RUSSIAN MINISTRY OF EDUCATION AND SCIENCE

SAINT PETERSBURG ELECTROTECHNICAL UNIVERSITY "LETI"

Approved by:

Director of the Department of Education

_____ Galunin S. A.

«____»_____ 2022 г.

WORKING PROGRAM

Discipline

«Mathematical and Software of Computer Systems, Complexes and Computer Networks»

Specialization

2.3.5 – Mathematical and Software of Computer Systems, Complexes and Computer Networks

Saint Petersburg

2022

STRUCTURE DISCIPLINE

Curriculum:	
Faculty:	The School of Computer Technologies and Informatics
Department:	Department of Software Engineering and Computer Applications
Year	4
Semester	8

Activities:

Lectures

Individual study

Type of interim assessment

Exam (term)	8

The program was considered and approved on the Department of Software Engineering and Computer Applications meeting on 19/04/2022, protocol № 4.

SUBJECT SUMMARY
«MATHEMATICAL AND SOFTWARE OF COMPUTER SYSTEMS, COMPLEXES AND COMPUTER NETWORKS»

The course covers materials on the mathematical foundations of programming; computers, systems and networks; languages and programming systems; software development technologies; operating systems; methods of storage and access to data, organization of databases and knowledge; data protection and software systems.

GENERAL POINTS

1. Study and systematization of knowledge about the mathematical foundations of programming. Improve knowledge of the complexity of algorithms, the algebra of regular expressions, and the predicate calculus.

2. Formation of skills in building information-computer networks and distributed information processing. Improve skills in using different programming languages and paradigms, building translators, and optimizing programs.

3 Mastering methods of data storage and access, data and software systems protection.

CONTENT OF THE DISCIPLINE

Introduction

The content, purpose and importance of the discipline in the preparation of postgraduates, its relation with other disciplines and the preparation of PhD thesis. General classification of tasks to be solved.

Topic 1. Mathematical Fundamentals of Programming

Algorithm and its refinements: Turing machines, Markov algorithms, recursion. Equivalence of data of formal models of algorithms. Indecidability. Examples of indecidability.

Algorithm complexity. P versus NP problem. Polynomial-time reduction. Cook's Theorem on the NP-completeness of the well-formed formula feasibility problem. Examples of NP-completeness, approaches to their solution. Exact and approximate combinatorial algorithms.

Examples of efficient (polynomial) algorithms: fast search and sorting algorithms; polynomial algorithms for problems on graphs and networks (depth and width search, about the minimal spanning tree, about the shortest path and destinations).

Automata. Automata experiments. Kleene algebra. Kleene’s theorem for regular languages.

Boolean algebra. Boolean functions, canonical forms of Boolean functions. A complete set. Post’s completeness criterion. Minimization of Boolean functions in classes of normal forms.

First Order Predicate Calculus. Interpretation. Feasibility and general validity of the first-order formula. The concept of a model. Gödel's completeness theorem

Ratio and function. Equivalence and partition relation. The set factor. Partial order relation. Theoretical-multiple and algebraic definitions of the lattice, their equivalence. Lattice properties. Boolean lattices. Full lattices.

Formal languages and ways to describe them. Classification of formal grammars. Their use in lexical and syntactic analysis.

Lambda calculus, reduction rules, the uniqueness of the normal form and the rules for achieving it, representation of recursive functions.

Fundamentals of combinatorial analysis. Method of generating functions, inclusion and exclusion principles. Application Examples.

Error-correcting codes. Alphabetic coding. Data compression methods.

Fundamentals of Cryptography. The tasks of ensuring the confidentiality and integrity of information. Theoretical-information and theoretical-complexity approaches to determining strong cryptographic. U.S. DES encryption standard and Russian data encryption standard GOST 28147-89. Public key cryptosystem (RSA). Digital signature. Methods of key generation and distribution.

