

REPORTS

Gomel Training School for Olympiads in Informatics

Michael DOLINSKY

*Department of Mathematics, Gomel State University “Fr. Skaryna”
Sovetskaya str., 104, Gomel, 246019, Republic of Belarus
e-mail: dolinsky@gsu.by*

Abstract. The article describes technology used to teach programming and preparing for olympiads in informatics. Remarkable feature is a balanced education of four age ranges (preschoolers, grades 1–4, grades 5–8, grades 9–11) in four directions – thinking, mathematics, programming, algorithmization – and permanent Internet competitions to improve motivation. Distance learning system DL.GSU.BY is the effective technical base of the teaching.

Keywords: programming teaching, olympiad in informatics, distance learning tools.

1. Introduction

The author has been training pupils of different ages for the in programming and preparing them for olympiads in informatics in Gomel School 27 since September of 1996 and with help of the Distance Learning system DL.GSU.BY (further DL) since September of 1999.

The primary goals and objectives of this process are as follows:

- To develop in each child such properties as willingness to learn, analytical skills, self-dependence, and creativity.
- To give each child the base computer literacy.

- To help each child to understand what is “programming” and to decide whether his/her professional future will be with computer programming or without it.
- To prepare everybody who invests sufficient time to reach medals of national and international olympiads in informatics.

The results of this work during 1996–2015 are reflected in the following achievements of pupils of Gomel and Gomel region (Performance Statistics, 2015):

- More than one hundred pupils chosen programming as profession.
- Dozens of pupils entered universities without exams.
- 189 diplomas and 32 honorable mentions from national Olympiad in Informatics of Belarus.
- 37 diplomas of international collegiate programming contests (Sankt-Petersburg, Russia).
- 8 gold, 11 silver and 8 bronze medal of International Olympiad in Informatics (IOI).

Table 1 below lists the medal results from IOI 1997–2015 of the teams of different regions of Belarus as well as Minsk team and team of the Lyceum of Belarus State University separately represented on national Olympiad in Informatics of Belarus.

Important hallmark of the Gomel training school for olympiads in informatics is the start of learning as early as possible. It provides earlier (grade 10 or earlier) successes in the olympiads including IOI (all from city of Gomel unless noted otherwise):

- Aliaksei Danchanka (grade 9): 1998 – participation, 1999 – silver, 2000 – silver.
- Mikhail Svarychevski (10): 2000 – silver, 2001 – bronze.
- Raman Dzvinkouski (9): 2002 – bronze.
- Uladzimir Miniailau (9): 2005 – silver, 2006 – gold, 2007 – silver.
- Henadzi Karatkevich (5): 2006 – silver, 2007–2012 – gold.
- Uladzislau Padtsiolkin (9): 2011 – silver, 2012 – silver, 2013 – bronze.
- Siarhei Kulik (9, from Mozyr¹): 2001 – silver, 2012 – bronze, 2013 – gold.

Table 1
Medals from the national olympiad

Region	Total	Gold	Silver	Bronze
Gomel region	28	9	11	8
Minsk region	11	1	5	5
Lyceum of BSU	9	1	3	5
Vitebsk region	7	0	3	4
Minsk	5	0	3	2
Grodno region	4	0	2	2
Brest region	1	0	0	1
Mogilev region	0	0	0	0
Total	65	11	27	27

¹ Mozyr – town in Gomel Region.

- Adam Bardashevich (10, Mozyr): 2001 – silver, 2012 – gold.
- Fedar Karabeinikau (8): 2014 – participation, the first under bronze cutoff.

Unconditionally, their success is based on their own hard work as well as their parents' efforts on education and the creation of conditions for rapid growth. In addition, the success of Mozyr pupils is strongly connected to Alexey Borunov, Mozyr coach for programming contests. Nevertheless, the author believes that the education environment described in this paper essentially helped everybody mentioned above and many other medalists of national and international olympiads in informatics to achieve such remarkable results.

