

Models of data and their processing for introductory courses of computer science

Mikola Kryshchuk¹, Juris Lavendels², Vjaceslavs Sitikovs²

¹*Ukrainian National Technical University,*

²*Riga Technical University*

Abstract. Evolution of secondary school course on Informatics during 1990s is discussed. This evolution gave for secondary school graduates rather high level knowledge of applications usage. On the other hand Informatics became purely pragmatic matter within generally academic nature of secondary school curricula. As the result graduates of secondary schools are not prepared enough for mastering of university's course in Computer Science. Authors are not aware of serious studies and methods of prevention of such negative habits caused by evolution of Informatics course. In this article one method applied in university's introduction course of Computer Science is considered. The method provides use of algorithmic system closed to the human mind, and to some extent compensates topics removed from the Informatics course in secondary school.

Keywords: Algorithm analysis and design, informatics, model of data structures.

I INTRODUCTION

Universities, including technical universities, enroll secondary school graduates with good computer end-user skills and with the confidence that their knowledge in the field of ICT is complete and good [1, 2]. School graduates often choose ICT (Computer Science) studio (specialty), although their knowledge regarding specific of the work of Computer Science technician is superficial. Moreover, understanding of specific of such position like programmer is absent at all.

In this situation university's staff involved in Computer Science introductory course teaching is complicated position to work with the very "florid" audience with a distinctly different starting knowledge, interests and motivations. It is necessary to ensure that the course to be interesting and exploitable for whole audiences regardless of prior knowledge of particular student.

II EVOLUTION OF THE INFORMATICS COURSE IN SECONDARY SCHOOL

The course of Informatics in Latvia in the past 50 years experienced a significant evolution [1] and this process is continuing right now. In Latvia, the course of Informatics in secondary schools emerged in 1963/1964 school year in some schools with in-depth learning of mathematics. Since 1985 the course related to Informatics and computational foundations became mandatory in secondary schools. At that time was developed a tendency to teach pupil to think in algorithmically manner, to develop algorithms as well

as using schools' computer equipment to start program development. The course had pronounced academic nature and it was in conformance with the society requirements to the education in the ICT field. During this time the secondary school prepare pupils very well to university's engineering courses, including Computer Science. The first year students well understood the basics of Computer Science. Students understood that the computer just executes algorithm created by human and understood that this algorithm should be presented in the form suitable for computer - in some programming language.

At the beginning of the 1990s both in Latvia and Ukraine appear new trends which give the opportunity to pupil in a great extent to choose training courses by themselves. In practice it leads to reduction of exact sciences hours in pupils' personal portfolios.

In parallel with these trends computers appeared in human communications. Using of personal computers evolved very quickly. It was arose public demand for schools to prepare pupils to use personal computers in everyday life - for working with text documents, using of electronic mail etc. Schools included rapidly computers to their infrastructure and mentioned above public demand was implemented in Informatics course without increasing the number of course's hours. Training of computer usage as end-user extrudes from the courses classical academic topics leaving only one of them - "information and communication technology concepts". On the other hand Informatics course in secondary schools now is considered in terms of ECDL certificate program.

Therefore Informatics course was transformed from academic topic to highly pragmatic course [3, 4, 5], which is basically in contradiction with the mission of the secondary school. This solution ensures public demand, however to the large extent undermine students' preparedness for study in university, especially for engineering studies [3]. Preparation of students for engineering studies partially is resolved by optional courses considering classic Informatics topics, however pupils' knowledge and preparedness for such a course differ essentially.

III SOME ISSUES OF COMPUTER SCIENCE INTRODUCTORY COURSE IN UNIVERSITIES

In previous time university's introductory course for Computer science begins with practical implementation of already acquired algorithmic skills - understanding of data concept and realization of algorithms on flowchart level. This practical realization was based on some programming language that students are study [6]. At that the emphasis was made on the solution in the form of an algorithm, whereas the programming language was considered just as a tool of algorithm formulation for the computer and for checking of achieved result.

At present time the situation has changed significantly – students have solid experience with the computer, they have not difficulties with the file creation, copying, editing, using of different applications and they desire immediately to work on their computers. Initially it seems that mastering of programming basics could not be difficult for nowadays secondary schools graduates. However in practice students cannot develop algorithm for particular task solution and realization of this algorithm on some programming language because students have not knowledge of the exact and detailed formulation of the solution to another executor. In other word they have no knowledge of algorithmization. On the other hand, it is not usually possible to provide in university's program an introductory course of informatics as it will cause the delay of the course of Computer Science and thereby will lead to the change the whole study programme [7, 8].