Topic 2. Computers, Systems and Networks

Modern computer architecture. Memory organization and processor architecture of modern computers. Paged and segmented virtual memory organization. A cache. Arithmetic pipeline and instruction pipeline, Parallel execution of independent commands, vector commands. Specialty processors. Machines that provide data flow-driven computations. Input/output (I/O), I/O channels and processors, interface devices

Classification of computer systems (CS) by the way of organizing parallel processing. Multiprocessor and multi-machine complexes. Computer clusters. Problem-oriented parallel structures: matrix CS, systolic structures, and artificial neural networks.

Purpose, architecture and principles of computer networks. Local and global area network, hardware and software for interconnecting different networks.

Methods and means of data transmission in computer networks, communication protocols.

Features of local network architecture (Ethernet, Token Ring, FDDI).

Internet, domain organization, TCP/IP. Computer networks and distributed data processing.

Topic 3. Programming Languages and Systems. Software Development Technology

Programming languages Procedural languages (Fortran, C). Functional programming languages (Lisp), logical programming (Prolog), object-based languages (Java).

Procedural languages. Basic control structures, program structure. Data: variables and constants, data types (Boolean, integer, float, char; range and enumeration types; pointer), data structure (array and record). Procedures (functions): procedure call, data transmission (by link, by value, by result), variable localization, side effects. Exception handling. Procedure libraries and their use.

Object-oriented programming. Classes and objects, inheritance, interfaces. Object environment. Reflection. Class libraries. Object handling tools (containers and iterators).

Distributed programming. Processes and synchronization. Semaphore, Hoare monitors. Object-oriented distributed programming. CORBA. Shared memory parallel programming. Threads. OpenMP interface. Parallelization of sequential programs. A distributed memory parallel program. SPMD and MIMD. Message Passing Interface (MPI).

Translator design. Optimizing translator structure. Intermediate representation: sequence of symbols, sequence of lexemes, syntax tree, abstract syntax tree. Levels of intermediate representation: high, medium, low. Forms of intermediate presentation.

Analysis of the source program in the compiler. Automated (regular) grammars and scanning, context free grammars and parsing, organization of a character table of a program with a block structure, hash functions. Top-down (LL(1)-grammars) and bottom-up (LR(1)-grammars) methods of parsing. Attribute grammars and semantic programs, abstract syntax tree design. Automatic construction of lexical and syntactic analyzers from formal descriptions of grammars. lex и yacc programs. Gentle system.

Optimizing programs when compiling them. Optimization of basic blocks, loop-invariant code motion. Analysis of control flow and data stream graphs. Dominance relation and its properties, construction of vertex dominance area boundary, selection of strongly related graph components. Dependency Graph Construction. Translation of a program to and from an SSA representation. Global and interprocedural optimization.

Object code generation in compilers. Retargetable compilers, gcc (GNU Compiler Collection). Term rewriting. Application of optimization heuristics (integer programming, dynamic programming) for automatic generation of object code generators (BEG, Iburg systems, etc.).

Machine-oriented languages, assembly language. Machine commands and constants. Commands to the translator; their types, principles of implementation. Macro means, macro calls, macro definition languages, conditional macro generation, implementation principles.

Programming Systems (PS), typical components of PS: languages, translators, link editors, debuggers, text editors. Modular programming. Module types. Linking control and data modules.

A software suite. The system part and the content. Software suite languages. Computer graphics. Machine graphics software and hardware. Graphics Packages.

Program development and maintenance technology. Program life cycle. Stages of development, degree and ways to automate them. Reverse engineering Decomposition and assembly techniques, inheritance mechanisms, encapsulation, and type assignments. Modules, interaction between modules, hierarchical program structures.

Debugging, testing, verification, and program complexity assessment. Test generation. Test generation systems. Program slices (slice, chop) and their use in debugging programs and for generating tests.

Program specification. Specification check. Schematic, structural, visual programming. User interface development, CUA standard, multimedia interface environments.

Topic 4. Operating Systems

Modes of computer systems operation, operating systems structure and functions. Basic blocks and modules. Hardware support for operating systems (OS) functions: interrupt system, memory protection, address conversion mechanisms in virtual memory systems, channel and peripheral device management.

Types of processes and their management in modern operating systems. Process representation, contexts, generation hierarchies, states, and interactions. Multi-tasking (multi-program) mode of operation. Process control commands. Process interaction. The client-server model and its implementation in modern OS.

