Olympiads in Informatics, 2025, Vol. 19, 189–208

DOI: 10.15388/ioi.2025.13

School Startup in Olympiad in Informatics

Marina S. TSVETKOVA, Vladimir M. KIRYUKHIN
Russian Academy of Natural History, Russian Federation, Moscow, 105037, box 47
e-mail: ms-tsv@mail.ru, vkiryukh@gmail.com

Abstract. The article describes the modern content of an advanced school course in Informatics (K 7-9 and K 10-11), corresponding to the new educational standard implemented in Russia since 2023. This content is in full compliance with the IOI Syllabus. This allows us to build effective preparation programs for various Olympiads in informatics, including the International Olympiad in Informatics (IOI).

Keywords: school education, curriculum in Informatics, Olympiad in Informatics, Modern Content of an advanced school course in Informatics (K 7-9 and K 10-11), competencies of Olympiad in Informatics

1. Introduction

As a rule, updating educational standards for schools in any country is aimed at developing the content of education, updating it considering the development of science and technology. This is especially important for the subject of informatics, which is the basis for the digitalization of professions, and reflects the digital transformation of the economy and science in the country.

In Russia, school informatics has already gone through several stages of development: it started in schools in 1985 as a subject based on Algorithmics. At the suggestion of the Academy of Sciences, in the 90s the subject was supplemented with programming and a section on information technology, which was facilitated by the appearance of personal computers in schools. An advanced course in informatics, filled with topics on the theoretical foundations of informatics, was included in schools for grades 10–11 since 2005 (Kiryukhin, 2010), but mainly reflected topics for the Unified State Exam for admission to universities. It is important that this exam is an elective exam for senior schoolchildren in technological or physical and mathematical profiles, in which the course in informatics was studied as an advanced one.

Education in informatics remained within the framework of the basic program for schoolchildren in grades 7/8–9 until 2022. Only electives and additional education courses could compensate for the needs of children interested in informatics. In 2022, as part of the new federal educational standard (Educational Standard, 2022) advanced

courses in informatics were included in the course in informatics in both basic and high school at the federal level, and in them, updated sections of theoretical foundations of informatics, algorithms and programming, which reflect the topics of Olympiad in Informatics.

Based on the new educational standard, since September 2023 federal programs for basic general education (Curriculum Advanced Informatics K7-9, 2023) and for high school (Curriculum Advanced Informatics K10-11, 2023) have been in effect. This allowed schoolchildren involved in Olympiad preparation to choose to study an advanced course in informatics from the 7th grade (14 years old) free of charge using the appropriate textbooks provided by the state.

Currently, the Russian School Olympiad in Informatics is held considering the current advanced standard for the subject, and the winners and prize winners of the Olympiad finals receive the right to enter without exams for free education at any university in the country in the Olympiad profile and successfully participate in international Olympiads. It cannot be denied that this standard and federal programs in informatics for basic general education and high school have played a significant role in this.

Below are sections of the advanced course in informatics for primary and high school, the sections of which fully reflect the content of Olympiad training in informatics and can determine work with schoolchildren in preparation for various levels Olympiads, including international Olympiads and the IOI.

2. Theoretical Foundations of Informatics. Information Structures (Graphs, Trees)

This section covers topics of graph descriptions and graph problems, using the example of a game tree.

K7-9	**K10-11**
Graph. Vertex, edge, path. Directed and undirected graphs. Length (weight) of an edge. Graph weight matrix. Length of a path between graph vertices. Finding the optimal path in a graph. Starting vertex (source) and ending vertex (sink) in a directed graph. Calculating the number of paths in a directed acyclic graph. **Tree.** Root, vertex (node), leaf, edge (arc) of a tree. Height of a tree. Subtree. Examples of using trees. Enumeration of options using a tree.	**Graphs.** Basic concepts. Types of graphs. Description of graphs using adjacency matrices, weight matrices, adjacency lists. Solving algorithmic problems related to graph analysis (constructing an optimal path between graph vertices, determining the number of different paths between vertices of a directed acyclic graph). **Trees.** Binary tree. Search trees. Methods of tree traversal. Representation of arithmetic expressions as a tree. Discrete games of two players with complete information. Construction of a tree of enumeration of variants; description of the game strategy in tabular form. Winning and losing positions. Winning strategies.

3. Theoretical Foundations of Informatics. Data Coding, Data Compression

This section shows the basics of discrete data representation and theoretical approaches to assessing the amount of information. For senior school, the topic of number representation in computer memory is considered, as well as methods of applying coding to data compression, encryption during archiving or transmission over communication channels.

Theoretical foundations of informatics Information is one of the main concepts of modern science. Discreteness of data. The ability to describe continuous objects and processes using discrete data. Information processes are processes associated with storage, transformation and transmission of data. **Symbol. Alphabet. Alphabet power** Diversity of languages and alphabets. Natural and formal languages. Alphabet of texts in Russian. Binary alphabet. Number of different words (code combinations) of fixed length in binary alphabet. Transformation of any alphabet to binary. Number of different words of fixed length in alphabet of certain power. Coding of symbols of one alphabet using code words in another alphabet; code table, decoding. **Binary code** Representation of data in a computer as texts in a binary alphabet. Information volume of data. Bit is the minimum unit of information amount – binary digit. Byte, kilobyte, megabyte, gigabyte	**Theoretical foundations of informatics** Information, data and knowledge. Information processes in nature, technology and society. Continuous and discrete quantities and signals. The need for discretization of information intended for storage, transmission and processing in digital systems. **Binary coding** Uniform and non-uniform codes. Decoding messages written using non-uniform codes. Fano condition. Construction of uniquely decodable codes using a tree. Al. A. Markov's graph. Units of measurement of information quantity. Alphabetical approach to assessing information quantity. Theoretical approaches to assessing the amount of information. Law of additivity of information. Hartley's formula. Information and probability. Shannon's formula.
Positional and non-positional number systems Alphabet. Base. Expanded form of writing a number. Conversion of numbers written in other number systems to the decimal system. Roman numeral system. Binary number system. Conversion of natural numbers to the binary number system. Octal number system. Conversion of numbers from the octal system to the binary and decimal systems and back. Hexadecimal number system. Conversion of numbers from the hexadecimal system to the binary, octal and decimal systems and back.	**Number systems** Expanded notation of integers and fractions in the positional number system. Properties of positional notation of a number: number of digits in the notation, divisibility of the number by the base of the number system. Algorithm for converting an integer from the P-number system to the decimal number system. Algorithm for converting a finite P-number fraction to the decimal number system. Algorithm for converting an integer from the decimal number system to the P-number system. Converting a finite decimal fraction to the P-number system.

