaluation part of the course includes two tests – “Types of data. Language operations. Control structures” and “Arrays and methods”, as well as mid-term control of student knowledge organized in distance regime with limitations imposed on the time of submission and number of available attempts [5].

Thus, for studying the object-oriented Java language the content and method of the study course “Algorithmization and Programming of Solutions” have been changed.

It is too early to speak of the results, but this method has been successfully applied for more than 10 years at the course “Computer learning” providing the basics of algorithmization and programming for chemistry and electricity program students [6].

4 Topics of Java teaching method integration into related subject

Using student laboratory tasks submissions in both subjects, covered topics were arranged according to difficulty level. Processing results was proved that development environment influences the level of assimilation of topics. The same topics in both development environments were marked with different difficulty level (Table 2).

Table 2. Topics complexity (1 very difficult – 5 very easy)

Topic	Eclipse environment	VBA in Excel environment
Input-output operations.	5	4
Announcement of variables.	5	2
Cycle operators.	4	3
Use of mathematical functions.	4	4
Arrays	2	5
Multidimensional arrays	2	5
Nested loops.	4	2

For example, topics related to the array use are easily understandable within Excel VBA environment, but issues related to the cyclical structures in Eclipse environment.

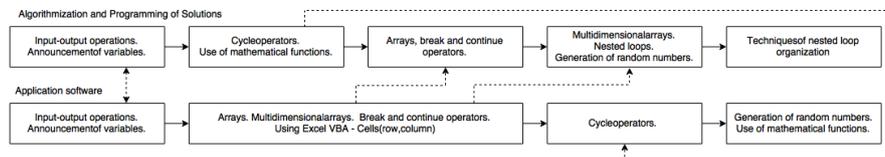


Fig. 3. Java teaching method topics integration in to related subject.

For successful implementation of in this paper described Java teaching method, changes in topic order and explanation are done for related subject “Application software” for the first year students. (Fig.3). This approach allows to improve the quality of mastering topics in “Algorithmization and Programming of Solutions” subject, looking harder thread using other in educational process linked development environments and tools.

5 Conclusions

The choice of initial programming language is very important. The methods and forms of education for future IT-specialists must be chosen correctly. In this paper, the structure and new content of the basic course ”Algorithmization and Programming of Solutions” are presented, as well as the methodological approaches to teaching the Java programming language. This course on Java is taught to the first year students of the Faculty of Computer Science and Information Technology of Riga Technical University. The realization of the course according to the structure expounded in the paper is meant to:

- Promote the professional orientation of the students in mastering object-oriented programming which is currently the leading tool of professional programming;
- Promote the interest of today’s students, yesterday’s schoolchildren in Java and in gaining solid knowledge in object-oriented programming.
- In future, the authors plan to conduct and experimental work in training the first years students in object-oriented programming.

References

1. Prokofjeva N., Uhanova M., Zavjalova O., Katalņņikova S. Structuration of Courses at Studying Disciplines of Programming. No: Proceedings of the 10th International Scientific and Practical Conference "Environment. Technology. Resources", Latvija, Rēzekne, pp.159.-163.lpp. ISBN 978-9984-44-173-3. ISSN 1691-5402. 2015. Rēzekne: Rēzeknes Augstskola, (2015)
2. Actual problems of Education , Inter-Higher School Scientific and Educational Conference, <http://www.tsi.lv/en/content/mip-2015>, MIP (2015)
3. Environment. Technology. Resources. Proceedings of the International Scientific and Practical Conference, <http://journals.ru.lv/index.php/ETR>
4. Edsger W. Dijkstra. A Discipline of Programming. Prentice Hall, Inc., 217 p. (1976)

5. Prokofjeva N., Anohina-Naumecca A., Lebedeva O. Administration of Knowledge Assessment at Riga Technical University. Proceedings of the 8th International Multi-Conference on Computing in the Global Information Technology (ICCGI 2013), IARIA, pp.34-39, ISBN: 978-1-61208-283-7. (indexed in ThinkMind Digital Library), Nice, France (2013)
6. Uhanova M.A. “Study course in basics of programming for chemists-technologists”, Papers of international scientific methodological conference Informatization of engineering education – INFORINO2012, 10–11 April 2012, Moscow, Russia, pp. 331–332. [In Russian] <http://inforino2012.mpei.ru/doc/proc.pdf>