Parallel processes, generation and control schemes. Organization of interaction between parallel and asynchronous processes: message exchange, mailboxes organization. Critical sections, mutually exclusive process primitives, Dijkstra semaphores and their extensions. The problem of deadlocks in asynchronous execution of processes, algorithms for deadlock detection and prevention.

Operational tools for process control in their implementation on parallel and distributed computing systems and networks: standards and software tools PVM, MPI, OpenMP, POSIX.

Single-level and multilevel disciplines of cyclic process maintenance on the central processor, quantum choice.

Data access management. File system, organization, distribution of disk storage Data exchange management between disk and RAM. The working set of pages (segments) of the program, algorithms to determine it.

External device control.

Optimizing computer multitasking. Windows, Unix, Linux. Organization features, the services provided by the user interaction.

Operational network management tools. ISO/OSI Open Systems Interoperability Reference Model. Routing and management of data flows in the network. Local and global networks. Network operating systems, the client-server model, network management tools in UNIX, Windows NT. TCP/IP protocol family, IP addresses structure and types, Internet domain addressing. TCP and UDP.

Remote access to network resources. Organization of e-mail and teleconferences. FTP and HTTP file transfer protocols, HTML, WEB-pages and WWW-servers development.

Topic 5 Methods for Storing and Accessing Data. Organization of Databases and Knowledge Bases

Data type concept. Abstract data types. Objects (basic properties and distinctive features).

Basic data structures, processing and search algorithms. Comparative characteristics of data storage and retrieval methods.

Basic concepts of relational and object data models. Theoretical foundations of the relational data model (RDM). Relational algebra, relational calculus. Functional dependencies and normalization.

CASE and its use in the design of databases (DB). Organization and design of the physical layer of the database. Indexing.

Generalized architecture, composition and functions of a database management system (DBMS). Characteristics of modern DB technologies. DBMS examples.

The basic principles of transaction management, logging and recovery.

SQL database language. Tools for defining and modifying the DB schema. Integrity constraints. Access control. Data manipulation tools.

SQL language standards. Interactive, embedded, dynamic SQ.

Basic concepts of client-server technology. Characteristics of SQL server and client. Client-server networking.

Information retrieval systems. Classification. Methods for implementing and speeding up the search.

Knowledge representation methods: procedural representations, logical representations, semantic networks, frames, production systems. Integrated methods of knowledge representation. Knowledge representation languages. Knowledge bases.

Expert systems (ES). Applications for ES. ES architecture. Mechanisms of inference, subsystems of explanation, communication, acquisition of knowledge ES. The life cycle of an expert system. Examples of specific ES.

Topic 6. Data and Software Systems Protection

Hardware and software to protect data and programs Data and programs protection with encryption.

Protection against unauthorized access in Windows NT. Security and resource access differentiation system in Windows NT. The NFTS file system and Windows NT services.

Protection against unauthorized copying. Methods of arranging non-copiable marks, setting up the program to be installed on a particular computer, setting up the hardware configuration.

Protection against destructive software impacts. Malware and its classification. Boot and file viruses, bookmark programs. Methods for detecting and removing viruses, software recovery.

Information protection in computer networks Novell Netware, Windows NT, etc.

Conclusion

Systematization of material. Recommendations for passing the exam.

If the discipline is implemented in groups with small numbers, classes on individual sections can take place in the form of introductory lectures, issuing and explaining the task on the topic, and the current control may take the form of presentation and explanation of the completed task by the graduate student.

General recommendations for individual assignments are available to the postgraduate student in printed or electronic form (on the University website), or the postgraduate student can obtain recommendations from the instructor in charge of the discipline during consultation hours. The assignment is formulated taking into account the topic of the postgraduate student's thesis research within the discipline studied.

