

Stated by:

Deputy Chairman of State University department of Russia

Shadrikov V.D.

12 October 1994

State Standard of High Professional Education

State requirements

Describing minimum scope and level of schooling for the graduated students, specialty 220400 – Computers and automated systems software

Active from 1 September 1994

Moscow, 1994.

1. General characteristics of specialty 220400 - Computers and automated systems software
 - 1.1 This specialty is based on the Order of State High Education Committee of Russian Federation N 180 from 05.03.94
 - 1.2 Qualification of the graduated is Engineer, education program term is 5 years
 - 1.3 Characterization of professional field of the graduated
 - 1.3.1 Position of the specialty in science and industry. Computers and automated systems software is a field in science and industry, which includes the systems and methods of human activities, directed on creation and application of software for computers, automated systems. It also includes development of new fields of application for computers and automated systems.
 - 1.3.2 Objects of professional activity. Objects of professional activity for an engineer of specialty 220400 – Computers and automated systems software are programs, program systems and complexes, their mathematic and algorithmic models, methods of their planning and realization, production and maintenance methods.
 - 1.3.3 Types of professional activities. Engineer of specialty 220400 – Computers and automated systems software, according to his fundamental and special education can perform the following types of professional activities:
 - project management

- scientific exploration
- industry management
- maintenance

Engineer of specialty 220400 can work in scientific, project, design, technological, commercial offices as well as in banks and factories.

Engineer of specialty 220400 must be able to solve, using his knowledge and experience, professionally and with full responsibility, the following tasks:

- Make choice of an appropriate mathematical model of a process in order to make effective software, which is going to use or investigate this process
- Make technical requirements and specifications for software according to user wishes and technical possibilities
- Make system application or auxiliary software for computers and automated systems according the modern methods, means and technologies
- Make choice of appropriate hardware and software for effective execution of software projects
- Make evaluations of software quality during the software development cycle: planning, execution and maintenance, as well as making modernizations in this software in order to improve its stability and quality
- Make documentation according to standards for different people, using the software
- Develop new methods, means or technologies for software in scientific, design, management, technological, economical, social and humanitarian fields.

2. Education level requirements for the graduated students of specialty 220400 – Computers and automated systems software

2.1 General education requirements for an Engineer:

Engineer must:

- Be aware of the modern studies in humanitarian and socio-economical studies; be able to scientifically analyze social problems and processes; be able to use methods of this sciences in different types of professional and social activities
- Know ethical and legal norms about relationships between a person and another person, society and nature; take this norms into account in development of ecological and social projects.
- Have solid understanding of processes in nature, know modern methods of understanding nature and use those methods for the projects in the fields of natural science.
- Be able to continue education and work in a foreign country (for full realization in 10 years)
- Know a scientific approach to a healthy living and know different fitness methods
- Know the culture of thought, be able to logically express his ideas vocally or in written form
- Be able to scientifically organize his own work, use information technologies of his field
- Know the basics of management from technical, financial and human prospective.
- In case of new discoveries in science or industry to be able to review his own experience, analyze his own possibilities and adopt the new knowledge using modern information technologies
- Understand meaning and social importance of his own future profession, problems of the fields of science, defining his profession and see their place in the total body of science.
- Be able to plan in his own professional field using system approach, build and use models for analyze and prognosis of different processes, perform quantity and quality analyze.
- Be able to express tasks and make a goal within his professional field, use scientific methods for that task

- Be able to work in a team, know management methods, be able to organize work of employees, be able to make management decisions, know the basics of education science.
- Be ready methodically and psychologically to change his professional field or work with people with other professions

2.2 Knowledge and experience requirements in different branches of learning

2.2.1 Requirements in humanitarian and socio-economical branches of learning

Knowledge and experience requirements follow the federal Requirement for minimum level of education "General humanitarian and socio-economical branches of learning", defined by Russian Federation State High Education State Committee 18 August 1993