Arithmetic operations in the binary number system. Representation of integers in P-number systems. Arithmetic operations in P-number systems.	Binary, octal and hexadecimal number systems, the relationship between them. Arithmetic operations in positional number systems. *Balanced ternary number system. Binary-decimal number system*.
	Representation of integers in computer memory Limited range of numbers with limited number of digits. Overflow of the bit grid. Unsigned and signed data. Signed bit. Binary complement code of negative numbers. Bitwise logical operations. Logical, arithmetic and cyclic shifts. Encryption using the bitwise operation «exclusive OR». **Representation of real numbers in computer memory** Significant part and order of number. Range of values of real numbers. Problems of storing real numbers related to limitation of the number of digits. Carrying out operations with real numbers, accumulation of errors during calculations.
Text encoding Uniform code. Non-uniform code. ASCII encoding. Eight-bit encodings. The concept of UNICODE encodings. Decoding messages using uniform and non-uniform codes. Information volume of text. **Color coding** Color models. RGB, CMYK, HSL models. Coding depth. Palette. Raster and vector representation of images. Pixel. Evaluation of the information volume of graphic data for a raster image. **Sound coding** Bit depth and sampling frequency. Number of recording channels. Evaluation of the information volume of sound files.	**Text encoding** ASCII encoding. Single-byte encodings. UNICODE standard. UTF-8 encoding. Determining the information volume of text messages. **Image encoding** Estimating the information volume of graphic data at a given resolution and color encoding depth. Color models. Vector encoding. Graphic file formats. Three-dimensional graphics. Fractal graphics. **Sound encoding** Estimating the information volume of audio data at a given sampling frequency and encoding bit depth.
Data transfer rate Units of data transfer rate. Data distortion during transmission.	**Data transfer rate** Transfer time dependence on the information volume of data and characteristics of the communication channel. Causes of data transfer errors. Codes that allow detecting and correcting data transfer errors. Hamming's distance. Bit repetition coding. Hamming's co-des.

	Data compression algorithms RLE algorithm. Huffman algorithm. LZW algorithm. Lossy data compression algorithms. Reducing the color coding depth. Basic ideas of JPEG, MP3 compression algorithms. **Data Encryption** Symmetric and asymmetric ciphers. Simple substitution ciphers. Caesar's cipher. Vigenere's cipher. RSA encryption algorithm. *Steganography*.

4. Theoretical Foundations of Computer Science. Logical Algebra

This section, in addition to the theoretical foundations of the logic algebra, which are important for constructing algorithms, shows the logical foundations of a computer and engineering approaches to constructing circuits on logical elements.

Logical statements Logical meanings of statements. Elementary and compound statements. Logical operations: «AND» (conjunction, logical multiplication), «or» (disjunction, logical addition), «NOT» (logical negation), «exclusive or» (addition mo-dulo 2), «implication» (consequence), «equi-valence» (logical equivalence). Priority of logical operations. Determining the truth of a compound statement given the known truth values of the elementary statements it contains. **Logical expressions** Rules for writing logical expressions. Construction of truth tables of logical expressions. Simplification of logical expressions. Laws of Boolean algebra. Construction of logical expressions according to the truth table. Logical elements. Introduction to the logical foundations of a computer. Adder.	**Algebra of logic** Concept of proposition. Propositional forms (predicates). Quantifiers of existence and universality. **Logical operations** Truth tables. Logical expressions. Logical identities. Proof of logical identities using truth tables. Logical operations and operations on sets. **Laws of algebra of logic** Equivalent transformations of logical expressions. Logical equations and systems of equations. **Logical functions** Dependence of the number of possible logical functions on the number of arguments. Complete systems of logical functions. Canonical forms of logical expressions. Perfect disjunctive and conjunctive normal forms, algorithms for constructing them using a truth table. **Logical elements in a computer** Trigger. Adder. Multi-bit adder. Construction of circuits on logical elements using a given logical expression. Writing a logical expression using a logical circuit. *Microcircuits and their production technology.*

5. Mathematical Modeling, Computer Modeling

This is a key section of the advanced course in informatics, which allows you to apply the acquired knowledge on all theoretical topics of informatics and connect mathematical modeling with approaches to computer modeling of information systems.

