Methodological Aspects of the Initial Training of Students for Participation in Programming Contests

Galina E. Atanasova
Dept. of Natural Sciences and Education
Angel Kanchev University of Ruse
gea@ami.uni-ruse.bg

Plamenka T. Hristova
Dept. of Natural Sciences and Education
Angel Kanchev University of Ruse
ptx@ami.uni-ruse.bg

Abstract

Bulgaria is one of the few countries in the world where extracurricular activities and competitions in Informatics start from early school age (prior to fifth grade – 11 years). The implementation of this activity is associated with many challenges of organizational and methodological nature. Development of new approaches and their approbation in real conditions contribute to increasing the effectiveness and quality of this kind of training. Sharing the collected experience and methodological materials will facilitate teachers in the development of extracurricular activities in informatics. The present paper presents the authors' experience in extracurricular activities for young competitors training in informatics.

1. Introduction

The relevance and the importance of the problem for training and development of gifted children derives from the need of the modern globalized society of creative thinking individuals with a comprehensive and systematic understanding of the nature and relationship between objects, events and people. The contemporary world with its rapid growth of information, constantly changing technology and lifestyles requires the formation of individuals who can promptly and flexibly respond by changing their thinking, habits and behavior.

Informatics attracts students with its topicality and future professional orientation prospects. The training and participation in competitions are one way to meet the interest in it.

In recent years, there is a tendency to lower the age of the competitors in Informatics. Early training in competitive programming provides opportunities and time for participants to develop and achieve better results in the higher age groups.

Unlike the teaching of mathematics, subjects that train students in programming are not included in basic training courses in Bulgaria. This requires the training to be delivered entirely in extracurricular forms. There is no state-imposed curriculum, methodological materials and textbooks.

Largely, the ideas that underlie the training of students with mathematical skills, are also applicable to the students that actively train for participation in competitions in informatics. Similarly, in the training in Informatics the idea of a pyramid organization of work has forced its way through. This includes selection and subsequent work with students of different levels and with varying intensity, consistent with their individual circumstances and wishes. But at this moment, the pedagogical aspects of discovering, training and developing of young talents in the field of informatics are still quite far. These aspects are in the initial stage of development and they need widening and deepening, taking into account the specifics of the field.

The process on finding, training and developing young talents in the field of informatics has its own specifics that must be considered. Lately, more and more researchers have been focusing on ways to prepare talented students in schools in Informatics. In [Gar08] K. Garov reviews a study of informatics and IT in Bulgarian school. He discusses the history of competitions and Olympiads in Informatics and he offers a model for management of training of the participants in Olympiads in Informatics and IT and system support tasks for the training of outstanding students to participate in Olympiads in Informatics. In [Yov09] B. Yovcheva offers a spiral approach in which to perform the training of 10 and 11 years old pupils in Informatics.

According to Nicklaus Wirth "the programming is constructive art" [Wir75]. This assertion raises the question: What methods and approaches should be used to prepare the children to explore this constructive creative activity? One possible approach is via exploring specific examples to summarize the basic

In: A. Bădică, M. Colhon (eds.): Proceedings of the 2015 Balkan Conference on Informatics: Advances in ICT

elements for programs construction and then put them in a systematic way. This consequently allows students to apply the principle of reuse. Actual programming should cover systems of complex requirements, which conclude a high level of intellectual activity. For this reason, the idea of "pure training recipe" is probably wrong. The choice of teaching methods must be supported by careful selection and presentation of the key examples. Of course, we should not think that anyone can derive equal benefit from the study of these examples. Typical of this approach is that many rely on the learner, his diligence and intuition. This is especially true for relatively more complex and longer sample programs. They are "normal" case in practice and they are much more suitable for demonstrating subtle but important elements called organization and style of programming.

2. Extracurricular activities in informatics

Extracurricular activities in informatics have the following main objectives [3]:

- Preparation of elite competitors in Informatics;
- Anticipatory learning and programming skills for students in primary grades;
- Development of the intellectual potential of the nation by creating conditions for the development of gifted children from the earliest possible age through sets of computer science (intellectual skills such as analysis, synthesis, induction, deduction, generalization, critical thinking, discipline of thought and others);
- Extra training in programming of a larger number of students;
- Personal qualities development (discipline, orderliness, consistency, teamwork, etc.);
- Constructive thinking adoption;
- Basic school course knowledge of informatics expansion;
- Students' additional computer literacy.

