Training in Programming using Innovative Means

G I Shivacheva, N R Ruseva
Trakia University of Stara Zagora, Faculty of Technics and Technologies, Departament of Electrical Engineering, Electronics and Automatics
38 Graf Ignatiev str., 8600 Yambol, Bulgaria

e-mail: galya.shivacheva@trakia-uni.bg

Abstract. Training in programming, related to studying algorithms, is challenging for the lecturers. The aim of this paper is to offer options about how to use innovative means in programming by presenting their specifics and to suggest how to limit some of the disadvantages. During synchronous learning, whether conducted in a computer room or online in a virtual classroom, the three standard ways of presenting algorithms can be applied as creating a flowchart for a specific algorithm with Flowgorithm is presented dynamically rather than as a static image, as on the site. With the introduction of Accumulative frame models in programming training, it can be described verbally through a list of invariant frames, which is recommended to be used in the study of algorithms that consist of more than one basic algorithms. This is a fourth way of presenting algorithms, which can be in the form of a table.

1. Introduction

Training in programming which relates to studying algorithms is challenging for the lecturers. Particularly relevant and important now is the use of innovative tools, as a response to the extraordinary emergent need for distance learning because of the global pandemic. And it is known that the more senses involved in the collection of information, the more of it is perceived. That is why it is appropriate to get involved in programming training means to present the information and in particular the algorithms in different ways in order to be perceived by different senses. Graphical and tabular presentation information is considered easier than text.

In computer science, three main ways are used to describe algorithms:

- Verbal, using natural language;
- Graphic, using block-schematic language;
- Programming language, which allows for the execution of the algorithm by a computer.

Modern programming languages are numerous and differ in syntax. Students or learners in general may need to learn and use more than one language. Therefore, it is better not to focus on the language, but on the basic algorithms, which are the same, regardless of which of the languages are implemented.

There are many means by which it is easier and clearer to present graphically through a flowchart and various means of visualizing the implementation of an algorithm, which also favors its faster understanding and perception.

Verbal description can also be used to represent algorithms through invariant frames, which can also be presented in tables [1].

The aim of this paper is to propose options for the use of innovative tools in programming training, reflecting their specifics and suggesting how to reduce some of their shortcomings.
2. Innovative tools suitable for programming training

Programming training is mainly related to the study of data structures and algorithms that are suitable for visual presentation.

To reduce the problems related to the syntax of programming languages and to facilitate the perception of algorithms in their study, visualization tools are used, which illustrate and reduce the level of abstraction.

There are visualization tools that perform one or more of the following functions:

- Description of an algorithm using a block diagram language;
- Description of the algorithm by programming language, the elements of which are represented by separate blocks and the creation of the program, resembling the arrangement of jigsaw puzzle;
- Testing of an algorithm represented by a block diagram or by a programming language;
- Step-by-step execution of program code and interactive presentation in tabular form of the names of the variables and their current values during execution. They are considered and analyzed in detail in [2].

Flowgorithm is convenient and easy to use when presenting algorithms with simple and indexed variables that are elements of a one-dimensional array.

Python Tutor can also be used for two-dimensional arrays and dynamic information structures. As the complexity of the presented structures increases, their perception becomes more difficult, because the visualization also becomes more complicated. In this paper we will present two additional tools: Lucidchart and Bandicam.

Lucidchart is a cloud solution that can create flowcharts by dragging and dropping the selected block to the appropriate location. There are options for changing the color, font and saving the created scheme in a different file format. Allows integration with Google documents and compatible with Microsoft Visio (import and export documents). The resulting block diagrams can be saved in file format and pdf, png and jpg [3].

Bandicam is a video recording program that must be installed on a computer in order to be used. It can be used to record video from a webcam or to record on the screen. Video capture is done with a button that is convenient for the user and is set in the settings. One click of the button starts and stops recording. The created video can be saved in AVI or MP4 file formats [4].

3. Accumulative frame models

Accumulative frame models (AFMs) are relatively self-contained, logically distinct units that are reusable in a variety of situations. A named frame structure consisting of slots is used for their description.

In the case of training for acquiring programming skills and style, frame models are proposed under the name “invariants” (passages of program code, which with small non-structural changes are found in the solutions of problems for programming) in [5].

Each frame-invariant in the programming training includes slots for type, name and description, related to the variables that are used in the program code for its implementation.