Control Signal. Feedback. Receiving signals from digital sensors (touch, distance, light, sound, etc.). Examples of using the feedback principle in control systems of technical devices, including robotics. Examples of robotic systems (traffic control system in a transport system, welding line in a car factory, automated control of home heating, autonomous vehicle control system, etc.). **Model** Problems solved with the help of modeling. Classifications of models. Material (natural) and information models. Continuous and discrete models. Simulation models. Game models. Evaluation of the adequacy of the model to the modeled object and the purposes of modeling. **The concept of a mathematical model** Problems solved using mathematical (computer) modeling. The difference between a mathematical model and a natural model and a verbal (literary) description of an object. Stages of computer modeling: problem statement, construction of a mathematical model, software implementation, testing, conducting a computer experiment, analysis of its results, refinement of the model.	**Systems** System components and their interaction. System effect. Management as an information process. Feedback. **Models and modeling** The purpose of modeling. Adequacy of the model to the modeled object or process, the purposes of modeling. Formalization of applied problems. Presentation of modeling results in a form convenient for human perception. Graphical presentation of data (diagrams, tables, graphs). **Stages of computer-mathematical modeling** Problem statement, model development, model testing, computer experiment, analysis of modeling results. Discretization in mathematical modeling of continuous processes. Modeling of motion. Modeling of biological systems. Mathematical models in economics. Computational experiments with models. Computer modeling of control systems. Processing of experimental results. Least squares method. Estimation of numerical parameters of modeled objects and processes. Restoration of dependencies based on experimental results. Probabilistic models. Monte Carlo methods. Simulation modeling. Queueing systems.

6. Algorithms and Programming

The section includes topics of Olympiad informatics. At the choice of schoolchildren, it is envisaged to study two languages from those recorded in the standard for in-depth study (Python, Java, C++, C#).

We will consider in more detail the contents of the section in comparison with K7-9 and K10-11 by years of study.

Algorithms and programming / K7

The concept of an algorithm. Algorithm executors. An algorithm as a plan for managing an executor.

Algorithm properties. Methods of recording an algorithm (verbal, in the form of a flowchart, program).

Algorithmic constructions. The "following" construction. Linear algorithm. Limitations of linear algorithms: the impossibility of foreseeing the dependence of the sequence of actions performed on the initial data.

The "branching" construction: complete and incomplete forms. Fulfillment and non-fulfillment of a condition (truth and falsity of a statement). Simple and compound conditions.

The "repetition" construction: cycles with a given number of repetitions, with a condition of fulfillment, with a cycle variable.

Auxiliary algorithms. Using parameters to change the results of auxiliary algorithms.

Analysis of algorithms for performers.

Execution of algorithms manually and on a computer. Syntax and logical errors. Failures.

The coordinate system in computer graphics. Changing the color of a pixel.

Graphic primitives: segment, rectangle, circle. Contour properties (color, line thickness) and fill. Constructing images from graphic primitives.

Using cycles to construct images. Hatching a closed area of simple shape (rectangle, triangle with base parallel to coordinate axis).

Principles of animation. Using animation to simulate object movement. Controlling animation using the keyboard.

Algorithms and programming / K8

Programming language (Python, C++, Java, C#). Programming system: program text editor, translator, debugger.

Variable: type, name, value. Integer, real and symbolic variables.

Assignment operator. Arithmetic expressions and the order of their calculation. Operations with integers: integer division, remainder from division. Checking the divisibility of one integer by another.

Operations with real numbers. Built-in functions.

Random (pseudo-random) numbers.

Algorithms and programming / K10

Determining the possible results of the simplest executor control algorithms and computational algorithms. Determining the initial data with which the algorithm can give the required result.

Stages of solving problems on a computer. Instrumental tools: translator, debugger, profiler. Compilation and interpretation of programs. Virtual machines.

Integrated development environment. Methods of debugging programs. Using trace tables. Debugging output. Step-by-step execution of a program. Breakpoints. Viewing variable values.

Programming language (Python, Java, C++, C#). Data types: integer, real, symbolic, logical. Branching. Complex conditions. Cycles with a condition. Cycles by variable. Interchangeability of different types of cycles. Cycle invariant. Compiling a cycle using a predetermined cycle invariant.

Documenting programs. Using comments. Preparing a program description and user instructions.

Algorithms for processing natural numbers written in positional number systems: breaking a number into individual digits; finding the sum and product of digits; finding the maximum (minimum) digit.

Finding all prime numbers in a given range. Representing a number as a set of prime factors. Fast exponentiation algorithm.

Processing data stored in files. Text and binary files. File variables (file pointers). Reading from a file. Writing to a file.

Splitting a task into subtasks. Subroutines (procedures and functions). Recursion. Recursive objects (fractals). Recursive procedures and functions. Using a stack to organize recursive calls.

Using the standard library of the programming language. Connecting third-party subroutine libraries. Modular principle of program construction.

Numerical methods. Exact and approximate solutions to problems. Numerical methods for solving equations: enumeration method, bisection method. Approximate calculation of curve lengths. Calculating the areas of figures