The following general characteristics of intellectual brilliance and individual differences between the children must also be taken into account in teaching Informatics:

- Increased cognitive demand that occurs in a wide curiosity and exploratory behavior;
- Increased demand for mental workload;
- Perseverance to achieve self-set goal;
- High level of generalization and abstract reasoning;
- Ability to formulate unusual ideas and new ways of solution.

These individual characteristics influence the pace of preparation of young children. There are children who need more time to feel confident in themselves and who overcome slowly the psychological barriers created by participation in competitions. Students who are not prepared well enough (theoretically, practically or psychologically) should not participate in the competitions. This can lead to refusal of engagement with informatics, and even to a mental breakdown. However, we should not go to the other extreme and believe that the child is not prepared well enough for a long period of time and not allow him/her to participate in competitions. The effect can be the same.

So the question arises: "When is a child ready to take part in his/her first competition in informatics?" The answer to this question should be given by the head of the group in which he/she is prepared, taking into account the following factors:

- Theoretical training;
- Practical training;
- Desire for expression;
- Emotional stability.

3. Principles for a methodology choice

The chosen methodology defines the role of training activities and both sides in the training process interactions. It determines the tasks and activities sequence. The problem with this choice is extremely complex because of the need for teaching-learning unity reflection. To enable the chosen methodology to lead to effective learning process, the following steps are recommended:

• The specific tasks selection to be consistent with the adequate level of hardships;

• To identify dependent and independent sequence of activities that lead to achieving the objectives. The dependent one stems from the logic of educational content and the independent does not affect the achievement of the objectives (when the time of the contest is irrelevant, since the student is prepared to take part in it);

• The successful construction of the relationship between dependent and independent consecutive activities leads to systematization and simplifies the achievement of training objectives.

The foundation of a successful methodology for training in programming contests stands on a detailed analysis of tasks and their types, targets for a given competitor, his cognitive level and pace of development.

Analysis of the above issues in the ongoing training of young competitors in programming contests gives grounds to look for approaches and methodologies of training, fully tailored to the specific needs of such kind of competitors. The age of the participants in these contests and modern competition reality justify the usage of an active approach to the selection of a methodology for training in programming contests. Underlying this approach is the idea that human activity is formed under the influence of its needs, the benefits of which are reflected in the objectives of the actions. Particular attention is paid to the grounds as an important structural component of the cognitive activity of the student. The motivation for actions proposes and forms a major instigator for its implementation. In competitors' training it can be argued that the conditions for creation of a clear motive for learning will be reflected in precise objectives and will provide better results and consequently quality of training.

The advantages of this active approach to the choice of methodology for the competitors training for programming contests is reflected in the combination of the characteristics of object-studied area and age of the children. Clarifying the need for training is an important prerequisite for successful performance and future good ranking. It serves as an inducement for the activity of learning in the process of training.

The implementation of such an active approach has led to the following methodology for the young competitors' training for participation in the programming contests:

- Cognitive tasks consideration as a separate unit, regardless of syntactic specifics of a programming language;
- Using modern approaches and techniques for awareness, understanding and learning of the basic tasks set;
- Consolidating the knowledge capable of analyzing individual gaps of each competitor;
- The knowledge and skills application by solving other similar tasks;
- Non-standard problems solving which consist an element of creativity;
- Analysis and assessment of the acquired skills to solve contest tasks.

The used language for the practical training is C++. Its usage is required by the regulations of the International Olympiad in Informatics respectively of the National Olympiad in Informatics.

4. Proposed methodology for programming contests training

As already was highlighted, and according to one of the founders of modern scientific research Donald Knuth [Knu73] programming is an art, so the pure stereotype of training will develop giftedness of students. Classes should be directed to the development of giftedness in children. Specific logical thinking and unconventional solutions should be promoted and children, who display them, should be encouraged to develop them.

In the process of development of extracurricular activities in informatics in Bulgaria there were formed five age groups: 4-5 grades, 6-7 grades, 8th grade, grades 9-10 and grades 11-12 (hereinafter, respectively group E, D, C, B, A).

According to the adopted structure of curricula, the basics of algorithms are studied in the early age groups (E and D), which include students up to 13 years. For these age groups the school curriculum does not offer utilization of such knowledge. Therefore, small pupils should learn faster knowledge and skills brand new to them by attending extracurricular forms in Informatics. The basic algorithms are presented on the basis of knowledge in mathematics acquired by students in school.