2.2.2 Requirements in mathematical and natural science branches of learning

Engineer must:

In mathematics:

Have knowledge:

- About mathematics as a special way to understand the world, about mathematics body of knowledge
- About mathematic models and formal methods of description and analysis of processes.
- About storage, processing and transfer of information
- About modern information models, place of mathematics in general science body of knowledge

Know and be able to use:

- Basic methods and notions of mathematical analysis, analytical geometry, general and linear algebra, theory of complex variables functions, operational calculus, probability theory and mathematical statistics.
- Calculus of integration and differentiation approximations, linear algebra, solving of algebraic and differential problems, spectral representation and transformation
- Basic notions of set theory
- Boolean algebra, methods of transformation of Boolean functions and usage of Boolean algebra in computer elements
- Basic theory of mathematical logic and theory of logical conclusions
- Basic notions of computers
- Principles of hardware and software in information systems
- Basic methods of information theory, information flows in natural systems, mathematical models of information processes and systems in industry

Have experience of:

- Using mathematical language to express quantity and quality relationships between objects
- Investigation of mathematical models and limits of their application
- Analytical and computational solution of plain, differential and difference equations
- Usage of computational methods of linear algebra and function approximation
- Usage of theorems and methods of mathematical logic and predicative calculus

In physics and ecology:

Have knowledge:

- About Universe and general and its evolution
- About general unity of natural science, incompleteness of human knowledge and its possibilities to evolve.
- About discretion and continuity in nature
- About relationships between chaos and order in nature, ordered structure of objects, conversion of ordered structure into chaotic and vice versa.
- About dynamic laws of nature
- About probability as a real characteristics of nature systems and statistical laws in nature
- About measurements and their specifics in different nature sciences
- About fundamental constants of natural sciences

- About symmetry and conservation laws
- About relationship between empiric and theoretic in knowledge
- About nature states and their changes in time
- About individual and collective objects in nature
- About time in natural sciences
- About latest discoveries in natural science and their prospective for creation of new technical systems
- About specifics of biologic form of nature, principles of recreation and evolution of life forms
- About biosphere and its evolution
- About integrity and homeostasis of life systems
- About relationships between a life being and nature, life societies and ecosystems
- About ecological principles of nature preservation and rational usage of nature resources, new ecology-friendly technologies
- About physical and biological models
- About results of your own work from prospective of biosphere and bio-social nature of human

Know and be able to use:

- Basic notions, laws and models of mechanics, electrical and magnetic forces, oscillations and waves, quantum physics, statistical physics, thermodynamics and ecology
- Methods of theoretic and experimental research in physics and ecology
- Calculative ranges of different values in different fields of natural science

2.2.3 General professional branches of knowledge

Engineer must:

Have knowledge:

- About modern computer programming and algorithmic languages, their usage, specifics and evolution
- About trends in structures and algorithms development and processing
- About problems and trends in information processes and structures theory, new methods of their formal description and verification
- About basic trends in program semantic definition, specification and verification
- About problems and evolution of system software
- About trends in PC, systems and networks architecture
- About prospects in computer graphics software and hardware, architecture of high-performance graphic stations
- About computer-aided design of information and computation systems
- About basic notions of management theory in different branches of knowledge
- About scientific and management methods of liquidation of results of nature disasters, catastrophes and other non-standard situations

Know and be able to use:

- Modern methods and means of algorithm and program development, methods of structure programming, algorithms in the high-level programming languages, debugging, testing and documenting of computer programs.
- Methods of effective realization and optimal processing of data structures, methods and means of file system organization
- Methods of program semantic definition, specification and verification
- Formal models of basic computational structures and processes, their technical realization, process management and synchronization methods, computer system relationship protocols, methods of structure and process analysis
- Basic models and theoretical methods of programming languages, functional algorithms of translators and compilers
- Methods of building and demonstration of 3-dimentional objects, principles of computer graphic software and hardware, basic methods and algorithms of building and transformation of computer images, graphic communication methods, graphic controllers and processors functionality