Branching. Compound conditions (writing logical expressions in the programming language being studied). Finding the minimum and maximum of two, three, and four numbers. Solving a quadratic equation with real roots. Logical variables. Dialog debugging of programs: step-by-step execution, viewing values, debug output, choosing a breakpoint. Cycle with a condition. Euclidean algorithm for finding the greatest common divisor of two natural numbers. Splitting a natural number in a positional system with a base less than or equal to 10 into separate digits. Decomposition of a natural number into prime factors. Cycle with variable. Algorithm for checking whether a natural number is prime. Analysis of algorithms. Determining possible results of an algorithm for a given set of input data; determining possible input data leading to a given result. Processing data flow: calculating the number, sum, arithmetic mean, minimum and maximum values of sequence elements that satisfy a given condition. Processing symbolic data. Symbolic (string) variables. Character-by-character string processing. Counting the frequency of a symbol in a string. Built-in functions for string processing. Tabular quantities (arrays). One-dimensional arrays. Compiling and debugging programs implementing typical algorithms for processing one-dimensional numerical arrays in one of the programming languages (Python, C++, Java, C#): filling a numerical array with random numbers, according to a formula or by entering numbers; finding the sum of the array elements; linear search for a given value in the array; counting the array elements that satisfy a given condition; finding the minimum (maximum) element of the array. Concept of algorithm complexity.	using numerical methods (rectangle method, trapezoid method). Finding the maximum (minimum) of a function of one variable using the bisection method. Processing symbolic data. Built-in functions of the programming language for processing symbolic strings. Algorithms for processing symbolic strings: counting the number of occurrences of a symbol in a string; splitting a string into words by whitespace characters; searching for a substring within a given string; replacing a found substring with another string. Generating all words in a certain alphabet that satisfy specified restrictions. Converting a number to a symbolic string and back. Arrays and sequences of numbers. Calculation of generalized characteristics of array elements or a numerical sequence (sum, product, arithmetic mean, minimum and maximum elements; number of elements satisfying a given condition). Linear search for a given value in an array. Sorting a one-dimensional array. Simple sorting methods (bubble sort, selection sort, insertion sort). Merge sort. Quick sorting of an array (QuickSort algorithm). Binary search in a sorted array. Two-dimensional arrays (matrices). Algorithms for processing two-dimensional arrays: filling a two-dimensional numerical array according to given rules; searching for an element in a two-dimensional array; calculating the maximum (minimum) and sum of elements of a two-dimensional array; rearranging rows and columns of a two-dimensional array. *Development of programs for solving simple data analysis tasks (data cleaning, classification, deviation analysis).*
Algorithms and programming / K9 Splitting a task into subtasks. Auxiliary algorithms (subroutines, procedures, functions). Parameters as a means of changing the results of a subroutine. Function result. Logical functions.	**Algorithms and programming /K 11** **Formalization of the concept of an algorithm**. Turing machine as a universal model of computation. Church-Turing thesis. Post machine. Normal Markov algorithms. Algorithmically unsolvable problems. Halt

Recursion. Recursive subroutines (procedures, functions). Recursion termination condition (base cases). Using recursion to enumerate options. Sorting arrays. Built-in sorting capabilities of the selected programming language. Sorting by several criteria (levels). Binary search in an ordered array. Two-dimensional arrays (matrices). Basic algorithms for processing two-dimensional arrays (matrices): filling a two-dimensional array with random numbers and using formulas; calculating the sum of elements, the minimum and maximum of a row, column, range; searching for a given value. Dynamic programming. Problems solved using dynamic programming: calculating functions specified by a recurrence formula; counting the number of options, choosing the optimal solution.	problem. Impossibility of automatic debugging programs. Estimation of the complexity of computations. Operating time and amount of memory used, their dependence on the size of the initial data. Estimation of the asymptotic complexity of algorithms. Algorithms of polynomial complexity. Enumeration algorithms. Examples of various algorithms for solving one problem that have different complexity. Search for prime numbers in a given range using the "sieve of Eratosthenes" algorithm. Multi-digit integers, problems of long arithmetic. Dictionaries (associative arrays, mappings). Hash tables. Building an alpha-frequency dictionary for a given text. Natural language text analysis. Extracting sequences by pattern. Regular expressions. Frequency analysis. Stacks.Queues. Graph algorithms. Building a minimum spanning tree of a weighted connected undirected graph. Depth-first graph traversal. Breadth-first graph traversal. The number of different paths between vertices of a directed acyclic graph. Dijkstra's algorithm. Floyd-Warshall algorithm. Trees. Implementation of a tree using reference structures. Binary trees. Construction of a tree for a given arithmetic expression. Recursive algorithms for tree traversal. Using a stack and a queue to traverse a tree. Dynamic programming is a method for solving problems with saving intermediate results. Problems solved using dynamic programming: calculating recursive functions, counting the number of variants, optimization problems. The concept of object-oriented programming. Overview of programming languages. The concept of programming paradigms. *Learning a second programming language*

7. Digital Literacy

It is important that for both basic and advanced study of the subject, the standard has a special section that is mandatory for all schoolchildren to study at both the basic and advanced levels, this is Digital Literacy. The section includes topics that cover computer literacy (computer), communication literacy (network), information literacy (software) and information security.

Digital Literacy K7-9	Digital Literacy K10-11
Safety precautions and rules for working on a computer A computer is a universal computing device that operates according to a program. Types of computers: personal computers, embedded computers, supercomputers. Mobile devices. The main components of a computer and their purpose. Processor. RAM and long-term memory. Input and output devices. Touch input, mobile device sensors, biometric authentication tools. The history of the development of computers and software. Generations of computers. Modern trends in the development of computers. Supercomputers. Parallel computing. Personal computer. Processor and its characteristics (clock frequency, bit depth). RAM. Long-term memory. Input and output devices. The volume of stored data (computer RAM, hard drive and solid-state drive, smartphone ROM) and access speed for different types of media.	**Safety and hygiene requirements when working with computers and other components of the digital environment** Principles of operation of computers and computer systems. Von Neumann architecture. Harvard architecture. Automatic execution of a program by a processor. RAM, read-only and long-term memory. Data exchange using buses. External device controllers. Direct memory access. Main trends in the development of computer technology. Parallel computing. Multiprocessor systems. Supercomputers. Distributed computing systems and big data processing. Mobile digital devices and their role in communications. Embedded computers. Microcontrollers. Robotic production.
Software Computer software. Application software. System software. Programming systems. Files and folders (directories). File types. File properties. Typical file sizes of different types (text page, e-book, photo, song recording, video clip, full-length film). Principles of file system construction. Full file name (folder, directory). Path to file (folder, directory). File manager. Working with files and folders (directories): creating, copying, moving, renaming and deleting files and folders (directories). Searching for files. Data archiving. Using archiving programs. Computer viruses and other malware. Programs for protection against viruses.	**Software** Software for computers and computer systems. Types of software and their purpose. Features of mobile device software. Parallel programming. System software. Operating systems. Utilities. Device drivers. Installation and uninstallation of software. File systems. Principles of placement and naming of files in long-term memory. Templates for describing file groups. Organization of personal information archive. Backup. Password protection of the archive. Antivirus programs.