Our long experience has shown that what is difficult to achieve when working with children at an early age is the habit to combine computer skills with their knowledge of mathematics and formal rules of programming languages.

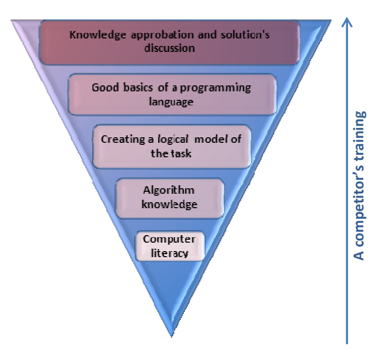

Figure 1. The training steps ascending order.

Preparation of students includes acquiring knowledge and skills of various nature: pure programming, mathematical, technical, and logical one. Insufficient acquisition of some types of knowledge leads to later failures and disappointments. The complex nature of the preparation may initially distress and frighten young students, so new knowledge must be taught gradually and practiced over a long period of time [Hri10, Hri11-1]. The training steps are shown on figure 1 in the ascending order.

The first step for beginners, starting to learn informatics, is acquiring excellent computer literacy.

Through gaming tasks, students must adhere to work with the PC and learn the basic commands of the operating system. The period up to 12 years is the time to provide functional literacy to children. Computer literacy acquisition is essential and it is a prerequisite for their motivation for acquiring the necessary knowledge and skills. The most important practical skills that pupils need to learn are:

- Working with the mouse;
- Knowledge about the location of the letters, numbers and special characters on the keyboard. Excellent knowledge and use of the keyboard (including the transition from one alphabet to another) is mandatory;
- Readiness to start a program in the Windows environment;
- Storing files to removable media (external disk, flash memory, etc.) and manipulation with files from any medium;
- Redirecting input from the standard input device to a file;
- Creating own folders on specified device.

The second but not less important step should be training algorithms. Learners must learn to create algorithms for solving a problem.

The experience gained shows that knowledge of algorithm development, which should be mastered first, is easier to perceive and learn quickly when children are taught by flowchart specifications and step by step animation. This is the way they understand better the nature of the algorithm concept and its properties.

The pupils have to be able to do task condition analysis which includes:

- defining the input data;
- subdividing the main task into groups with elementary operations that lead to the solution;
- making evaluation of the possibilities of reducing some fragments from the base algorithm to the already known elementary algorithms (familiarization with the modularity and object orientation);
- defining the complete results and how they have to be output.

They should learn basic algorithmic structures – sequence, if, switch, while, do-while cycles. They have to get acquainted with the concepts such as variable, array, data type, and the following commonly used basic algorithms:

- Two variables values exchange;
- Working with the coordinates of the point in the plane;
- Finding the maximum of two/three variables;
- Calculating the sum or average value of the numbers as array elements;
- Finding the product of array elements;
- Finding the number of array elements corresponding to a given condition;
- Finding the value and/or the index of the maximum / minimum element in the array;
- Sorting out the array elements;
- Searching in the sorted array;
- Creating a new array with elements of an existing array, satisfying a given condition.

The study of basic numerical algorithms in the early stages of training contributes to better and more durable understanding of the difference between solving a task by hand and using the computer.

The optimization of the range of tasks that young students deal with in the course of their training is one of the most important elements of their overall learning process. The choice of objectives and the coherence of their study largely determine the level and results that can be achieved in the competitions for age group E. The system of tasks and basic numerical algorithms are constantly updated, following the trends outlined in the tasks given at competition for this group. This feedback, from the competitions to the training, contributes to the maintenance of the set of basic numerical algorithms up to date.

The third step - new knowledge should be served in small portions, backed by appropriate examples. In this age group, a lot of attention should be paid to analysis of the condition of the task. It is important for children to understand that behind the many words, in the condition of each programming task lies a logical model which they have to reach as they answer the following questions:

a) "What is given in the task?"

b) "What are you looking for in the task?"

c) "How should you get from the given data to the wanted result?"

Reaching the answers to these questions, the young learners realize that what is given in the condition of the problem is their input data for the algorithm. What is searched in the task description is the information that has to be displayed, and the answer to the third question gives the sequence of actions that leads from the given to what is sought.

The fourth step in the training of informatics in this age group is to teach the basics of a programming language and the principles of programming.