The remainder of the paper is structured as follows: Chapter 2 represents the author's educational principles; Chapter 3 describes the elaborated education courses, exercise packages and permanent Internet competitions; Chapter 4 contains the author's educational strategy; Chapter 5 represents Gomel Regional Olympiad in Informatics; Chapter 6 is devoted to newly elaborated "Accelerated Learning" approach to teaching; finally Chapter 7 contains conclusions.

2. The Author's Educational Principles

This chapter describes the principles used by author to organize the teaching process.

Constancy. Lessons are held in informatics cabinet of Gomel School 27 every Wednesday and every Sunday, even on holidays (including summer). When the author is absent from Gomel, he gets somebody to fill in for him. The site DL.GSU.BY supports the education process around the clock, so everyone can study anytime and anyplace at their own convenience.

Inclusiveness. The author does not deny anyone who comes to learn. Now the lessons are organized in such a way that one can begin studying not only from the first grade of school but also from preschool ages – as young as 4 years (Dolinsky, 2013).

Efficiency. The main criteria of efficiency of lessons is the ratio of the study time to total usage time. The author tries to organize the study so that each pupil works hard each minute of each lesson, either learning something new or consolidating their skills.

Individuality. The author has for many years believed that efficiency in the sense described above can be achieved only through individual and differentiated instruction. That is, each pupil moves on at their own rate and accordingly at any given time all pupils can be at different points of the learning process.

Self-dependence. Obviously such individuality can be achieved only through independence of learning, since a teacher physically can't immediately teach (to explain, to listen, etc.) all pupils simultaneously, particularly if the pupils work with different materials. Furthermore, the self-dependence is an important objective of the education. By and large, the author's view is that teaching independence is even more important than teaching any specific knowledge. It is especially true in the area of programming where one needs to learn and relearn all professional life.

Optimality (of material selection). The author tries to select the most useful material and construct the process of studying it in the most effective system. The author can't claim to have achieved perfection, but it is true that he does try to teach today better than yesterday and tomorrow better than today.

Demands (to comply with the rules). Unfortunately, the list of rules is ever expanding, complicating the compliance: keep silence in the audience; know where the classes are held; bring the notebook and pen; come to class before lesson starts, leave when the lesson is over; do exercises himself, (don't cheat); etc.

Using the site DL.GSU.BY. Since 1999, the author has headed the development of this project and actively used it for teaching. This approach has the following advantages:

- Fast verification of solutions (from a few seconds to a few minutes) and as a result – multiplying the intensity of the teaching process.
- Variety of task types – differentiating the study and keeping interest of students with different levels of preparing and motivation.
- Automatic presentation of tasks and differential study (Dolinsky, 2012) – the technical basis for customized education.

Teacher activity. The author tries to organize the studying process in such a way that each student works independently as much as possible.

3. Educational Courses, Exercise Packages, and Permanent Competitions

The teaching process is based on using the distance learning system DL and includes the following learning and training courses: “Programming-professionals (individual)”, “Programming-professionals (collegiate)”, “Programming-beginners”, “Programming-professionals (individual) work on errors (W/E)”, “Programming-professionals (collegiate) W/E”, “Programming-beginners W/E”, “Preparing for IOI”, “Methods of algorithmization”, “Basic programming”, “Start to program”, “Informatics”, and “Mathematics”.

The course “**Programming-professionals (individual)**” is open from beginning of December to middle of July/August – until IOI. It contains weekly Sunday 5-hour (9:00–14:00 GMT+3) individual contests – tasks from past olympiads of regional level until the first decade of January (when Gomel regional olympiad in informatics is held), then from olympiads of national level until end of March (when national olympiad in informatics of Belarus is held), and from olympiads of international level until the close.

The course “**Programming-professionals (collegiate)**” is open from the middle of July/August (after IOI is finished) until the end of November, when Collegiate Programming Contest in Sankt-Petersburg is held. It has weekly Sunday 5-hour (9:00–14:00 GMT+3) contest with tasks from past collegiate contests.

So, the whole year (including holidays) every Sunday there is 5-hour individual or collegiate open on-line contest for all comers. Immediately after the olympiad is finished the tasks become available for practice in courses “**Programming-professionals (individual) (W/E)**” and “**Programming-professionals (collegiate) W/E**”, respectively.