METHODOLOGICAL SUPPORT FOR THE DISCIPLINE

The list of basic and additional academic literature

necessary for mastering the discipline

№	Name, bibliographic description	Number of copies in library (on department)
3	Compilers: Principles, Techniques, and Tools [Text] = Compilers: Principles, Techniques, and Tools: monograph / A.V.Aho, R.Seti, J.D.Ullman; Translated from English by I.V.Krasikov. - Moscow: Williams, 2001.	35
4	Introduction to Cryptography: New Mathematical Disciplines [Text]: textbook for universities / V.V. Yashchenko, N.P. Varnovsky, Y.V. Nesterenko et al: Ed. by V.V. Yashchenko. - SPb: Peter; M. ICNMO, 2001.	7
5	The art of computer programming = The art of computer programming: in 3 vol. Textbook: translated from English / D.E. Knuth. - Moscow : The World, 1976 - 1978	T1 – 10 T2 – 11 T3 – 11
6	Operating systems: design and implementation / E. Tanenbaum, A. Woodhull. - SPb. : Peter, 2006	26
7	Databases. Theory and Practice: textbook for universities in "Informatics and Computing Technology" and "Information Systems" / B.Y. Sovetov, V.V. Tsekhanovskiy, V.D. Chertovskoy. - Moscow: Higher School, 2005.	250
8	Networks: troubleshooting, maintaining & repairing networks [Text] = Troubleshooting, maintaining & repairing networks / S.J. Bigelow. - SPb: BHV-Peterburg, 2005.	8
9	Introduction to Discrete Mathematics [Text]: textbook for universities in Applied Mathematics / S.V. Yablonsky; Edited by V.A. Sadovnichy. - 3rd ed. - M.: Vyssh. shk., 2001	95
10	Introduction to Database Systems [Text] = An introduction to Database systems: a monograph / K.J. Deith; [Translated from English by Yu.G. Gordienko et al.] - 7th ed. - Moscow: Williams, 2001.	4
2	Parallel Computer Systems [Text] / V.V. Korneev. - Moscow: Nolige, 1999.	2

The list of information "Internet" resources used in the development of the discipline

№	URL
1	http://libgost.ru/gost/25-GOST_7_32_2001.html ГОСТ 7.32-2001 Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления.
1	http://www.intuit.ru/studies/courses/1/1/info
2	http://www.intuit.ru/studies/courses/1156/190/info
3	http://www.intuit.ru/studies/courses/508/364/info

Information technologies (operating systems, software for general and specialized purposes, as well as information reference systems), material, and technical base used in the implementation of the educational process in the discipline correspond to the requirements of the federal state educational standard of higher education.

A description of information technology and facilities is given in the EMCD

Specific forms and procedures for current knowledge control and intermediate certification including the list of examination questions (Appendix 1), as well as methodological guidelines for students to work independently in mastering disciplines are brought to the attention of students during the first weeks of training.

Appendix 1

The list of examination questions on the discipline

«Mathematical and software of computer systems, complexes and computer networks»

1. Algorithm and its refinements: Turing machines, Markov algorithms, recursion. Equivalence of data of formal models of algorithms. Indecidability. Examples of indecidability.
2. Algorithm complexity. P versus NP problem. Polynomial-time reduction. Cook's Theorem on the NP-completeness of the well-formed formula feasibility problem. Examples of NP-completeness, approaches to their solution. Exact and approximate combinatorial algorithms.
3. Examples of efficient (polynomial) algorithms: fast search and sorting algorithms; polynomial algorithms for problems on graphs and networks (depth and width search, about the minimal spanning tree, about the shortest path and destinations).
4. Automata. Automata experiments. Kleene algebra. Kleene’s theorem for regular languages.
5. Boolean algebra. Boolean functions, canonical forms of Boolean functions. A complete set. Post’s completeness criterion. Minimization of Boolean functions in classes of normal forms.
6. First Order Predicate Calculus. Interpretation. Feasibility and general validity of the first-order formula. The concept of a model. Gödel's completeness theorem.
7. Ratio and function. Equivalence and partition relation. The set factor. Partial order relation. Theoretical-multiple and algebraic definitions of the lattice, their equivalence. Lattice properties. Boolean lattices. Full lattices.
8. Formal languages and ways to describe them. Classification of formal grammars. Their use in lexical and syntactic analysis.
9. Lambda calculus, reduction rules, the uniqueness of the normal form and the rules for achieving it, representation of recursive functions.