- Basic principles of organization and algorithms of operating systems and shells
- Models and organization of databases, basic data manipulation language constructions
- Basic principles of organization and functionality of PC, devices as well as PC networks and complexes; Modern PC types, classes and usage limits.
- Modern methods of system analysis of objects, processes, operation research and decision making process
- Basic model classes and modeling methods, principles of process modeling, methods of formalization, automatization and realization of computer models
- Basic notions of management theory, basic methods of analysis and synthesis of linear or discrete management systems
- Principles of workplace safety and security

Have experience in:

- Usage of mathematical models and methods for analysis, calculation and optimization of deterministic or random processes
- Making choice of technology and methods for planning, development, debugging, testing and documentation of programs, written in high-level procedure-based programming languages
- Usage of computer graphic technology and graphic interface
- Usage of modern system programming tools: operating systems, operating shells and utilities
- Usage of system modeling method for research and planning of system, development of model algorithms on basis of programming languages and applications

2.2.4 Special professional branches of knowledge

Engineer must:

Have knowledge:

- About problems and trends in modern software design, about basic methods and tools of computer-aided design, production, testing and quality assurance of software
- About trends in non-standard computer architecture developments
- About trends in functions and architectures of problem-oriented software systems
- About trends and problems in modern software market
- About modern methods and means of electric and digital schema's calculation; methods of computer-aided design and industry production of such schema's
- About trends in microelectrodes, prospect schematic solutions in digital and analog technology
- About basic trends in software performance and utilities of software performance measurement

Know and be able to use:

- Basic concepts of function-based, logical-based and object-oriented program development, methods means and development tools for those types of program development.
- Methods of planning and development of software, principles of building, structure and usage of utilities for software development
- Methods of management for software developers
- Methods of parallel program design and parallel algorithms
- Methods of Artificial Intelligence systems planning and development
- Methods, models and utilities for computer-aided research data analysis
- Basic notions of software performance measurement, principles of utilities for software characteristic measurement
- Modern digital and analog computer hardware
- Architecture, algorithms and functionality of real-time systems, methods of real-time program development
- Technical basis, structures, methods and algorithms of interactive graphic software and multimedia software

- Software copyright basics
- Architecture, algorithms, planning and realization methods for the digital signal processing systems

Have experience in:

- Building of information technology models and application of these models
- Making choice of modern computer and network architecture
- Making choice of tools for planning, realization, performance measurement and quality assurance of different software

Additional requirements for the special education of engineers could be defined by a particular university according to specialization

2.2.5 Minimum education program contents for engineers with , specialty 220400 – Computers and automated systems software:

Index	Branch of learning name and description	Total hours
GSE.00	General humanitarian and socio-economical branches of learning Knowledge and experience requirements follow the federal Requirement for minimum level of education “General humanitarian and socio-economical branches of learning”, defined by Russian Federation State High Education State Committee 18 August 1993	1800
EN.00	Mathematics and natural science	2000
EN.01	Mathematics	1160
EN.01.01	Higher mathematics Algebra: basic algebraic structures, vector spaces and linear representations; Geometry: analytic geometry, multi-dimensional Euclid geometry, differential geometry of curves and surfaces, topology Analysis: Differential and integral calculations, functions and functional analysis theory, complex variables functions theory, differential equations Probability theory and statistics: Fundamental probability theory, mathematical basis of probability theory, random processes models, statistical models, hypothesis verification, principle of maximum likelihood, methods and procedures of parameter testing, statistical methods and algorithms for experiment data process	760
EN.01.02	Discrete mathematics Sets and their specifications, Venn diagrams, relationships and their features, division and equivalence principle, order relationship, functions and mapping, operations Boolean algebra, discrete structures, graphs, networks, codes; basic graph theory; routes, cycles; links; planar and oriented graphs. Boolean functions and function schema’s, switch functions, functional completeness theorem, examples of functional complete basis’s Integer numbers and polynomials; recurrent equations, codes with error corrections Mathematical logic: number systems, expressions, predicates, general calculations; output definition; calculable functions; calculation models; non-calculable functions; solvability and denumerability; expression logic; normal forms; performability and validity; logic-mathematical language; predicates calculations; theory of logical output; deductive systems; completeness and consistency of predicates calculation; Gödel theorem about incompleteness; resolution method; output search tactics	200
EN.01.03	Computational mathematics Computer mathematical calculations: floating-point and fixed-point number representations; limits and error borders of representation; operations with numbers; arithmetical operations; Theoretical basis of calculation methods: miscalculation; stability and complexity of algorithms (memory and time-required); calculation methods of linear algebra; non-linear equations and systems solution; function interpolations; numerical integration and differentiation; solution of common differential equations; methods of function approximation; Fourier and Walsh transformation, fast Fourier transformation; equal function approximation; overview and analysis of computation methods, used in	200