Network Connecting computers into a network. Internet. Web page, website. Structure of web resource addresses. Browser. Search engines. Searching for information by keywords and images. Reliability of information obtained from the Internet. Modern Internet communications services Global Internet. IP addresses of nodes. Network data storage. Methods of individual and collective placement of new information on the Internet. Big data (Internet data, in particular social network data).	**Network** Principles of construction and hardware components of computer networks. Network protocols. Internet. Addressing on the Internet. TCP/IP stack protocols. Domain Name System. Dividing an IP network into subnets using subnet masks. Network administration. Obtaining data on the computer's network settings. Checking for a connection with a network node. Determining the route of packet movement. Principles of construction and hardware components of computer networks. Network protocols. Internet. Addressing on the Internet. TCP/IP stack protocols. Domain Name System. Dividing an IP network into subnets using subnet masks. Network administration. Obtaining data on the computer's network settings. Checking for a connection with a network node. Determining the route of packet movement.
Network. Activities on the Internet Internet services: communication services (mail service, video conferencing, etc.); reference services (maps, schedules, etc.), search services, software update services. Government services. Cloud data storage. Collaborative document development tools (online offices). Software as a web service: online text and graphic editors, software development environments.	**Network. Types of activities on the Internet** Internet services. Geoinformation systems. Real-time geolocation services (location of mobile phones, determination of highway congestion, etc.); Internet trade; booking tickets and hotels, etc. State electronic services and services. Social networks – organization of collective interaction and data exchange. Network etiquette: rules of conduct in cyberspace. The problem of authenticity of received information. Open educational resources.
Network. Web page development HTML language. Web page structure. Page title and body. Logical markup: headings, paragraphs. Development of pages containing images, lists and hyperlinks.	**Network. Internet applications** The concept of the server and client parts of a site. Client-server technology, its advantages and disadvantages. Basics of HTML and Cascading Style Sheets (CSS). JavaScript scripts. Forms on a web page. Hosting websites. Hosting service. Uploading files to a site.
Information security Legal protection of programs and data. Free and shareware programs. Free software. Network etiquette, basic norms of information ethics and law when working on the Internet. Strategies for safe behavior on the Internet.	**Information security** Russian Federation legislation in the field of software. Licensing of software and digital resources. Proprietary and free software. Commercial and non-commercial use of software and digital resources. Responsibility

The concept of information security. Information security threats when working on the global network and methods of counteracting them. Rules for secure authentication. Protecting personal information on the Internet. Safe strategies for behavior on the Internet. Prevention of involvement in destructive and criminal forms of network activity (cyberbullying, phishing, etc.).	established by Russian legislation for the illegal use of software and digital resources. Technogenic and economic threats are associated with the use of ICT. General problems of information protection and information security. Information security tools in computers, computer networks and automated information systems. Legal support for information security. Electronic digital signature, certified sites and documents. Preventing unauthorized access to personal confidential information stored on a personal computer, mobile devices. *Malicious software and ways to combat it.*

8. Information Technology

This section includes traditional topics about tools for computer processing of various types of data, but it is supplemented with topics on Artificial Intelligence, Data Analysis (Big DATA).

The section is aimed at testing schoolchildren of various competencies based on modern technologies for processing graphics, text, multimedia, spreadsheets and databases – the topics of this section. It is important that artificial intelligence methods have already been implemented in all data processing systems (search services, editors), which must be practically mastered when studying this section. These are voice search, translators, proofreaders, interlinear translators, handwritten-to-printed text converters, automatic settings for templates when working with graphics, texts, in spreadsheets, presentations, video editors and databases.

К 7-9	**К 10-11**
Information technology Text documents and their structural elements (page, paragraph, line, word, symbol). A word processor is a tool for creating, editing and formatting texts. Typing rules. Text editing. Character properties. Font. Font types (chopped, serif, monospaced). Bold and italic. Paragraph properties: borders, paragraph indentation, spacing, alignment. Style formatting. Structuring information using lists and tables. Multilevel lists. Adding tables to text documents. Inserting images into text documents. Wrapping text around images. Including dia-	**Information technology** Word processor. Editing and formatting. Spell and grammar check. Search and autoreplace tools in a word processor. Using styles. Structured text documents. Footnotes, table of contents. Collaborative work with documents. Review tools in word processors. Cloud services. Business correspondence. Abstract. Rules for citing sources and formatting bibliographic references. Formatting a list of references. Standards of bibliographic descriptions. Introduction to computer typesetting of text. Technical means of text input. Specialized means of editing mathematical texts.