At present, the training is carried out based on the C++ language. The students have to learn the syntax and semantics of this language. They must know very well the programming environment and the successive steps to prepare a working program. The choice of programming environment is determined by the current environment provided by the competitive system that is used in the conduct of competitions in all age groups.

Teaching the language should start in parallel with the training algorithms. This helps studied algorithms to be programmed and immediately tested on a computer. In this way is achieved:

- Reinforcement of the knowledge and skills already acquired;
- Better understanding of the role of algorithms in solving problems in Informatics;
- Maintaining the interest of young children in programming activities.

The students in age group E have to master:

- environment for the programming language;
- standard data types;
- basic arithmetic and logical operations;
- assignment Operators, if, switch and loops;
- redirection to the standard input and output;
- the use of one-dimensional array;
- word processing.

An important component in the small competitors training is filling in the gaps in their mathematical knowledge. Very often for the solving of problems in Informatics it is necessary to use knowledge that is studied later in the school mathematics course. This requires the teacher of the extracurricular classes in informatics to familiarize the young students with this knowledge so that they are able to comprehend and apply it in their programs. In some extracurricular forms in informatics there is a practice the fifth grade students to attend an extracurricular mathematics class in which to study mathematics preemptively.

Some of the most common difficulties in the extracurricular activities in informatics are associated with their different aims in comparison to the aims in the mass oriented school. It is important for the pupils from the beginning of their education to understand the substantial role that the algorithms perform in solving tasks by means of a computer and that the exercises in the basic algorithms would have helped them in their future training.

Another difficulty spotted by us is the inability of many students to decompose a question properly and put it for algorithm development. This difficulty can be overcome by using the mechanism of the “dry run” and needs a lot of practice work.

An important moment in teaching compiling algorithms is that children should be left alone to draw algorithms of the tasks, programmed by them because when participating in contests the children are alone in front of the computer, only with the description of the problem. Sometimes it is enough to tell the student how to use basic algorithms or how to use already known tasks for developing the new task.

During the training in developing algorithms, the teacher must pay special attention to the most frequently made mistakes by the students. Based on our longstanding observations during the work with gifted children from the schools of informatics, the most frequently made mistakes can be grouped in the following way:

1. Mistakes, related to the properties of the algorithms

In the most general case, the algorithms must have the four basic properties: synonymy, determination, terminality and effectiveness. The main difficulties in creating algorithms and making mistakes are connected first of all with non-compliance with some of these properties:

- Improperly determining the sequence of the steps that must be implemented.
- For example, when calculating a sum in a cycle, zeroing of the variable in which we calculate the sum to be located in the body of the cycle.
- Improperly determining the type of tasks to which the algorithm can be applied.

For example, the input data may not be entered by a reading block, but may be assigned to variables within the algorithm itself. In this manner, the algorithm always depends on the same data.

- Unfinished work of the algorithm within the allotted time

For example, the execution reaches an unexpected exit because of a wrong condition to end the cycle, or completion of the algorithm without a written output.

2. Mistakes, associated with an cycle improperly organized

- Loops.
- Wrong number of cycle repeats.
- Changing the parameter's value inside the cycle with the parameter's body.

3. Mistakes, associated with improperly initialized variables

- Non-compliance with the rule for zero adder and counter before a cycle and cleaning with a one when multiplication has been accumulated in a cycle.
- Using a variable before initialization with some value.

4. Errors, associated with incorrect determination of the area of admittable values

- By interval – inside boundaries and outside the interval.
- By a set of values – verification for belonging to the set.

5. Other mistakes

- Division by zero.
- Wrong indexation when trying to create a new subarray from the existing array, fulfilling given conditions.
- Necessity of using unknown mathematical mechanisms like rooting, exponentiation, coordinates to points, etc.

It is important that students learn more about the so-called "protective programming" on the algorithmization level. The algorithm should also process the cases – "This is impossible to happen". For example, to check whether the divisor is different from zero before division; to check whether an index of an array exceeds the limits of the array, etc. This approach is called inclusion of assertion in programming. The assertions, included in the programming code, bring it to behave in foreseeable manner and help to detect the nature and the reason of the problems.

The essence of the proposed method is shown in the figure 2. It consists of learning the basic algorithms in the following sequence [Hri11-2]:

- Introductory task, illustrating the basic algorithm;
- Series of complex tasks in which the underlying algorithm is an integral part;
- The acquired knowledge approbation through work on a task which uses the same basic algorithm and was given in competitions;
- Review and analysis of the proposed solutions and their quality evaluation;
- Offering a recommended solution if the proposed solutions are not the best quality;
- Analysis of the solution proposed by the author;
- Discussion with emphasis on the key points of the basic algorithm usage.