To prepare students for practical programming in a short time we use the model of program - algorithmic system which provides accurate language for input data description, actions with the data and language for description of program's results. In parallel with the model there are considered elements of programming language as equivalent for algorithm's description in the form of flowchart.

IV DATA MODEL AND ITS APPLICATION FOR DESIGN OF ALGORITHMS

Our approach to mastering of algorithmization and programming basics is rooted on a use of simplified

model of the computerized solution. The model consists of imaginary data storage - data bureau with predefined semantics and of manipulation with data (elements of flowchart with also exactly-defined semantics). The approach provides equivalence of a solution model and elements of programming language. In a such way students can on the very beginning of the course realize elementary algorithms based on graphic image of the algorithm and a on particular operators of programming language that are equivalent to considered flowchart's elements.

Introduced data bureau is useful both the very beginning of the course and considering the placement and relationship, e.g. global and local data in the context of multi-modular program. Data storage is imagining as a bureau with drawers, at that for each type of data is stipulated specific type of drawer. In each drawer is kept just one value (Fig. 1).

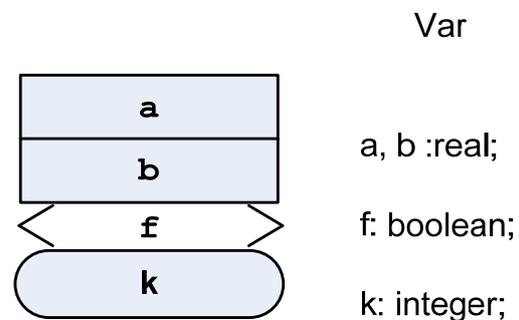


Fig. 1. Data bureau and its equivalence of variables' descriptions in programming language

Students have vague imagination of variables in programming languages [9] because they have no appropriate grounding of computer operations; however they understood immediately semantics of equivalent data bureau:

- Any change of the data value means new data recording to the drawer;
- Inserting data into the drawer the data acquire the type (integer, real, ...) that means conversion to the shape of the drawer or refusal in the case when such conversion is not possible;
- It is possible to take the value from the drawer unlimited number of times, but only if it has previously been recorded there;
- If a new value is inserted to the drawer then previous value is lost and is not renewable.

Depending on the nature of the course semantics of data bureau can be given a different granularity, for instance:

- inserting the value to the drawer (without further specification) or
- previous leaflet with written value is removed from the drawer and is destroyed (if there is leaflet in the drawer), the new stored value is

written on the new leaflet, which is inserted to the drawer.

In the similar manner are explained data structures (arrays, records, collections, etc.). Elements of flowchart and equivalent operators of programming language also are explained in relation to the bureau, see examples in Fig. 2 and Fig. 3.

$$S := p * (p - a) * (p - b) * (p - c) ;$$

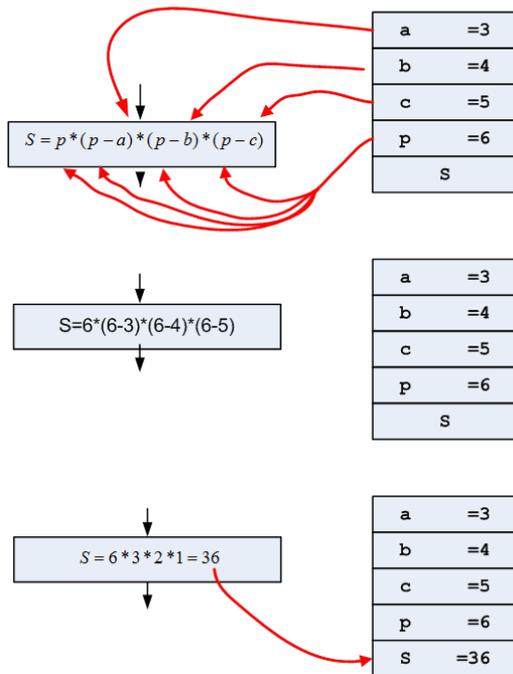


Fig. 2. Semantics of process and assignment

Explaining the teaching material using data bureau, students learn algorithmization in a graphical format proper to human thinking. Semantics of data bureau and elements of flowchart is equivalent to the semantics of programming language operators. This in a great extent facilitates mastering of programming, for instance parameters transfer mechanism and explaining of variables visibility (Fig. 4.).