grams and formulas in a text document. Page settings, page numbers. Adding headers, footers, and links to a document. Spell checking. Hyphenation. Voice input. Optical character recognition. Computer translation. Using Internet services for text processing.	
Introduction to graphic editors Raster images. Using graphic primitives. Editing operations for graphic objects, including digital photographs: resizing, cropping, rotating, mirroring, working with areas (selecting, copying, filling with color), color, brightness, and contrast correction. Vector graphics. Creating vector images using built-in tools in a word processor or other programs (applications). Adding vector images to documents. Preparing multimedia presentations. Slide. Adding text and images to a slide. Working with multiple slides. Adding audiovisual data to a slide. Animation. Hyperlinks.	**Graphic editor** Input of images using various digital devices (digital cameras and microscopes, video cameras, scanners, etc.). Resolution. Cropping. Perspective correction. Histogram. Levels correction, color correction. Desaturation of color images. Retouching. Working with areas. Filters. Multilayer images. Text layers. Layer mask. Channels. Saving a selection. Preparing illustrations for websites. Animated images. Vector graphics. Primitives. Changing the order of elements. Alignment, distribution. Grouping. Curves. Vector drawing formats. Using contours. Vectorization of raster images. Principles of constructing and editing three-dimensional models. Grid models. Materials. Modeling light sources. Cameras. Additive technologies (3D printers). The concept of virtual reality and augmented reality.
Spreadsheets understanding Data types in spreadsheet cells. Editing and formatting tables. Built-in functions for finding maximum, minimum, sum, and average. Sorting and filtering data in a selected range. Creating charts (histogram, pie chart, scatter chart). Selecting a chart type. Conversion of formulas when copying. Relative, absolute and mixed addressing. Conditional calculations in spreadsheets. Summation and counting of values that meet a given condition. Processing large data sets. Dynamic programming in spreadsheets. Numerical modeling in spreadsheets. Numerical solution of equations using parameter selection. Finding the optimal solution.	**Data analysis** The main tasks of data analysis: forecasting, classification, clustering, deviation analysis. The sequence of solving data analysis problems: collecting primary data, cleaning and assessing the quality of data, selecting and/or building a model, transforming data, visualizing data, interpreting results. Software and Internet services for processing and presenting data. Big data. Machine learning**.** *Intelligent data analysis*. **Data analysis using spreadsheets** Calculating the sum, arithmetic mean, and the largest (smallest) value of a range. Calculating the correlation coefficient of two data series. Plotting bar, line, and pie charts. Plotting function graphs. Selecting a trend line, solving forecasting problems. Numerical solution of equations using parameter selection.

	Optimization as a search for the best solution under given conditions. Objective function, constraints. Local and global minimum of the objective function. Solving optimization problems using spreadsheets.
Tabular models Table as a representation of a relation. Databases. Selecting rows in a table that satisfy a given condition. Developing a single-table database. Compiling database queries using a visual editor.	**Information technology** Tabular (relational) databases. A table is a representation of information about similar objects. Field, record. Table key. Working with a ready-made database. Populating a database. Searching, sorting, and filtering data. Queries for data selection. Queries with parameters. Calculated fields in queries. Multi-table databases. Types of relationships between tables. Foreign key. Database integrity. Queries to multi-table databases. Basic principles of database normalization. SQL data management language. Creating simple queries in SQL to select data from one table. Non-relational databases. Expert systems.
Information technology The role of information technology in the development of the economy of the world, country, region. Open educational resources. Professions related to computer science and information technology: web designer, programmer, mobile application developer, tester, software architect, data analysis specialist, system administrator. Familiarization with promising areas of information technology development (using artificial intelligence and machine learning as an example). Smart city systems (computer vision and big data analysis).	**Information technology. Artificial intelligence tools** Machine translation and speech recognition services. Cognitive services. Image identification and search, face recognition. Self-learning systems. Artificial intelligence in computer games. Using artificial intelligence methods in training systems. Using artificial intelligence methods in robotics. Internet of things. Prospects for the development of computer intelligent systems. Neural networks.

9. Methodological Aspects of Improving Standards for Primary and High School in Countries Involved in the Olympiad Movement in Informatics

Improving the educational standard in informatics in schools requires the emergence of new methods of supporting gifted schoolchildren. Let us consider several important methodological aspects that allow us to do this.

The main task of supporting gifted schoolchildren is the availability of such training for all children in the country. If the educational standard guarantees the availability of

training in new informatics content, then the guarantor of the implementation of such training are the schools of the country. At the same time, schools face various problems: training qualified teachers for new computer science topics, the availability of courses and teaching materials on these topics, including online, and creative events for schoolchildren to demonstrate their giftedness in these new topics of study.

It is important to note that the foundation of all new topics in informatics remains the mathematical foundations of informatics, algorithmization and programming. These sections of the course can be called the fundamental core of the subject of informatics. This fundamental core has been deepened in the school standard in accordance with the IOI syllabus, which is important for the development of programs for teaching gifted children considering these topics (complex algorithms, graphs, strings, expansion of programming languages with languages such as Python, C++ and Java). Gifted children can choose training in these sections, which are focused on the national olympiad in informatics and IOI, starting from the 7th grade of school. At the same time, to solve the problems described, it is important to introduce accessible forms of work with gifted schoolchildren. Traditional forms in many countries are special IT lyceums or centers for preparation for the Olympiad at the country level.

To cover younger schoolchildren, as well as to teach them new specialized modules of informatics, flexible forms of organizing the education of schoolchildren are required. The creation of such forms can be implemented based on the "Hybrid Learning" model, which optimizes the selection of teachers, including the involvement of university students as course curators, combines online, mobile and face-to-face forms and can cover children throughout the country. Such models can be quickly implemented in schools. These include: a school IT-Lab, IT Class at school or IT Club for a group of schools, IT Lyceum, IT Bus as a mobile IT Lab, as well as a partner IT Campus Online for the country, for example, based on a university or IT business park.

All these models create an accessible information educational environment for supporting gifted schoolchildren in the IT sphere. Their comprehensive implementation will allow creating a network of IT platforms for children with a choice of various specific forms for territories in the country. These models of organizing work with children will allow the introduction of new additional IT modules of their standard in computer science, which are flexibly configured according to the choice of students, for example:

- Programming languages (second and third programming languages of choice).
- Programming AI applications (bots, computer vision, voice assistants, translators, biometric recognizers, etc.).
- Algorithms and software tools for big data analysis.
- Control of unmanned robots and machines.
- Software procedures for information security, cryptography.
- Programming of desktop robots and digitally controlled machines.
- Circuitry and chip programming.
- Geoinformatics, cartography systems, geonavigation.
- Additive 3D printing technologies.
- Media education and digital arts.
- Advanced computer technologies and quantum computing.