	linear algebra software	
EN.02	<p>Informatics</p> <p>Information theory basics: information, information technology, information processing process, computer as technical mean for realization of information technologies, hardware and software structures from prospective of computer user, algorithms and data for storing and processing text and number-based information, computer user environments, organization and means of man-computer interface, multi-environments and hyper-environments; Artificial Intelligence systems; Computer networks and information technologies in networks; basics of telecommunication and distributed data processing; economical and legal aspects of information technologies</p>	120
	General natural science branches of learning	510
EN.03	<p>Physics</p> <p>Physical basics of mechanics: notion of state in classical mechanics; move equation; conservation law; basics of relativity mechanics, relativity principle in mechanics, kinematics and dynamics of solid body, fluids and gases; electricity and magnetism; electrostatic and magnetic static in vacuum and matter; Maxwell equation in integral and differential form; quasi-static currents; relativity principle in electrodynamics; oscillations and waves physic: harmonic and inharmonic oscillator; physical meaning of specter analysis; kinematics of wave processes; normal modes; interference and diffraction in waves; basics of Fourier optics; quantum physics: wave-particle dualism; indetermination principle; quantum states; superposition principle; quantum movement equation; operators of physical values; energy specter of atoms and molecules; nature of chemical links; Static physics and thermodynamics: three bases of thermodynamics; thermodynamic state functions; phase equilibrium and phase transformations; basics of non-weighted thermodynamics; classical and quantum statistics; kinetic processes; systems of charged particles; condensate state</p>	440
EN.04	<p>Ecology</p> <p>Biosphere and human; biosphere structure; ecosystems, relationships between an organism and environment; human ecology and health; global ecological problems; ecologic principles of rational usage of nature resources and nature preservation; basics of nature resources economy; eco-safe technology; basics of ecology laws; professional responsibility; international relationships in the fields of nature preservation</p>	70
OPD.00	General professional branches of learning	1640
OPD.01	<p>Software design</p> <p>Main steps of software designs on PC; software quality criteria; dialog programs; user-friendliness; software lifecycle; technical requirements and software specifications; methods of algorithmization; programs on high-level programming languages; standard data types; major structures: iterations; forks; loops; procedures: building and usage; user-defined data types; records; files; dynamic data structures; lists: basic types and realization; recursive algorithms design; methods of program architecture; module programs; basics of validation; architecture and features of high-level programming languages</p>	220
OPD.02	<p>Structures and data processing algorithms in computers</p> <p>non-linear data structures: classification; trees: oriented; sorted and binary; trees in computer memory: queued and linked memory positioning; operations with trees; graphs and their presentation in computers; graph algorithms; searches; total search: rundown with returns; methods branches and borders, dynamic programming; fast search: binary and consecutive search in arrays, hashing; using trees in search binary; random-binary; optimal and balanced search trees; search in graphs; sorting; internal and external sorting; sort algorithms; sort and search algorithms complexity and effectiveness analyze; files: organization and processing, tree representation: b-trees; algorithms complexity theory: NP-complex and difficult tasks</p>	170
OPD.03	<p>Computers organization and functionality</p> <p>Computer architecture: basic notions basic structure of software and hardware;</p>	110