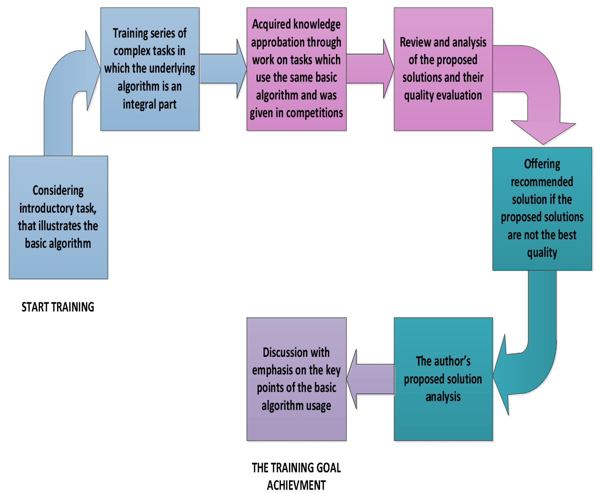

Figure 2. The proposed methodology sequence

The student has to learn the peculiarities of the program text entered via the keyboard. He/she has to be familiar with the programming environment to check how the program works. Reaching the ultimate goal - running program, requires him/her to know the stages through which the programming environment works. During the preparatory phase (compilation), the extent to which the student has mastered the strict rules of the particular programming language is shown. For the next stage (linking), when the program code is associated with the code of other programs taken by the libraries included, the student needs to know about these libraries - what they include and how to use them. The last stage (creation and testing of the executable program code for semantic (logical) errors, he/she must know that it is necessary to implement the program repeatedly, with various well-selected test examples to verify the overall performance of the program.

The proposed methodology includes elements that develop the following:

- Students' independent thinking;
- Reaching a higher level and more complex

structure in the development of capabilities;

- Striving for innovation and originality in problems solving;
- Striving for expression.

The tasks set optimization that the gifted students deal with in the course of their training, is one of the most important elements of the learning process. The choice of the tasks set largely determines the level that the student could reach at the end of his/her training. That is a prerequisite for the system of tasks development for initial training in competitive programming [Hri07].

The sample methodological system comprises 7 subsystems of tasks including the concepts of algorithm, programming language C++, standard input and output operations, linear programs, programs with if, programs with cycles, programs with one-dimensional arrays, processing of string information. These provide the theoretical and practical knowledge and skills necessary for the purposes of training [2].

Each subsystem is built on the following five basic elements:

1) Main objective;
2) Knowledge and skills;
3) Practical work;
4) Sequence of tasks;
5) Methodological features using tasks.

The methodology suggested and the basic system of tasks developed have been successfully applied in the classroom “Initial programming" via "Bistra & Galina" Foundation[1]. The main goal of the training in this classroom is to introduce the children aged between 9 and 13 years to the creation of algorithms and writing simple programs. Via these activities they reveal the beauty of informatics and they are motivated to involve in extracurricular forms in Informatics, which are run at the Center for students’ technical and scientific work in Ruse.

Similar objectives and the methodological materials availability provoked us to use the same set of tasks in the work in the classroom. This necessitated scaling, simplifying and reducing the number of tasks in order to place them in the classes and achieve the goal of education - generating interest and motivation for continuing education.

A specific feature of this training is two times shorter academic hours, but the intensity of the employment is much higher (4 hours per day, 8 days long). There is not enough time to consolidate knowledge and build lasting skills. The emphasis is on creating interest in students, which should grow into motivation for more serious activities in the field of computer work and extracurricular activities in this area. The majority of the children, who participated once in the classroom, continue their activities in this direction in extracurricular classes in informatics.

5. The results

The important criteria for evaluation of the proposed approach for informatics training for beginners is student performance in the following national competitions in Informatics: Winter Competition in Informatics (WCI), National Olympiad in Informatics Final Round (NOI3), Spring tournament in Informatics (STI) and Autumn tournament in Informatics „John Atanasov” (ATI)[2].

Table 1 presents the results of Ruse students in chronological order.