Described method of algorithmization and methodology of algorithm's realization in program form are practically approved and applied. Semantics of data bureau is practically applied with the different levels of details - at the beginning of the second month we could give up from data bureau on "leaflets" level. During the work data bureau progressively remains unnecessary – it is replaced by data description of a programming language. However considering new programming features, such as internal subprograms and parameters transmission, organization of the external program, visibility o variable etc. it is very useful to introduce new features again on the base of data bureau.

The proposed methodology puts students to work strictly in the algorithmic system with precisely defined semantics. This one hand helps the audience fairly quickly mastering the principles of algorithmization and algorithmic thinking, on the other hand, to some extent compensates the lack of important sections in secondary education informatics course.

This methodology leaves the syntaxes of a programming language in second plan - actually constructions of programming language remain as a solution for presentation of algorithm in the form acceptable for computer. Thus facilities of programming languages are considered just as means of description of a solution in the prescribed form, such as C or PASCAL. This approach provides a simple shift of programming language that is important because the industry is booming. Another advantage of the approach is its direct applicability in the teaching of programming languages built in application software, such as VBA.

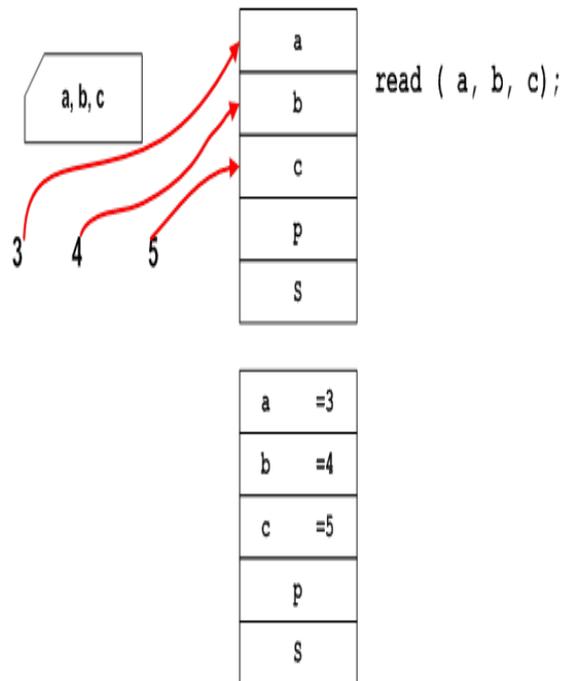


Fig. 3. Semantics of data input.

V CONCLUSION

During 1990s have been made significant changes in Informatics course of secondary school that provided public demand – secondary school graduates are skilled computer users in application level.

Simultaneously with positive aspects of this evolution also appeared negative trends - school graduates are poorly prepared for the university's course of Computer Science. These negative trends are little studied, and we are not aware of the universal method for correcting of these negative characteristics.

In this article is described one method (fig.1, 4) that could be applied in introductory course of Computer Science, which provides applying to teaching

algorithmic system model close to the human mind, and to some extent compensates informatics course sections removed from secondary school curricula.