10. Fundamentals of combinatorial analysis. Method of generating functions, inclusion and exclusion principles. Application examples.
11. Error correction code. Alphabetic code. Data compression methods.
12. Fundamentals of Cryptography. The tasks of ensuring the confidentiality and integrity of information. Theoretical-information and theoretical-complexity approaches to determining strong cryptographic. U.S. DES encryption standard and Russian data encryption standard GOST 28147-89. Public key cryptosystem (RSA). Digital signature. Key generation and distribution methods.
13. Modern computer architecture. Memory organization and processor architecture of modern computers. Paged and segmented virtual memory organization. A cache.
14. Arithmetic pipeline and instruction pipeline, Parallel execution of independent commands, vector commands. Specialty processors. Machines that provide data flow-driven computations. Input/output (I/O), I/O channels and processors, interface devices.
15. Classification of computer systems (CS) by the way of organizing parallel processing. Multiprocessor and multi-machine complexes. Computer clusters. Problem-oriented parallel structures: matrix CS, systolic structures, and artificial neural networks.
16. Purpose, architecture and principles of computer networks. Local and global area network, hardware and software for interconnecting different networks.
17. Methods and means of data transmission in computer networks, communication protocols.
18. Features of local network architecture (Ethernet, Token Ring, FDDI).
19. Internet, domain organization, TCP/IP. Computer networks and distributed data processing.
20. Programming languages Procedural languages (Fortran, C). Functional programming languages (Lisp), logical programming (Prolog), object-based languages (Java).

21. Procedural languages. Basic control structures, program structure. Data: variables and constants, data types (Boolean, integer, float, char; range and enumeration types; pointer), data structure (array and record). Procedures (functions): procedure call, data transmission (by link, by value, by result), variable localization, side effects. Exception handling. Procedure libraries and their use.
22. Object-oriented programming. Classes and objects, inheritance, interfaces. Object environment. Reflection. Class libraries. Object handling tools (containers and iterators).
23. Distributed programming. Processes and synchronization. Semaphore, Hoare monitors. Object-oriented distributed programming. CORBA. Shared memory parallel programming. Threads. OpenMP interface. Parallelization of sequential programs. A distributed memory parallel program. SPMD and MIMD. Message Passing Interface (MPI).
24. Translator design. Optimizing translator structure. Intermediate representation: sequence of symbols, sequence of lexemes, syntax tree, abstract syntax tree. Levels of intermediate representation: high, medium, low. Forms of intermediate presentation.
25. Analysis of the source program in the compiler. Automated (regular) grammars and scanning, context free grammars and parsing, organization of a character table of a program with a block structure, hash functions. Top-down (LL(1)-grammars) and bottom-up (LR(1)-grammars) methods of parsing. Attribute grammars and semantic programs, abstract syntax tree design. Automatic construction of lexical and syntactic analyzers from formal descriptions of grammars.
26. Optimizing programs when compiling them. Optimization of basic blocks, loop-invariant code motion. Analysis of control flow and data stream graphs. Dominance relation and its properties, construction of vertex dominance area boundary, selection of strongly related graph components. Dependency

Graph Construction. Translation of a program to and from an SSA representation. Global and interprocedural optimization.

27. Object code generation in compilers. Retargetable compilers, gcc (GNU Compiler Collection). Term rewriting. Application of optimization heuristics (integer programming, dynamic programming) for automatic generation of object code generators (BEG, Iburg systems, etc.).
28. Machine-oriented languages, assembly language. Machine commands and constants. Commands to the translator; their types, principles of implementation. Macro means, macro calls, macro definition languages, conditional macro generation, implementation principles.
29. Programming Systems (PS), typical components of PS: languages, translators, link editors, debuggers, text editors. Modular programming. Module types. Linking control and data modules.
30. A software suite. The system part and the content. Software suite languages. Computer graphics. Machine graphics software and hardware. Graphics Packages.
31. Program development and maintenance technology. Program life cycle. Stages of development, degree and ways to automate them. Reverse engineering Decomposition and assembly techniques, inheritance mechanisms, encapsulation, and type assignments. Modules, interaction between modules, hierarchical program structures.
32. Debugging, testing, verification, and program complexity assessment. Test generation. Test generation systems. Program slices (slice, chop) and their use in debugging programs and for generating tests.
33. Program specification. Specification check. Schematic, structural, visual programming. User interface development, CUA standard, multimedia interface environments.
34. Modes of computer systems operation, operating systems structure and functions. Basic blocks and modules. Hardware support for operating systems (OS) functions: interrupt system, memory protection, address