For participants in informatics cabinet of Gomel School 27, there is additionally interactive discussion of solutions.

Tasks are classified by themes and copied into appropriate branches of the course **“Methods of algorithmization”**. Thus, the latter is adjusted to the trends in the development of competitions and provides targeted preparation for olympiads. In addition, the tasks that were not solved are copied to the course **“Preparing for IOI”**.

To activate self-study and practice in the courses “Methods of algorithmization” and “Preparing for IOI” we hold the permanent Internet competitions: Autumn Cup, Winter Cup, Spring Cup, Summer Cup, and Whole Year Prize. In the Cups we award the three the best pupils who solve most tasks for the season (autumn, winter, spring, summer). In the “Whole Year Prize” we awarded the one pupil who solves the most tasks for the whole year (autumn-winter-spring-summer). In the course “Preparing for IOI” it is insufficient to solve one task, but in addition one needs to describe their solution in the DL forum corresponding to the subject of the task.

The courses “Programming-professionals (individual) (W/E)”, “Programming-professionals (collegiate) W/E”, “Methods of algorithmization”, “Preparing for IOI”, feature “Tests grants”. That is, a pupil can get the test data (input data, model answer, and program-checker) on which their solution failed. There’s a special FAQ and a dedicated DL forum to support pupils in the situation “I got the right answer on my computer, but my solution was rejected”.

For individual contests (after their end) as well as for all the courses, there are permanent result tables with links to solutions of all participants (as images to prevent copying and resubmitting).

The DL forums have links to authors’ systematic descriptions of solution as well as to descriptions by pupils of Gomel and university students.

The author elaborated a set of tutorials, two from them have been published as books in Sankt-Petersburg (Dolinsky, 2005, 2006).

Another important aspect is the joint participation of pupils and university students in the weekly Sunday olympiads where students are training for ACM ICPC (only for quarter-finals and semi-finals so far).

The weekly Sunday olympiads as well as practice in other weekdays is targeted towards medalists of national olympiad in informatics preparing for International Collegiate Programming Contests and IOI, as a rule pupils of grades 9–11.

For pupils of grades 5–8 preparing for regional olympiads is the course **“Basic programming”** with possibilities of automatic task presentation and differential study. The exercise system has tree-like structure. Correct solution provides transition to the next exercise. Wrong solution or pressing the button “I don’t know” transitions the student into subtree teaching to solve the problem. Own teaching subtrees may be settled for any such teaching tasks. This way we provide individual differential teaching that adapts not only to the level of preparation of a pupil but also to their current emotional and physical state. Pupils who know more and are in better shape branch less and thus advance faster. The course “Basic programming” has exercise packages with differential teaching on the following topics: introduction to programming, one-dimensional arrays, two-dimensional arrays, geometry, strings, sorting, queues. To increase the motivation

of beginners, there are weekly Sunday olympiads in the course “**Programming-beginners**” from 7:00 to 20:00 (GMT+3). One can solve the same tasks for practice in the course “**Programming-beginners W/E**”.

For pupils in grades 1–4 there is the learning course “**Start to program**” that largely contains the same tasks as “Basic Programming”, but in essentially linear form. Most of teaching trees are reorganized into sequences of tasks for simpler learning. There are permanent Internet-competitions “Season Cups” and “Whole Year Prize” for learning courses “Basic Programming” and “Start to program”. Note that in the competitions the tasks from the learning subtrees are not counted. In the course “Start to Program” only pupils of grades 1–4 and in the course “Basic Programming” only pupils of grades 1–8 are awarded.

The training course “**Informatics**” was created for pupils of grades 1–4 of Gomel School 27. Originally there was only one package of exercises: “Learning to think” (Dolinsky, 2013). But soon we started to copy exercises from “Start to Program” to for more advanced pupils in “Informatics”. Moreover, we re-open “Informatics” every academic year, which gives more possibilities to change it. So, now the course “Informatics” has practically the same material as the course “Start to Program” has, but in a better methodological form. The course “Informatics” also has permanent Internet-competitions “Season Cups” and “Whole Year Prize” where pupils of grades 1–4 are awarded.