	organization of traditional computer hardware layers; process management; management functions of CPU; general organization of computer software; processing methods management; memory management; functional memory organization; external computer memory; memory management tools; input-output management; I/O management microprocessors; computer assembly level; basic operators; data addresses; loadable modules	
OPD.04	Theory of computational processes and structures Basics of formal language theory; grammatical; automats: model of finite automate; recognizers and transformers; translators: structure methods; functional algorithms; code optimization methods; semantic theory of software; software schemas, methods of formal specification and verification; computational processes models; process integration; protocols and interfaces; asynchronous processes; Petri network: basic principles, algorithms, realizations, usage; principles and methods of technical realization of process and structures methods	180
OPD.05	Computer graphic representation of geometric object on plane; projecting methods: point, line, plane, strait line, surface, their intersections, positions; methods of plane projections; metric tasks; positioning tasks; axonometric projections; computer graphic hardware: graphic displays; object representation and computer generation; computer graphic software: basic structures (graphic objects, primitives and their attributes), graphic tools of high-level languages; graphic editors: basic constructions, object representation algorithms; graphic libraries and their usage; interactive computer graphic as computer-aided design subsystem	100
OPD.06	Operating systems Principles of operating systems (OS), computational process and its realization with OS; basic OS functions; overview of modern Oses and OS shells; standard service programs; hardware-depended OS features; management of computational processes, input-output processes, real-time memory; virtual memory management; hardware-independent OS features; method of user tasks planning; dynamic consecutive and parallel program structures; methods of OS architecture; safety and security of computer systems; interfaces and main standards of system software	100
OPD.07	Databases Databases; basic notions and definitions; physical and logical (conceptual) model of databases; external data model; hierarchical and network database models; user workspace; relational database models; basic notions and definitions; attributes and keys; relationship normalization; relational algebra; basic operations of relational algebra: conjunction; differentiation; Cartesian multiplication, projections, relation conjunction, choice operation; relational databases languages; network and hierarchical database types; CODASIL mode; examples of realization; database requests management	100
OPD.08	Computers and networks architecture methods of organization and types of computers; parallel computing: levels and methods of organizations; realization in multicomputer and multiprocessor environments; operational production line; vector, matrix and associative systems; homogeneous systems and environments; RISC-architectures; programming language environment architectures evolution; basics of computer metric systems; distributed data processing technology; computer networks principles and architecture; protocols; protocols hierarchy and modes: data storage, data transfer, session reset; computer network information transfer; data channels, modems; encoding and error protection; packet structure; routing; basic data transfer hardware; LAN; structure and principles of LAN; links configuration; standards, agreements and recommendations; computer networks software	100
OPD.09	Computer simulation Concept of model; model classification; conceptual models; logical model structure; levels and types of modeling; building of modeling algorithms; formalization and algorithmization of processes; mathematical modeling methods; imitation models; planning of imitation experiments with models; computer statistical modeling;	90