conversion mechanisms in virtual memory systems, channel and peripheral device management.

35. Types of processes and their management in modern operating systems. Process representation, contexts, generation hierarchies, states, and interactions. Multi-tasking (multi-program) mode of operation. Process control commands. Process interaction. The client-server model and its implementation in modern OS.
36. Parallel processes, generation and control schemes. Organization of interaction between parallel and asynchronous processes: message exchange, mailboxes organization. Critical sections, mutually exclusive process primitives, Dijkstra semaphores and their extensions. The problem of deadlocks in asynchronous execution of processes, algorithms for deadlock detection and prevention.
37. Operational tools for process control in their implementation on parallel and distributed computing systems and networks: standards and software tools PVM, MPI, OpenMP, POSIX.
38. Single-level and multilevel disciplines of cyclic process maintenance on the central processor, quantum choice.
39. Data access management. File system, organization, distribution of disk storage Data exchange management between disk and RAM. The working set of pages (segments) of the program, algorithms to determine it.
40. External device control.
41. Optimizing computer multitasking. Windows, Unix, Linux. Organization features, the services provided by the user interaction.
42. Operational network management tools. ISO/OSI Open Systems Interoperability Reference Model. Routing and management of data flows in the network. Local and global networks. Network operating systems, the client-server model, network management tools in UNIX, Windows NT. TCP/IP protocol family, IP addresses structure and types, Internet domain addressing. TCP and UDP.

43. Remote access to network resources. Organization of e-mail and teleconferences. FTP and HTTP file transfer protocols, HTML, WEB-pages and WWW-servers development.
44. Data type concept. Abstract data types. Objects (basic properties and distinctive features).
45. Basic data structures, processing and search algorithms. Comparative characteristics of data storage and retrieval methods.
46. Basic concepts of relational and object data models.
47. Theoretical foundations of the relational data model (RDM). Relational algebra, relational calculus. Functional dependencies and normalization.
48. CASE and its use in the design of databases (DB).
49. Organization and design of the physical layer of the database. Indexing.
50. Generalized architecture, composition and functions of a database management system (DBMS). Characteristics of modern DB technologies. DBMS examples.
51. The basic principles of transaction management, logging and recovery.
52. SQL database language. Tools for defining and modifying the DB schema. Integrity constraints. Access control. Data manipulation tools.
53. SQL language standards. Interactive, embedded, dynamic SQ.
54. Basic concepts of client-server technology. Characteristics of SQL server and client. Client-server networking.
55. Information retrieval systems. Classification. Methods for implementing and speeding up the search.
56. Knowledge representation methods: procedural representations, logical representations, semantic networks, frames, production systems. Integrated methods of knowledge representation. Knowledge representation languages. Knowledge bases.
57. Expert systems (ES). Applications for ES. ES architecture. Mechanisms of inference, subsystems of explanation, communication, acquisition of knowledge ES. The life cycle of an expert system. Examples of specific ES.

58. Hardware and software to protect data and programs. Data and programs protection with encryption.
59. Protection against unauthorized access in Windows NT. Security and resource access differentiation system in Windows NT. The NFTS file system and Windows NT services.
60. Protection against unauthorized copying. Methods of arranging non-copiable marks, setting up the program to be installed on a particular computer, setting up the hardware configuration.
61. Protection against destructive software impacts. Malware and its classification. Boot and file viruses, bookmark programs. Methods for detecting and removing viruses, software recovery.
62. Information protection in computer networks Novell Netware, Windows NT, etc.