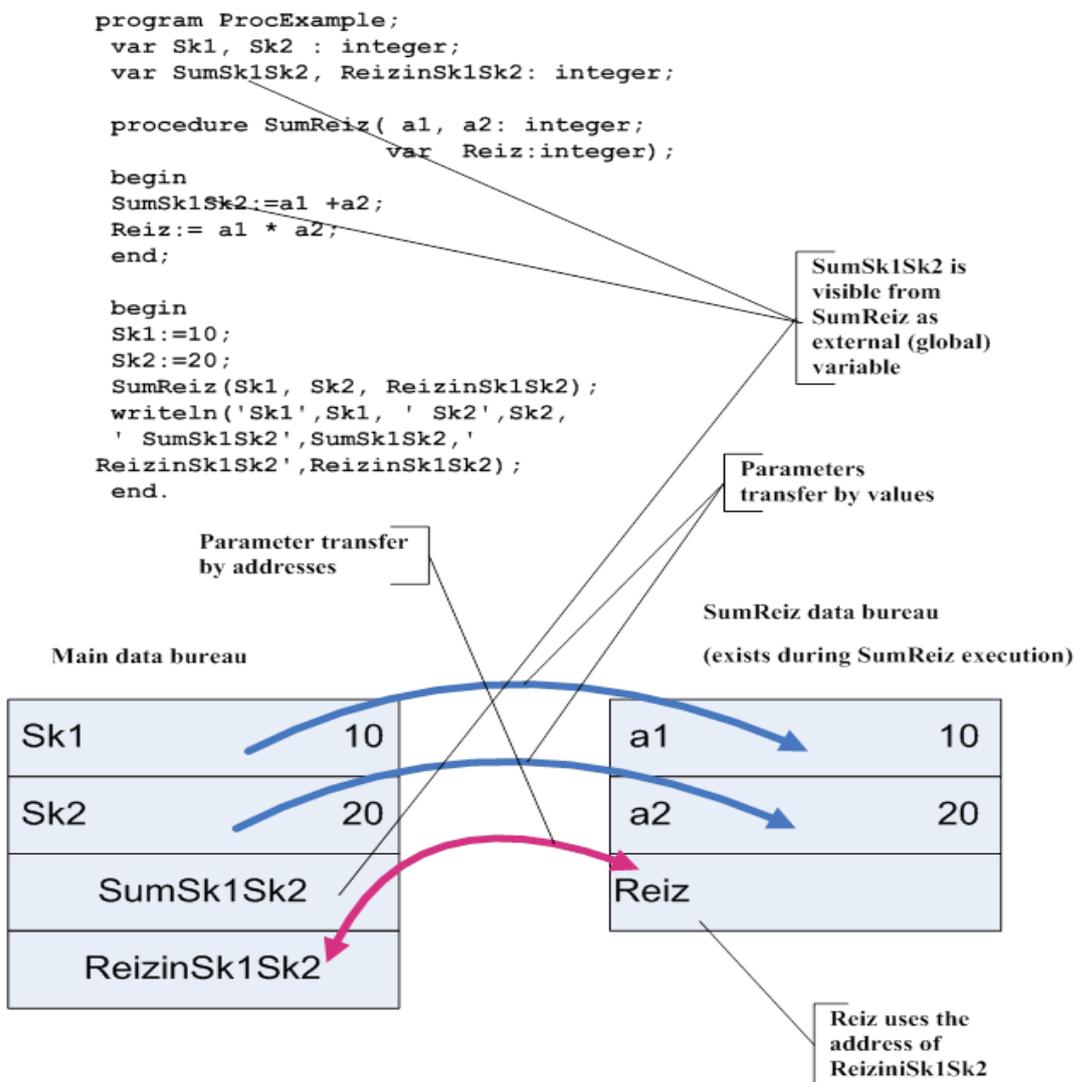


Fig. 4. Visibility of variables and mechanisms of parameter transfer to sub-programs.

VI REFERENCE

- [1] V. Vezis, "Informatics at the school," Dr. sc. comp work summary, University of Latvia, Riga, Latvia, 2004., pp. 36.
- [2] V. Vezis, "Contents of School Informatics," in *Proceedings of The LatSTE'2004*, LU, Riga, Latvia, 2004, pp. 92–99.
- [3] "Latvia general secondary education standard informatics," 2013. [Online]. Available: <http://likumi.lv/doc.php?id=257229>
- [4] "Ukrainen general secondary education standards," 2011. [Online]. Available: <http://zakon4.rada.gov.ua/laws/show/1392-2011-%D0%BF>
- [5] "Ukrainen general secondary education standard informatics," 2014. [Online]. Available: <http://refdb.ru/look/2479127-pall.html>
- [6] P. Zarins, V. Vēzis, "Mathematics and informatics didactic course optimisation problems and their solutions," in *Mathematics teaching: history and perspective (3. international Scientific Conference)*, LPA, Liepaja, Latvia, 2002 pp. 157–161.
- [7] A. Anohina-Naumeča, J. Grundspeņķis, J. Lavendels, V. Šitikovs, V. Vinogradova, "Improvement of the University Programme on the Base of Modularity," in *European Integration Studies: Research and Topicalities*, 2008, pp. 25-30. ISSN 1822-8402.
- [8] A. Anohina-Naumeča, J. Grundspeņķis, J. Lavendels, V. Šitikovs, V. Vinogradova, "Modularity in study programme "Computer systems" development," in *The quality of education: business computer systems (The Riga symposium)*, RTU, Riga, Latvia, 2010 pp 13-22. ISSN 1691-631X.
- [9] I. Rivkind, T. Lisenko, L. Chernikova, V. Chahodko, *Informatics, (in Ukrainian) Geneva, Kiev, Ukraine, 2010*, pp. 302. [Online] . Available: <http://4book.org/uchebniki-ukraina/10-klass/483-informatika-10-klas-rivkind>.