Table 1. The results for the smallest group E

Year	Competition	Total number of participants	Number of participants from Ruse	Number of prizes
2007	WCI	13	2	2
	STI	25	4	2
	ATI	24	2	0
2008	WCI	17	1	0
	STI	20	1	1
	ATI	13	2	0
2009	WCI	18	1	0
	STI	21	2	0
	ATI	31	2	1
2010	WCI	22	1	1
	STI	20	2	2
	ATI	31	3	0
2011	WCI	17	1	1
	NOI3	8	1	1
	STI	21	3	0
	ATI	26	2	2
2012	WCI	18	2	2
	NOI3	12	1	0
	STI	25	2	1
	ATI	16	2	1
2013	WCI	16	2	1
	NOI3	19	1	0
	STI	35	2	0
	ATI	43	2	1
2014	WCI	34	2	2
	NOI3	20	1	0
	STI	29	2	0
	ATI	32	2	1

Figure 3 shows a graph of the students attracted to the Center for students work in Ruse (CUTNT)[3] by the proposed methodology, prizes from competitions

[1] http://www.bgfound.org/ bg

[2] http://www.math.bas.bg/infos

[3] http://cutnt-ruse.com

summed up in years. The line of behavior shows a trend of steady growth. Assuming that the number of the prizes won at national level is a measure of the quality of education and level of training of athletes at this basic level, it can be concluded that the proposed methodology combined with the correct selection of tasks for training helps to increase quality continuously.

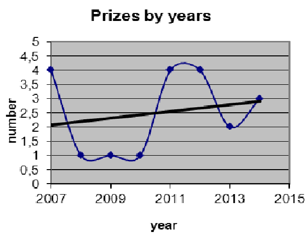

Figure 3. Prizes by years

Figure 4 represents the quality of the training proposed approach in its purest form, because of the highest possible competition during the national round of the NOI, objective selection of participants and the fact that these races were held during the approbation of the overall methodology.

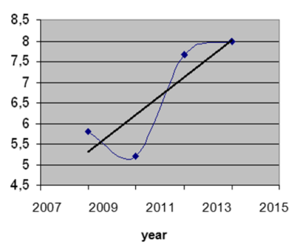

Figure 4. Participation in the national rounds of OI

6. Conclusions

The analysis of the results, which have been obtained by application of the suggested methodology, shows that:

1. In the initial training program students assimilate the algorithmic thinking principles and basic algorithmic structures much better when they are illustrated with the tasks that process numerical information. As a basis for teaching and learning the new concepts of informatics are used tasks from mathematics, in which there are no unknown mathematical concepts. Thus the attention of younger students is directly aimed at rationalization of learning new material.
2. The suggested system of tasks is scalable and applicable by other groups for basic training in informatics.
3. A very important and integral part of this methodology is a system of tasks that must constantly be updated with new ones.

7. References

[Gar08] Garov, K. Theory and practice of preparation of prominent and talented students to participate in contests and competitions in informatics and information technology, *Dissertation for awarding the educational and scientific degree "doctor"*, Sofia, 2008. (in Bulgarian)

[Hri11-1] Hristova P., G. Atanasova, Preparation of heads of schools in Informatics, *National Conference „Education in Informational society"*, Plovdiv, 2011, p. 303-310, ISBN 1314 0752

[Hri07] Hristova P., Training through tasks, *Proceedings of the University of Rousse "Angel Kanchev" Scientific Conference - 2007*, Volume 46, Book 6, Mathematics, Informatics, Physics, Pedagogy and Psychology, p. 50-55, ISSN 1311-3321

[Hri10] Hristova P., The Informatics Competitions for Group E During the School 2009-2010 Year, *Proceedings of the Anniversary International Conference Celebrating Prof. Sava Grozdev's 60th anniversary*, Bachinovo, Bulgaria, September 10-12, 2010, p.409 – 417

[Hri11-2] Hristova, P., Best practices in training informatics competition participants, *Proceedings of the 7th Annual International Conference on Computer Science and Education in Computer Science 2011*, Sofia/Dobriniste, Bulgaria, ISSN 1313-8624, p. 371-384

[Knu73] Knuth D., *The art of computer programming, Addison-Wesley Publishing Company*, Ontario, 1973

[Wir75] Wirth N., *Algorithms Plus Data Structures Equals Programs*, Prentice-Hall, 1975

[Yov09] Yovcheva, B. Spiral approach to teaching programming of 10-11 year-olds, *Dissertation for awarding the educational and scientific degree "doctor"*, Shoumen, 2009. (in Bulgarian)