

# Algorithms Constructor

Iryna Zaretska<sup>1</sup>, Grygorii Zholtkevych<sup>1</sup>, Andrii Radchenko<sup>1</sup>, and Alexey Minayev<sup>1</sup>

<sup>1</sup>V.N. Karazin Kharkiv National University, Kharkiv, Ukraine  
zaretskaya@karazin.ua, g.zholykevych@karazin.ua,  
wsooh16@gmail.com, alexlm55555@gmail.com

**Abstract.** The paper presents an environment for creating, editing, tracing and running, saving and loading algorithms in a flowchart form. It can be used in any educational process whenever the basics of algorithms are to be learned. The requirements, design of the system and its implementation are presented. Two modules are developed: for a trainer to fill in the system with the problems on creating algorithms and their right solutions in the form of a flowchart, and for a trainee to solve these problems by creating and tracing algorithms and gaining skills in algorithms construction.

**Keywords.** Software system, object-oriented design, algorithm, flowchart.

## 1 Introduction

In IT education wherever it takes place – at the elementary or the secondary school, university, any other secondary or higher educational institution or even just IT school – it is of utmost importance to gain fundamental knowledge and skills in basics of algorithms, which form the base for further education. To make the process of teaching and learning basics of algorithms more IT-like we developed a software system which in fact is an IDE (Integrated Development Environment) for working with algorithms – creating, saving in files, loading from files, editing, tracing and running. The system has two modules – for trainees and for trainers.

The trainer module allows user to formulate a problem and add it to an existing category or to a new one (all the problems are supposed to be divided into categories). Then a trainee creates a flowchart of an algorithm or a set of flowcharts (if there are several solutions) being the right solution to this problem by using flowchart blocks and filling them with commands. There is an option to add some extra blocks with commands aside from the right flowchart's blocks for a trainee to widen the variety of choices. The problem and its flowchart solutions should be saved.

The trainee module allows user to choose the problem from the list of problems of the certain category and to create the flowchart of an algorithm using blocks with commands created by a trainer. He or she can then check the solution, edit it and check again, run or trace the algorithm under different input data and see control paths and outputs, save it in any folder for further usage.

The proposed system was developed by request of the Ukrainian publishing agency “Ranok” in frame of the international project on joint school education as an electron-

ic support for several textbooks on informatics for schools published by this agency. The first version of the system with Ukrainian interface only was disseminated two years ago via the site of the publishing agency “Ranok” among the Ukrainian schools. The feedback was analyzed and corrections and improvements implemented. The paper presents the new version of the system.

The system goes with the predefined initial set of about 50 problems and their solutions developed by the authors. All problems are divided into categories (linear algorithms, forks, iterations, and combined algorithms). New categories and problems can be easily added using trainer’s module as well as any of the problems of the initial set can be deleted or edited by a trainer.

The system can be used for individual training as well as in the class work. In second case a teacher can assign different problems to different students and check their solutions saved in agreed network folder later, which considerably decreases teachers’ loads.

## **2 Related Work**

In the last decade of previous century the theory of R-charts was proposed by Russian professor I.V. Velbitskiy. He developed a new technology of visual programming by R-charts which allowed drawing programs instead of writing them. He implemented the universal graphical environment to create programs in any existing programming language [1-4]. His research was continued and enhanced with UML notation [5]. The R-charts were used mostly on industrial level for teaching the software technology in the process of software systems development. The theory and technology are quite complicated to be used at the introductory stage of learning algorithms.

Some research in the area were conducted in Kherson state university by professors A. Spivakovskii , M. Lvov and their followers [6-9]. They created a special instrumental language that allowed interpreting an algorithm. It was created especially for teaching students basics of algorithms but it had no graphical tool for constructing flowcharts.

Nowadays there exist a lot of off-line and on-line flowcharts construction tools but they are mostly used for modelling business processes, work and data flows, activities, etc. Usually they have quite complicated interfaces for the beginners, do not allow user to check or to trace or to run an algorithm, explore different control paths under different input data, examine output data and make conclusions, which the process of learning requires. There exist several similar software systems like Algorithm Flowchart Editor (AFCE) (<https://github.com/viktor-zin/afce/tree/gh-pages/download>) or Flowgorithm (<http://www.flowgorithm.org/>) which can be used in educational process, but one cannot trace or run an algorithm to explore it and no set of problems and no interface to create them are proposed.

### 3 Requirements for Algorithms Constructor

About 200 school teachers participated in the process of requirements elicitation during the teachers' conferences, multiple webinars and university post-diploma education. Surveys, questionnaires, and prototyping were used in the process, which led to the following set of requirements. The system has two types of users with roles "trainee" and "trainer". Their main functions are shown in Fig. 1 and Fig. 2. The system supports three interface languages – English, Ukrainian and Russian. The prototype of the user interface design looks like shown in Fig. 3. The system is a desktop application. The deliverable is an executable file (\*.exe). The presentation and business logic levels are implemented with Python. The data storage (problems, flowcharts, solutions) is implemented as json-format files.