For mass support of children passionate about computer science, the main role is played by such models as IT laboratories and IT clubs. For example, there are countries where school IT laboratories begin working with children from elementary school to high school in three age groups – elementary, basic, advanced. Small, specialized modules of 12–36 hours of training per year are used in training. In this case, the student can choose up to three training modules per year. This model implements the methodology – «test of the IT profession». It has become widespread in the last 5 years as a form of additional education based on individual schools in all territories. For example, in Russia, this model has become widespread in the form of such digital educational laboratories or clubs as «IT Cube», «Quantorium», and for rural schools – «Growth Point». It is important to note that the implementation of such a model requires modern, but small in number laboratory equipment, including computers, the Internet and sets of applied digital equipment. This allows schoolchildren to choose innovative areas of further professional training.

The IT class model has been widespread since 2010 as a systemic form of support for children interested in informatics. It implements the normative model – specialized training from grades 5 or 7 in schools and operates according to the standard of an in-depth level of informatics study. At the same time, children from IT classes can also choose modules – electives for additional training of their choice. IT classes require a professional – informatics teacher and curators – students at IT universities for specialized IT practices. By choosing an IT class, children receive an early choice of professions in the IT field and further education at an IT university.

The IT lyceum model is a classic model that has become traditional for working with gifted children since the end of the 20th century. Selection to lyceums usually begins in the 7th grade. Such lyceums have become the main training grounds for the national Olympiad in informatics and robotics. This model requires a highly qualified teaching staff with appropriate ICT competencies, equipment for several IT classrooms, and modern and diverse digital equipment in sufficient quantity. Leading universities and IT commercial companies can provide assistance to IT lyceums in this regard, where high school students can do an internship. Often, such IT lyceums are national or open at large universities in the country. In most cases, students leave home to study at an IT lyceum and live on campus. It is obvious that IT lyceums do not solve the problem of mass work with gifted schoolchildren, but prepare the IT elite, which is also important for the country.

During the pandemic, a new mass form of support for gifted children has become widespread – the «Digital School» model based on IT Campus or IT Parks at universities. It has significantly expanded the possibilities of educational courses for applicants at universities, since in addition to preparatory courses for admission to the university, it has become an online platform for popularizing modern IT, preparing and holding open, including international Olympiads in various new tracks of Olympiads in informatics.

A special feature of the «Digital School» model is the implementation of a hybrid form of education. Some events are held online, open to all comers or selected by level of training. Online rounds of Olympiads are also offered, based on the results of which

finals are held in a face-to-face format, when participants are still getting to know the university, being on site at the final. Such digital university schools cover children of all ages, any territories, have a flexible structure of many courses, and also have no staff shortage, as they invite students to work in the relevant departments.

Thus, all the proposed models, close to schoolchildren everywhere in the country, become real methodological resources for the development of children who are passionate about informatics, for mastering new educational IT modules in the informatics standard, technologically and do not lag behind innovations.

10. Discussion of the Prospects for the Development of New Forms of Teaching Gifted Schoolchildren in Informatics

Along with traditional forms of work with gifted schoolchildren in informatics, it is very important to consider IT innovations and help children choose promising development tracks. And here, the expansion of IT training tracks and the coordination of maximum coverage of all children involved in the above-mentioned models of schoolchildren's preparation are of particular importance.

The experience of the last 10 years has proved that the following organizational activities are required:

- Active development of various tracks of Olympiads in informatics, that organically complement the traditional national olympiad in informatics in the country, and which are also based on algorithmization and programming, but are used in an applied IT environment.
- Implementation of small organizational forms of IT training for children in the school infrastructure with the involvement of students from IT universities and IT business partners, and involving children in IT practices.
- Equipping IT classes and IT laboratories based on schools, additional education clubs.
- Development of mobile IT bus laboratories with access to samples of the most modern IT devices (for example, 3D helmets, drones, 3D pen, 3D scanner, devices with AI) and the Internet.
- Development of children's Digital Schools in universities.

If we talk about new tracks of the Olympiads in informatics, these could be the Olympiads in robotics, information security, artificial intelligence, financial literacy (business informatics), IT hackathons (integration of informatics tracks and other school subjects) and the Games of the Future (combining informatics and sports tracks). All of them, one way or another, rely on the traditional sections of the informatics Olympiad, but are supplemented by applied IT areas, showing the penetration of digital into all school subjects, which is now of great interest to children.

The experience of the International School of Informatics for Juniors (ISIJ, 2025), which was organized to support junior participants in the field of informatics, made it possible to identify the potential of IT parks in countries to implement the coordina-

tion function for national Olympiads and their development. In particular, to conduct an open selection stage in an online format. This can be done in an organized manner with children within walking distance for them at the sites of IT laboratories, IT classes in areas throughout the country. Such experience was gained in 2023 in Uzbekistan, where the national IT park has representative offices in each district of the country to work with children. This affects the popularization of informatics and supports new tracks of Olympiads in the IT field directly in partnership with schools, providing the personnel potential of the IT park to work with children. ISIJ allowed countries to begin the formation of a children's Info Park in the country as an integrator of IT education for children, coordinating the work of all models of IT education, including holding various new national IT Olympiads based on informatics school.