The “Basic Programming”, “Start to program”, and “Informatics” courses are based on the Pascal programming language.

Finally, the course “**Mathematics**” contains different tasks in mathematics, including such exercise packages as flash tasks on mathematics of grades 1–5, tasks from the international mathematics contest “Kangaroo” (2001–2015, all grades), Canadian Math Contests (1998–2015, grades 7–11), “Math from informatics”. In the latter, the tasks are from the course “Programming-beginners”, but reformatted so that the pupil needs to manually enter the answer for given input data. The answers can be computed manually or using an appropriate program. For the course “Mathematics” there also are also permanent Internet-competitions “Season Cups” and “Whole Year Prize” where pupils of grades 1–8 are awarded.

Tables 2 and 3 represent participation statistics from 2008 to 2014.

Table 2

	2008/09	2009/10	2010/11	2011/12	2012/13	2013/14	2014/15
Mathematics	–	–	–	–	146	175	214
Informatics	–	–	–	249	301	179	212
Start to program	22	182	142	118	174	189	300
Basic programming	10	263	205	159	290	309	468
Methods of algorithmization	205	269	340	332	491	425	441
Preparing for IOI	31	25	8	7	12	17	3
Total	237	739	695	820	1424	1294	1638

Table 3

	2008/09	2009/10	2010/11	2011/12	2012/13	2013/14	2014/15
Programming-prof. (ind.)	98	116	128	165	157	141	133
Programming-prof. (col.)	51	59	73	101	55	113	71
Programming-beginners	77	67	59	89	108	69	211

4. The Educational Strategy

Currently we are providing education for four age ranges: preschoolers (from 4 years), junior school (grades 1–4), middle school (grades 5–8), and high school (grades 9–11), in four directions: thinking, mathematics, programming, algorithmization:

- **Thinking.** The author believes that development of thinking needs to be first to provide higher learning efficiency. For preschoolers and juniors the training course “Informatics” has special packages of exercises “Learning to think”, “Differences”, “Analogy”, “Learning to Count”, and “Tangram”. The courses “Start to Program” and “Basic Programming” have embedded exercises for development of thinking.
- **Mathematics.** It is clear that knowledge of mathematics is important in itself, but it is more important that development of abilities to solve mathematical problems automatically develops effective thinking skills. On one hand, there is training course “Mathematics” containing different mathematical tasks for pupils from grade 1 to grade 11. On the other hand, training courses “Informatics”, “Start to Program”, “Basic programming” have special packages of exercises where mathematics and programming are integrated. For example, the course “Informatics” has packages “Math (programs)” for grades 1–5. Each of them contains tasks from appropriate mathematics textbook, converted to programming tasks by parameterization. To solve the task, one need to solve it mathematically and then write the corresponding program. The course “Basic Programming” contains packages “Kangaroo grades 3–4” (2001–2008) and “Kangaroo grades 5–6” (2001–2009) where appropriate mathematics tasks are converted into programming tasks.
- **Programming.** We are using the Pascal programming language to teach pupils in grades 1–8. But the language is a means, not a goal. Special attention is given to debugging technology as well as structuring of program sources to improve their readability and understandability.
- **Algorithmization.** Best of all we are trying to develop skills for algorithm elaboration. The second direction is study of standard algorithms. Both directions are supported by weekly olympiad solving as well as the following practice on unsolved tasks and studying the needed theory. In addition, the course “Methods of algorithmization” is strongly structured around themes and subthemes and contains information about task sources (year, country, etc.).