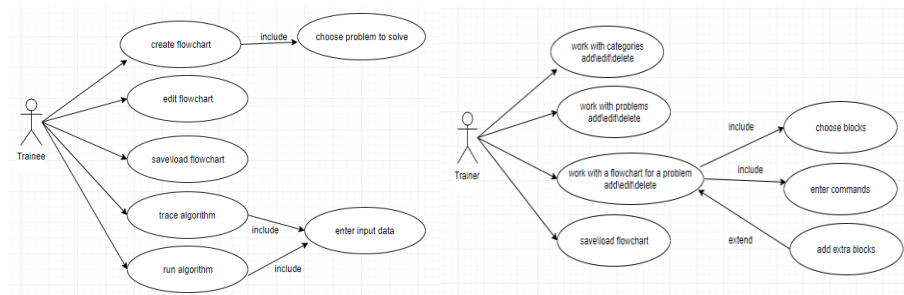


Fig. 1. Use case diagram for a trainee.

Fig. 2. Use case diagram for a trainer.

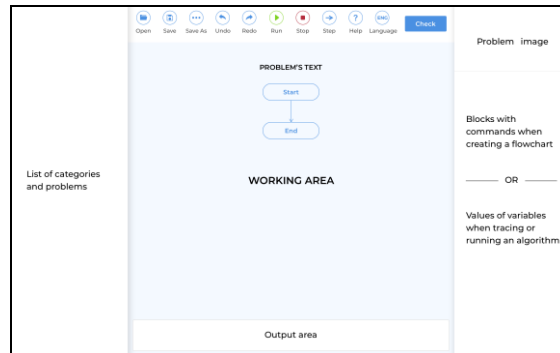


Fig. 3. User interface prototype.

### 4 Design and Implementation of Algorithms Constructor

We used an object-oriented approach and UML diagrams to represent the static structure of the system and its behavior. The main goal of the design was to make it reusa-

ble for both trainee and trainer modules and to follow SOLID principles [10, 11]. The whole number of classes is about 100 including GUI structure and data access classes. Here we present only the general structure of the whole system and its main concepts which are shown in Fig. 4, Fig. 5 and Fig. 6.

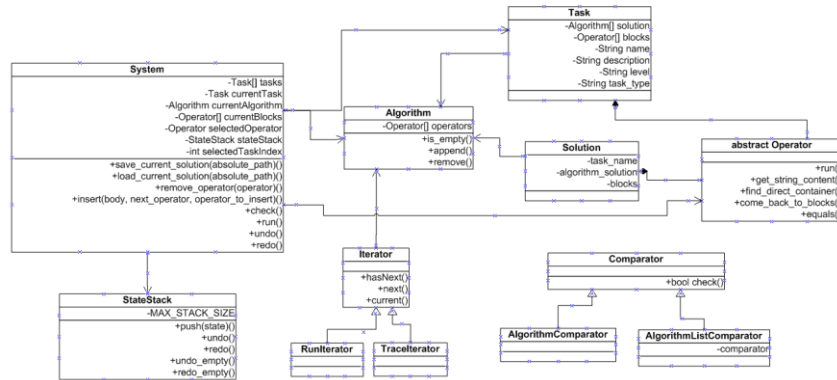


Fig. 4. Class diagram for the whole system structure.

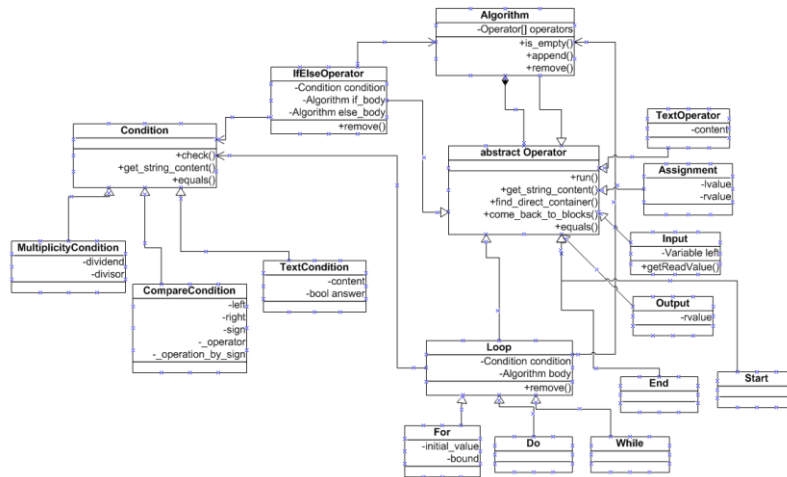


Fig. 5. Class diagram for Algorithm structure.