It can be argued that the formation of departments for working with children, such as Info Parks (informatics parks), at the IT parks of the country is a very promising area of cooperation between the school and the IT business community. For example, a successfully functioning model of an international digital school is the IT school of Innopolis University (Innopolis, 2025) `https://progmatica.innopolis.university/`, which is open to all children from 12 years of age from all over the world and is an online and face-to-face platform for preparing and holding the Innopolis International Open Olympiad in five tracks: mathematics (for informatics), informatics (programming), robotics, artificial intelligence, information security (OI Olympiad, 2025). (`https://dovuz.innopolis.university/pre-olympiads/innopolis-open/en`) Here children can get their first experience and start in the IT sphere, choosing different ones from the five above-mentioned tracks.

11. Conclusion

Based on the information provided in the article, it can be argued that the introduction of an advanced course in Informatics in the country's schools allows all motivated children to gain free access to Olympiad training in Informatics. But it is also important that in this case, all relevant specialized schools (for example, IT lyceums) receive a single program for Olympiad in Informatics for children, which provides teachers with the opportunity to work in a single standard regardless of the child's place of residence, as well as to receive a specialized textbook for the advanced course.

It is equally important that schoolchildren can master the fundamental principles of theoretical computer science and gain practical experience in applying knowledge in various modern programming environments, as well as try themselves in the profession as part of their first experience working with information systems, artificial intelligence systems and data analysis already at school. At the same time, all schoolchildren must also master digital literacy and information security culture, which is also useful for Olympiad participants.

The development of the federal standard based on the deepening of its scientific content, expansion by innovative technologies is dictated by the needs of any country for personnel in the field of high technologies and the need to modernize science in the

context of the digital transformation of the world. The special value of Olympiad in informatics is that it identifies and prepares human resources among young people for the digital economy.

The authors hope that the content of this article will be useful both to specialists involved in the development of the content of the national educational standard and to teachers working with talented schoolchildren in the field of informatics. Russia's experience shows that the use of the above-described advanced course in informatics starting from the 7th grade will allow talented schoolchildren to achieve high results not only in national, but also in international Olympiads in informatics, including the IOI.

Reference

Kiryukhin, V. (2010). Mutual Influence of the National Educational Standard and Olympiad in Informatics Contents. *Olympiads in Informatics*, 2010. Vol. 4, 15–29. https://ioinformatics.org/journal/INFOL061.pdf

Educational Standard (2022). *Register of sample educational programs, Ministry of Education of Russia*. https://fgosreestr.ru/educational standard

Curriculum of the Base School education (2023). *Register of sample educational programs, Ministry of education of Russia.* https://fgosreestr.ru/poop/federalnaia-obrazovatelnaia-programma-osnovnogo-obshchego-obrazovaniia-utverzhdena-prikazom-minprosveshcheniia-rossii-ot-18-05-2023-pod-370

Curriculum of the Hier school education (2023). *Register of sample educational programs, Ministry of Education of Russia.* https://fgosreestr.ru/poop/federalnaia-obrazovatelnaia-programma-srednego-obshchego-obrazovaniia-utverzhdena-prikazom-minprosveshcheniia-rossii-ot-18-05-2023-pod-371

Curriculum Advanced Informatics K7-9 (2023). *Register of sample educational programs Ministry of Education of Russia.* https://fgosreestr.ru/oop?edl=3&sub=19

Curriculum Advanced Informatics K10-11 (2023). *Register of sample educational programs, Ministry of Education of Russia.* https://fgosreestr.ru/oop?sub=19&edl=2

FLSB (2024). *Federal List of schoolbooks, Ministry of Education of Russia.* https://fpu.edu.ru/

RESH (2024). *Russian Electronic School, Ministry of education of Russia.* https://resh.edu.ru/

Sanitary Standard (2024). *Register of sample educational programs, Ministry of education of Russia.* https://fgosreestr.ru/sanitary_standard

ISIJ (2025). *International School of Informatics for Juniors*. https://isi-junior.ru/uch/infopark/

Innopolis (2025). *Innopolis University*. https://progmatica.innopolis.university/

OI Olympiad (2025). *Innapolis Open International Olympiad.* https://dovuz.innopolis.university/pre-olympiads/innopolis-open/en

M.S. Tsvetkova is professor of the Russian Academy of Natural Sciences, PhD in pedagogic science, prize-winner of competition "The Teacher of Year of Moscow" (1998). From 2002 to 2018 she is a member of the Central methodical commission of the Russian Olympiad in informatics and the pedagogic coach of the Russian team on the IOI. She is the author of many papers and books in Russia on the informatization of education and methods of development of talented students. She is the author of official textbooks and copybooks in Russia for primary school in Informatics. She is author and director of the International school in Informatic ISIJ (since 2017). She is the Russian team leader (2013–2017). She was awarded the President of Russia Gratitude for the success organizing the training of IOI medalists (2016). She was the Expert of Committee on Education and Science State Duma of the Russian Federation (2017–2021), and she has the Committee on Education and Science State Duma Gratitude (2021).

V.M. Kiryukhin is professor of the Russian Academy of Natural Sciences, PhD. He is the author of many papers and books in Russia on development of Olympiad movements in informatics and preparations for the Olympiads in informatics. He is the exclusive representative who took part at all IOI from 1989 to 2017 as a member of the IOI International Committee (1989–1992, 1999–2002, 2013–2017) and as the Russian team leader (1989, 1993–1998, 2003–2012). He received the IOI Distinguished Service Award at IOI 2003, the IOI Distinguished Service Award at IOI 2008 as one of the founders of the IOI making his long term distinguished service to the IOI from 1989 to 2008 and the medal "20 Years since the First International Olympiad in Informatics" at the IOI 2009. He was the chairman of the IOI 2016 in Russia and has the award medal of the President of Russia (2016) for organizing the Olympiad in Informatics in Russia and training IOI medalists since 1989. He is now the President of the International Organizing Committee of the ISIJ.