	quality control of modeling results; tools; modeling languages; analyze and interpretation of computer modeling results; modeling of information systems, computers and networks	
OPD.10	Basics of management theory Basic principles of system organization; stability; controllability and objectivity; invariance and sensitivity of management systems; mathematical models of management objects and systems; models representation types; methods of management system analysis and synthesis; digital management systems; computers and microcomputers in management systems; specifics of mathematical description of management systems, analysis and synthesis of computer-controlled management systems; software realization of management algorithms in digital systems	100
OPD.11	Industrial safety Human and environment; basics of work physiology and work comfort; safety and ecology of technical systems; safety in extreme situations; industrial safety management; basics of electricity safety; automatic control systems; psychological factors of information systems	100
OPD.12	Branches of learning, chosen by student	270
SD.00	Special branches of learning	2372
SD.01	Functional programming Programming with functions and procedures; recursive functions and lambda-computation of A.Church; programming in functional labels; function languages; strictly defined languages: basic notions; programming methods; representation and interpretation of functional programs; relation between functional and imperative programs; application of functional programming	100
SD.02	Logical programming Basic knowledge about logical programming language; basic elements of language and programming methods; correlation between goal statements; calculus in logical programming languages; recursive representation of data and programs; cutting and its application; input and output; embedded predicates; program debugging; examples of logical programming for Artificial Intelligence projects	90
SD.03	Object-Oriented programming Basic notions and models: object, class, data, method, access, feature inheritance; object and class systems; object-oriented programming: methods and algorithms; object-oriented languages; classification, architecture, interface methods, technology, application; interface: organization rules; methods and programming tools; object-oriented systems: methods, languages and programming styles	160
SD.04	Software development technology Software products, production process: methods, technology and tools; testing and debugging; documentation; software design; abstract data types; methods of effective data storage and processing; technological cycle of software systems; teamwork on software projects; work organization in groups and tools to support it; automation of software design; basic structures and technology of computer-aided design	170
SD.05	Parallel programming Methods and tools for parallel data processing: parallel computation methods; parallel computers; parallel programming; vector representation of consecutive algorithms, methods of canonical representation of algorithms in signal dependencies and flows graphs; matrix processors; tools for specification of parallel processes; interfaces for relationships between asynchronous parallel processes; synch-primitives; methods and languages of parallel programming: Ada-language, matrix language of data flows, Occam language: basic constructions and programming methods; application of languages for practical tasks; languages comparison; effectiveness of application; multiprogramming systems; parallel data processing in trans-computer systems	120
SD.06	Real-time systems Classification of real-time tasks, main requirements for computational systems of real	100

	time; architecture of real-time systems; programming languages: main constructions, methods and programming styles; architecture of real-time systems software; functional quality control; examples of typical systems	
SD.07	Artificial Intelligence systems Tasks and solutions: search in state environment; reduction and deduction; production systems: components, solution strategy, search methods: method of key states and key operators, methods of means and goal analysis; data representation in intellectual systems: notion-based, rule-based, logic-based, semantic networks, frames, scripts; databases; planning in AI-systems; examples of automated planning and solutions for problems; expert systems: relationship between user and system, decision making; natural language recognition systems; automated translation; visual recognition: systems of computer vision; image recognition; vision systems of robots; learning in intellectual systems	120
SD.08	Digital signal processing systems Signal models; computer signal processing tasks classification; methods and algorithms of signal parameters recognition; signal filtration: Kalman optimal discrete filter, signal filtration divergence; computer spectral analysis: signal representation, time and frequency sampling; Fourier fast transformation algorithms; measurement of spectral density of signal strength; sound and visual signals: computer representation, generation, storage and processing; technical tools for digital signal processing: signal schemas and processors, workstations, special systems, programming methods	90
SD.09	Interactive graphic systems Graphic workstation and hi-end graphic systems architecture; hardware-software implementation; method of graphic input and images visualization; 3D-graphic: representation of 3D forms; geometric transformations; methods of realistic 3D images visualization; fill methods; algorithms of hidden lines and surfaces removal; definitions of shade; graphic dialog systems; operating systems and shells; modern computer graphic standards; graphic systems application	160
SD.10	Computer processing of experimental data methods and models of experimental data representation; data models and processing tasks classification; data processing and clusters; decrease of dimensions; regression models; methods of minimal squares; definition of non-linear regression parameters; interpolation and experimental data smoothing with spline-functions; program libraries and software packets; user interface in the automated processing systems; data visualization: methods, algorithms and software tools; architecture of automated systems for scientific researches	120
SD.11	Software metrology quantity analysis of software quality; quality criteria: complexity, correctness; stability; work-load; metric theory of software: basic notions; metrics; interval; level and category measurement; basic models, methods and algorithms of measurement; computational complexity: time, programming, informational; measurement and estimations of program complexity on different steps of software design; program correctness: formal, deterministic, stochastic, dynamic; standards, measurement and verification methods; program stability: basic notions, measurement methods; software and hardware for quantity analysis of software quality	80
SD.12	Neurocomputer systems Principles of organization and functionality of artificial neural networks (ANN); ANN education; education methods; associative ANN; linear association; Hopfield network; learning matrix; multilayer ANN: classification, approximation precision, self organizing network and data transformation networks; special ANN: realization; definition and programming languages; ANN hardware: neurocomputers, ANN applications	90
SD.13	Software market legal system software, software systems and information technology as products in information services market; marketing: pricing methods; sales channels; promotion;	60