5. Regional Olympiads in Informatics

Since 2010 the tasks for the Gomel regional olympiad that qualifies 15 best participants to the national olympiad are created by the national Scientific Committee. But in Gomel region we have at least five olympiads whose tasks we prepare ourselves: in autumn two olympiads (school and city level) for grades 1–11 and in spring three olympiads (school, city, and regional level) for grades 1–9. For those five olympiads we introduce three age divisions: in autumn grades 1–4, grades 5–8, grades 9–11; and in spring grades 1–4, grades 5–7, grades 8–9. We use the tasks of olympiads for:

- a. Targeted training for higher level olympiads.
- b. More accurate indication to students and teachers of what and how to learn.

Tasks for grades 9–11 (in spring grades 8–9) include three groups of increasing complexity (note that each pupil needs to try solve all tasks from all groups):

- *First group (5 tasks)* – tasks on topics from the course “Basic Programming” (one-dimensional arrays, two-dimensional arrays, geometry, strings, greedy (simple algorithm based on sorting)).
- *Second group (5 tasks)* – tasks based on the course “Methods of algorithmization”: queues, recursion, dynamic programming, graphs, and brute force.
- *Third group (2 tasks)* – tasks from the course “Preparing for IOI”, as a rule on the following topics (or their combinations): research, complex dynamic programming, complex data structures, and complex problems on graphs.

This approach allows participation “with interest” for pupils who only began to study programming (first group of tasks). At the same time we can determine the preparation level of those who spent more time (second group of tasks) and point everyone to the topics that need more detailed work. Finally the third group of tasks provides engagement for all 5-hours for the most prepared students, the strategic goal for whom is preparing for international olympiad in informatics.

Tasks for grades 5–8 (in spring grades 5–7) also include three groups of increasing complexity:

- *First group (5 tasks)* – tasks on topics from course “Basic Programming”: introduction to programming, one-dimensional arrays, two-dimensional arrays, geometry, sorting.
- *Second group (3 tasks)* – strings, story problem, research task.
- *Third group (2 tasks)* – implementation task from a real olympiad, queues.

For the first group the simplest tasks from each topic are chosen. It is a way to check whether a pupil has studied the topic at all, because the task has a minimum of text and is standard for the topic.

The second group of tasks is to differentiate pupils (who have the preparation level to solve tasks from the first group) on the skills needed to solve the olympiad tasks. *The first important such skill* is development and debugging of own algorithms. The string processing task is one where the problem formulation is very simple and easy to understand. The main difficulty is to formulate the solution process (algorithm) in a

programming language. *The second important skill* for solving is the ability to read and understand the task text, to differentiate the important, minor, and insignificant things, to reformulate the problem in mathematical and programming terms. So our story problem is the task with detailed (or even cumbersome) formulation but with very simple solution. We use simplified versions of tasks from national olympiads. *The third important skill* is ability to research the solution space. The task statement may be very brief so that it is clear what needs to be done. The main problem is to invent a way to get the result. To get such tasks, we reformulate in programming terms tasks from international mathematics contest “Kangaroo” (grades 5–6).

Finally, the third group of tasks checks students’ readiness to solve original problems from real olympiads. First task is on implementation details from one the following topics: one-dimensional arrays, two-dimensional arrays, geometry, strings. The second is an original task on queues.

Tasks for grades 1–4 also include three groups of increasing complexity:

- *First group (10 tasks)* includes tasks from “Introduction to programming” (three tasks on integers, one task on each of characters, strings, length of strings, number of characters in string) and three tasks on using Pascal standard subroutines DELETE, COPY, POS to delete and copy strings and to find position one string within another.
- *Second group (5 tasks)* includes tasks on one-dimensional arrays: sum of all elements, number of elements with some feature, minimal/maximal elements, search for the first element with some feature.
- *Third group (5 tasks)* is to differentiate more prepared students and includes the tasks on the following topics: two-dimensional arrays, geometry, strings, research (base on “Kangaroo” tasks for grades 2–3), and a story problem.

Systematic and purposeful development of tasks for regional olympiads is an important means of improving the student and teacher preparation in the region.

Note that all five regional olympiads in all three age divisions are on-line events, so usually students from all over Belarus participate in them.