A verbal description using invariant frames can also be used to represent algorithms, which can also be formed as a table (Table 1). This is also one of the tasks related to the use of AFM in programming training: designing solutions to programming tasks in the form of verbal descriptions (composed of invariant frames with specific slot content).
Table 1. Instance frames to describe the exchange algorithm.

| №  | Frame name                  | Variable type | Variable name | Description of the variable | Program code |
|----|-----------------------------|---------------|---------------|-----------------------------|--------------|
| 1  | Declaring a variable        | int           | a             | Exchange variable           | int a;       |
| 2  | Declaring a variable        | int           | b             | Exchange variable           | int b;       |
| 3  | Declaring a variable        | int           | c             | Auxiliary variable for exchange | int c;       |
| 4  | Enter the value of a variable |              |               |                              | cin>>a;       |
| 5  | Enter the value of a variable |              |               |                              | cin>>b;       |
| 6  | Exchange of two variables   |              |               |                              | c=a; a=b; b=c; |
| 7  | Display the value of a variable |          |               |                              | cout<<a<<endl; |
| 8  | Display the value of a variable |          |               |                              | cout<<b<<endl; |

To successfully implement this new way of presenting algorithms, pre-training programming must include the following tasks:

- Acquaintance with frequently used invariants for programming (special with the content of the slot "Program code");
- Analysis of unknown program code, detection of invariants (including new ones), as well as the values of their parameters (content of the respective slots) [6];
- Modification of an unknown program code (by changing the content of the slots of invariant frames and entering new invariants).

Software tools for the use of accumulative frame models in programming are presented in [7].

A site (http://vltp.atwebpages.com/) has been created for the implementation of AFM-based programming training with functions for: acquaintance with invariants; reading with understanding of program code (detection of invariants), modification of program code; verbal description through invariants and design of input and output data [8].

When using the function for verbal descriptions through invariants, an invariant is collected from a predefined list and the values of the slots are filled in: variable name, variable type, description and program code for a task with a given condition. Based on the selected frame and the completed fields two text files are generated - the first is the code of the frame and the values of its slots and the second is with a verbal description, including instead of numeric codes names of the invariants. For more than one variable in one invariant - the names of the variables are set separated by commas, and in the field type of variable for each separately is set type - as in the list of parameters of a function - e.g. double a, double b.

The name of the invariant is selected from the following list:

- Declaring a variable
- Declaring an array
- Enter the value of a variable
- Enter the values of the elements of an array
- Search for a minimum item
- Search for the maximum item
- Exchange of two variables
- Array sorting
- Display the value of a variable
- Display the values of the elements of an array

This presentation of the algorithms can be used both during lectures in describing a new algorithm that incorporates some of the already studied comma-invariants and during exercise. When writing program code that implements an algorithm through a list of invariants, some of them can be described as subroutines, e.g. swapping the values of two variables.
4. Discussion

Based on the review of existing innovative tools related to programming training and in particular with the description of algorithms was selected for graphical description through a block diagram (block diagram language) when creating a site with basic algorithms, the Flowgorithm application, because it can also automatically generate program code (description using a programming language) and be executed with specific data that are entered from the keyboard.

Lucidchart is suitable for use for a block diagram in general without a specific condition for verification, because on its basis there is no way to generate program code or execute it through Flowgorithm.

Python Tutor was chosen to improve programming training by visualizing the implementation of algorithms.

Created site that includes for each of the algorithms they verbal description, graphical representation by flowchart created with an innovative tool and code realization and visualization.

Discussed are seven basic algorithms that are associated with early education programming and are implemented and only with simple variables. One of them is described with two different implementations. Others are considered in several variants related to their varieties. The following algorithms are presented in detail:

- Exchange the values of two variables
  - By auxiliary variable
  - By arithmetic operations
- Finding the smallest number
- Finding the largest number
- Sum of a number of numbers
- The product of a number of numbers
- Number of a number of numbers that meet a given condition
  - In general (without a specific condition)
  - Provided multiplicity of a given number
- Separate digits by number
  - Display the digits of a number in reverse order
  - Sum of digits of a number
  - Number of digits of a number
  - Each of them is described in three ways:
    - Verbal, using natural language;
    - Graphic, using block-schematic language
    - Programming language.

Figure 1 shows with Flowgorithm a representation of an algorithm for exchanging the values of two variables with a third auxiliary variable through a block diagram and a C++ programming language and its execution with user-entered input data.
Visualization of the algorithm has also been added in two ways: through Flowgorithm and Python Tutor. The same type of presentation allows the construction of a system that can be applied to all other algorithms in the learning process. The graphical description through a block diagram, the visual presentation step by step of the algorithm and the different colors used according to the type of data, as well as the display of the current values and the automatic generation of program code based on a block diagram contribute to easier perception and assimilation of the algorithms in question.