	presentation, discounts, support, experience from software and IT leaders, software and IT-systems as intellectual property, legal protection of software in Russia and around the world; copyrights on software and IT; Russian law about copyright protection; software foundations and software registration procedure; local and technical methods of copyright protection; software and IT licensing; end-user agreement: rights and responsibilities of end-user and vendor	
SD.14	Computer microelectronic and circuit technology Basic notions of circuit theory; Laplace transformations for the circuit analysis; transfer function and its relationship with differential equation; impulse and frequency characteristics; discrete spectrum; non-periodic signals and their spectra; characteristics and parameters of semiconductors; diodes and transistors, their features and application; amplification networks for alternative and direct current; parameters and characteristics of electronic networks; basic elements of analog and digital devices; operational and solving amplifiers; active filters; generators; analog keys; logical elements; combination circuits synthesis; descriptors, encryptors, multiplexors; comparers; adders, programming logical matrixes, codes transformers, digital automats; triggers; counters; registers; memory circuits; analog-digital and digital-analog transformers; methods of computer-aided circuit design	150
SD.15	Specialization branches of learning	532
SD.16	Branches of learning by choice of student	240
F.00	Optional classes	450
F.01	Military education	450
	Total theoretical education	8262
P.01	Practical education	12 weeks
	Including computational practical education	4 weeks

Educational program of Engineer is designed for a term of 265 weeks, including 153 weeks of theoretical education, 14 weeks of pre-qualification education, 14 week for the qualification work, not less than 35 weeks of vacation, including 4 weeks after the graduation

Notes:

1. When developing the education program for Engineer, University has right to:
 - a. Change the number of hours for a particular branch of learning not more than 5% and not more than 10% for the whole education cycle. Also number of hours per week must not exceed the standards and must be not less than the minimum levels, specified in this document
 - b. Define number of hours for general humanitarian or socio-economical branches of learning (except of foreign language and fitness)
 - c. In humanitarian or socio-economical branches or learning to educate students in according to authorized lectures and different types of group and individual practical tasks, exams and programs (developed in the university according to the religious, national, ethnic and professional specifics as well as scientific expertise of the lecturers), which provide appropriate knowledge for the selected branches of learning
 - d. Define appropriate level of education for different parts of general humanitarian, socio-economical, mathematics and general science branches of learning (part 2) according to selected specialty
2. Number of educational hours must not exceed 27 per week. This doesn't include fitness and optional courses
3. Optional courses are a part of education plan of the university, but they are not mandatory for the students
4. Gradational projects are considered as a part of education program and are prepared within the hours, defined for a particular branch of learning
5. Name of the specializations are defined by education and methodology committee in the field of automation, electronics, microelectronics and radio technology. Names of branches of learning are defined by the university

Authors:

Education and methodology committee in the field of automation, electronics, microelectronics and radio technology

U. A Bichkov

Main committee of professional educational programs and technologies

U. G Tatur
B.E. Samodaev
E.P Popova