6. Accelerated Learning

To improve the teaching results, recently a few new packages of tasks and exercises were introduced:

- **“Accelerated course – 2013”** includes the following topics: introduction to programming, one-dimensional arrays, two-dimensional arrays, geometry, sorting, strings, story problem, research task. Each topic includes three parts: theoretical minimum, tasks from Gomel olympiad for grades 1–4, tasks from Gomel olympiads for grades 5–8. Such approach allows the most capable pupils to move with maximal speed. At the same time pupils who encounter difficulties in this course can use the standard learning approach.

- **“Olympiads grades 5–8 by topic”** includes tasks from corresponding Gomel olympiads, grouped by topics. It allows to see potential rating of each student and it definitely shows to him and to his teacher the areas for further work.
- **“Olympiads grades 9–11 by topic”** is similar to the above in composition and purpose. It includes additional topics such as recursion, dynamic programming, graphs, brute force, and more complex tasks (with complexity level corresponding to national and international olympiads).
- **“Belarus olympiads”** includes tasks from national olympiads of Belarus (qualification and final stages) grouped by topic and in the order of increasing complexity. Note that we include in that course a special set of tasks with incomplete solutions. To get the full solution of such task one needs to know special theory as well as have good skills of developing and debugging complex algorithms. At the same time the olympiads up to IOI don’t demand full solutions for all tasks because winners are defined by sum of points. So it is important to develop skills for solving tasks partially. We gather such tasks into special theme “Incomplete solutions” and remove the tests cases that can’t be solved by partial solutions. Despite the simplicity, such solution can get from 20 to 80 points and can essentially improve the final result of a contestant in an olympiad. Moreover such simplified solutions may be useful to verify full solution.

We believe that development of these courses is essential not only for preparing to olympiads but also to identify “flaws” in the education system with a view to their eventual elimination.

Note that the author has written and documented the solutions for all tasks (for medal minimum points) from the course “Belarus olympiad” and also for more complex tasks for courses “Olympiads grades 9–11 by topics” and “Olympiads grades 5–8 by topic”. So, if a pupil can’t solve some task there are two options: ask for help from somebody who already solved it or read the description of the author’s solution.

The shortest way for a pupil from grades 5–8 to reach the medal level of national olympiad in informatics of Belarus is: “Accelerated course”, “Olympiads 9–11”, “Belarus Olympiads”.

7. Conclusion

We represented the current state of the system for preparing students of Gomel and Gomel region for olympiads in informatics as well as strategic development directions.

Note that the system is also used by pupils outside of Gomel region. A remarkable feature is permanent monitoring of preparation state for all pupils as well as a balanced education of four age ranges (preschoolers, grades 1–4, grades 5–8, grades 9–11) in four directions: thinking, mathematics, programming, algorithmization.

To improve motivation for ongoing education, we organize the permanent Internet competitions (Autumn-Winter-Spring-Summer, Whole Year).

References

- Dolinsky M. (2013). An approach to teach introductory-level computer programming. *Olympiads in Informatics*, 7, 14–22.
- Dolinsky M. (2014). Technology for the development of thinking of preschool children and primary school children. *Olympiads in Informatics*, 8, 63–68.
- Dolinsky M. (2005). *Algorithmization and Programming with TURBO PASCAL: From Simple to Olympiad Problems: Tutorial*. Sankt-Petersburg “Piter”. (In Russian).
- Dolinsky M. (2006). *Solving of Sophisticated Olympiad Programming Problems: Tutorial*. Sankt-Petersburg “Piter”. (In Russian).
- Performance Statistics of Gomel Pupils in International and National Olympiads in Informatics from 1997 to 2015*. (In Russian).
<http://dl.gsu.by/olymp/result.asp>



M. Dolinsky is a lecturer in Gomel State University “Fr. Skaryna” from 1993. Since 1999 he is leading developer of the educational site of the University dl.gsu.by. Since 1997 he is heading preparation of the pupils in Gomel to participate in programming contests and olympiads in informatics. He was the deputy leader of the team of Belarus for IOI’2006, IOI’2007, IOI’2008 and IOI’2009. His PhD is devoted to the tools for digital system design. His current research is in teaching Computer Science and Mathematics from early age.