To generate the block diagrams for a description of the algorithms a Flowgorithm or Lucidchart (used for representation of the algorithm in a general form) are used. Python Tutor is used for visualization of the algorithms, and hyperlinks to the respective visualization of program code have been added on the site.

With the help of Bandicam, videos were created that show how to create a flowchart with Flowgorithm and how to generate a preview URL to share the current execution point of the algorithms with Python Tutor. They have been added to the site as hyperlinks due to the limitation of the online version of WordPress.

For greater clarity and completeness, the automatically generated on the basis of the block diagram program code of C++ and its execution for input data set by us using Flowgorithm have been added, as well as visualization of the algorithm step by step tracking the change of the input data through intermediate results until the output data is obtained.

The verbal description includes a description of the algorithm in natural language. The block diagram, presentation and execution of the C++ programming language code are created by Flowgorithm. The program code is its description through C++ programming language, studied at the Faculty of Engineering and Technology in the disciplines: "Programming and use of computers - part two" and "Algorithms and data structures". The step-by-step visualization of the implementation is generated using Python Tutor and is added to the site as a hyperlink.

Python Tutor allows, after the code in your web browser is typed or copied from a development environment, to monitor its implementation step by step. The application allows you to save a link to the visualization of the specific code. The disadvantage of Python Tutor is the inability to enter data from the keyboard. The numerical data for the algorithms considered by us can be set by assigning specific values to the respective variables or by generating through the random number generator. The created site https://algorithmsbasic.wordpress.com/ shows both options. When setting of the input program for a range of numbers the second one is preferred, in order to get the opportunity for random selection and a wide variety of values. An advantage of the generation is that the interval from which the numerical data is selected can also be changed.
5. Conclusion
The clean of the created site https://algorithmsbasic.wordpress.com/, the uniform presentation of all considered algorithms, the used innovative means for creation of block diagrams, visualization of the implementation of the algorithms step by step, can contribute to the improvement of the process of programming training.

The recorded videos to illustrate the process of working with Flowgorithm and Python Tutor allow students to learn to use both applications independently.

The created site can be used as additional information in programming training, which students can use independently. including independently by students

During synchronous learning, whether conducted in a computer room or online in a virtual classroom, the same ways of presenting algorithms can be used as creating a flowchart for a specific algorithm with Flowgorithm is presented dynamically rather than as a static image, as on the site. In its construction it is good to actively involve students by trying to implement the verbally described algorithm or to make a graphical representation based on a given condition of the programming task.

In programming training, problems of the following type are usually solved: a condition of a task is given to describe an algorithm in one of three ways: verbally, through block diagrams or through a programming language. With the introduction of AFM in programming training, it can be described verbally and through a list of invariant frames, which is recommended to be used in the study of algorithms that consist of more than one basic algorithm. With the introduction of AFM in programming training can be described verbally and through a list of invariant frames, which is recommended to be applied in the study of algorithms that consist of more than one basic algorithm. For this type of description can use the relevant function of the site http://vltp.atwebpages.com/.

Acknowledgments
This study was supported by Grant 1.FTT/ 2020, “International Conference on Technics, Technologies and Education ICTTE 2020” from the Faculty of Technics and Technologies of Yambol, Trakia University of Stara Zagora, Bulgaria.

6. References
[1] Shivacheva G and Totkov G 2017 Accumulative frame models in programming training Proc. Int. Conf. on Technics, Technologies and Education ICTTE 2017 October 19-20 2017 (Yambol: Bulgaria) pp 277 - 293 (in Bulgarian)
[2] Shivacheva G 2018 Study of branched algorithms by means of visualization Proc. of the XXVII International Scientific Conference “Management and Quality” for young scientists (Yambol: Bulgaria) pp 71- 78 (in Bulgarian)
[3] https://www.lucidchart.com/
[4] https://www.bandicam.com/
[5] Somova E, Enev J and Totkov G 2014 Invariants in programming training International scientific on-line journal Natural & Mathematical science vol 4 no 3
[6] Shivacheva G, Totkov G and Doneva R 2017 Reading comprehension of program code Scientific works of the Union of Scientists in Bulgaria - Ploddiv, Series B. Technique and technologies vol XV pp 58-62 (in Bulgarian)
[7] Doneva R, Gaftandzhieva S, Pashev G and Totkov G 2020 A Software Tool For Programming Training Trough Accumulative Frame System International Journal of Scientific & Technology Research vol 9 pp 1389-1393
[8] Shivacheva G 2019 Structure Design of the Control Test Examples in Solving Programming Tasks Proceedings of the 14th International Conference on Virtual Learning ICVL pp 432-437