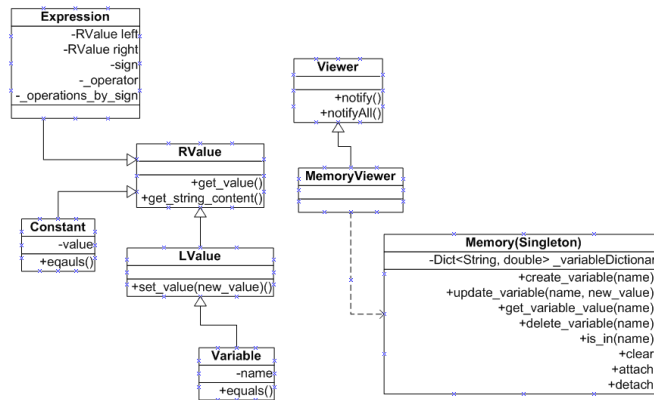


Fig. 6. Class diagram for tracing the values of variables.

The following two snapshots of the trainee module show the operation of the system.

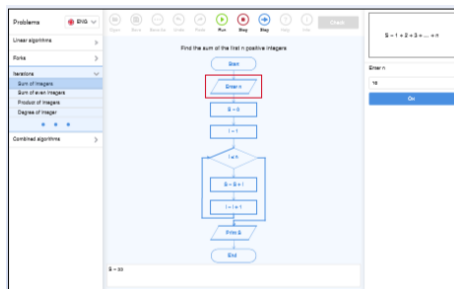


Fig. 7. Running an algorithm.

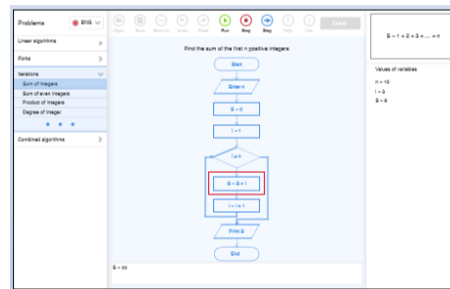


Fig. 8. Tracing an algorithm.

## 5 Evaluation Results

The first version of the system has been in use in Ukrainian schools for about two years. About 100 teachers participated in its testing. The feedback was positive but 50% of teachers found the interface too heavy and gloomy for children. So the interface was changed and the trainee's module was added. The new version was spread out via the site of the publishing office "Ranok". The same 100 teachers gave 100% positive response. They all confirm that it helps to motivate their students and greatly lessens the teachers' load.

## 6 Conclusions

The paper presents a software system for learning and teaching basics of algorithms through flowcharts construction. It can be used by trainees to learn how to create algorithms of different complexity levels using only basic algorithm's structures, "run" the constructed algorithms with different input data and explore all control paths. It also can be used by trainers to fill in the system with problems and their right solutions in the form of flowcharts. The system was developed exceptionally for educational purposes. It can be used in any educational institution from elementary school to university to learn and teach basics of algorithms and to understand basic algorithm structures.

## References

1. McHenry William K. R-Technology: A Soviet Visual Programming. In: Journal of Visual Languages and Computing, vol.1,#2, pp.199-212 (1990)
2. I.V. Velbitskiy: Next generation visual programming technology with R-charts. In: Plenary report, MEDIAS-2012. Dedicated to 100anniversary of Alan Turing (IEEE), Cyprus, pp. xiv-xxxiv (2012)
3. I.V. Velbitskiy. Next generation visual programming technology. In: 11-th IEEE EAST-WEST DESIGN & TEST SYMPOSIUM (IEEE EWDTs), Russia, Rostov on Don, Plenary report, Sept.27-30, pp. 404-410 (2013)
4. I.V. Velbitskiy. Graphical programming and programs verification. In: 9-th IEEE COMPUTER SCIENCE & INFORMATION TECHNOLOGIES CONFERENCE, ARMENIA, YEREVAN (2013)
5. L.F. Drobushевич. Common use of UML and R-chart notations in the training process for software system development methods. In: MEDIAS-2010, Cyprus, pp.73-77 (2010)
6. M. Lvov, A. Spivakovskii. Video-interpreter of search and sorting algorithms. In: Informatizatsiya osvity v Ukrayini: stan, problem, perspektivy, Kherson, pp. 100-102 (2003) (in Ukrainian)
7. A. Spivakovskii, N. Kolesnikova, N. Tkachuk, I. Tkachuk. Web-environment for learning basics of algorithms and programming. In: Upravlyayushye sistemy i mashiny, Kiev, pp. 70-75 (2008) (In Russian)
8. A. Spivakovskii, N. Kolesnikova. Video-interpreter of algorithms of integrated environment for learning basics of algorithms and programming. In: Third International conference "New information technologies in education for everybody", Kiev, pp. 399-404 (2008) (in Ukrainian).
9. Spivakovsky A., Kolesnikova N., Tkachuk N., Tkachuk I. An integrated training environment for the university course "Basics of algorithmization and programming". In: Information Technologies in education for all, Kiev, pp. 240-248 (2007)
10. Robert C. Martin. Design Principles and Design Patterns. In: <http://www.objectmentor.com>
11. Robert C. Martin. Designing Object Oriented Applications using UML, 2d. ed. In: Prentice Hall (1999)