ALGORITHMS AND METHODS OF CALCULATIONS

Syllabus

<table>
<thead>
<tr>
<th>Cycle of Higher Education</th>
<th>Field of Study</th>
<th>Speciality</th>
<th>Education Program</th>
<th>Type of Course</th>
<th>Mode of Studies</th>
<th>Year of studies, semester</th>
<th>ECTS workload</th>
<th>Testing and assessment</th>
<th>Course Schedule</th>
<th>Language of Instruction</th>
<th>Course Instructors</th>
</tr>
</thead>
<tbody>
<tr>
<td>First cycle of higher education (Bachelor’s degree)</td>
<td>12 Information Technologies</td>
<td>123 Computer Engineering</td>
<td>Computer Systems and Networks</td>
<td>Normative</td>
<td>full-time</td>
<td>2 year (spring semester)</td>
<td>5 credits (ECTS). Time allotment – 15 hours, including 36 hours of classroom work, and 78 hours of self-study.</td>
<td>3 semester – Final test</td>
<td>1 class per week by the timetable <a href="http://rozklad.kpi.ua/">http://rozklad.kpi.ua/</a></td>
<td>English</td>
<td></td>
</tr>
<tr>
<td>Lecturer: Doctor of Technical Sciences, Professor, Mykhailo Anatoliyovych Novotarskyi</td>
<td>Teacher of laboratory work: Doctor of Technical Sciences, Professor, Mykhailo Anatoliyovych Novotarskyi</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Access to the course

Outline of the Course

1. Course description, goals, objectives, and learning outcomes

The academic discipline "Algorithms and methods of calculations" is designed to provide basic mathematical training of students in the field of methods for solving scientific and engineering problems and programming, taking into account the peculiarities of the algorithmic implementation for numerical methods.

The subject "Algorithms and methods of calculations" has the code PO9 in the list of components of the educational program and belongs to the cycle of professional training.

Within the first part of the course the concept of algorithm, basic properties and types of algorithms are considered, three main types of universal algorithmic models are studied: recursive functions, Turing machine and normal Markov algorithms.


The purpose of the "Algorithms and methods of calculations" course is study of modern methods and technologies for developing and evaluating algorithms, fundamental training of students in choosing and using algorithm methods. The calculation methods should be resistant to errors, effective for calculating mathematical problems, analysis of approximate solutions, creating highly efficient algorithms and programs that take into account the peculiarities of the implementation of calculations, and promoting the development of logical and analytical thinking of students.

The subject of the academic discipline consists of:
- methods of algorithm analysis;
- methods for determining the computability and solvability of functions;
- Numerical methods for solving mathematical problems.

According to the requirements of the educational program, students after mastering the credit module must demonstrate the following learning outcomes:

Knowledge: basic calculation methods and corresponding effective algorithms for solving mathematical problems on a PC; calculation of approximate values and interpolation of functions, solution of systems of linear algebraic equations and nonlinear equations, numerical integration and differentiation, solution of differential equations with choice of method. Use of splines in interpolation of tables of values of functions and calculation of integrals. Analysis and processing of experimental results and a posteriori estimates of errors.

Ability to choose and justify the use in practice of certain calculation methods that are resistant to errors and most effective in their practical implementation on a PC.

Experience: the student must know the basic principles of developing algorithms and software for solving problems on a PC. He should have skills to explore computational algorithms, identify their advantages and disadvantages, choose the optimal algorithms for solving problems, data processing and develop programs for solving problems; perform analysis and processing of the results of problem solving, use optimization methods, a posteriori estimates of errors in solving problems by numerical methods.

2. Prerequisites and post-requisites of the course (the place of the course in the scheme of studies in accordance with curriculum)

The material of the discipline is related to the materials studied in the courses "Higher Mathematics: Differential Calculus, Linear Algebra", "Programming" and "Discrete Mathematics".

The knowledge and practical skills acquired in this discipline can be applied in the study of the following courses: "Computer Modeling", "Fundamentals of Software Engineering", "System Programming", "Distributed Computing Technology", "Network and Information Technology" and other.

3. Content of the course

The "Algorithms and methods of calculations" academic discipline includes the study of the following topics.

Section 1. Fundamentals of algorithm theory

Topic 1.1. Introduction. Main methods and tasks, connection with other disciplines.
The main sections and issues to be explored. Contents of individual and laboratory works. Features of laboratory work and individual tasks, requirements for them.

Topic 1.2. The concept of algorithm, properties and methods of algorithm setting.


Topic 1.4. Error. Classification, sources and rules for calculating errors. Significant cypher, rules for counting numbers according to Bradys, the general formula for error.


Topic 1.6. Turing machine. Functions, which are calculated by Turing.

Section 2. Fundamentals of numerical methods


**Topic 2.10. Iterative asynchronous methods.** Method of asynchronous iterations. Method of asynchronous iterations with fixed points.

**Basic:**

**Additional:**

---

**Educational content**

**4. Methodology**

**Profile for topics**

<table>
<thead>
<tr>
<th>Names of sections and topics</th>
<th>Number of academic hours</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total Lectures Practice Laboratory work CPC</td>
</tr>
<tr>
<td><strong>SECTION 1. THEORY OF ALGORITHMS</strong></td>
<td>0</td>
</tr>
<tr>
<td><strong>Topic 1.1. Introduction. Methods and tasks of the course, connection with other disciplines.</strong> The main sections and issues to be explored. Contents of independent and laboratory works. Features of laboratory work and individual tasks, requirements.</td>
<td>7 1 2 4</td>
</tr>
<tr>
<td><strong>Topic 1.2. The concept of algorithm, properties and methods for algorithm setting.</strong></td>
<td>7 1 2 4</td>
</tr>
<tr>
<td><strong>Topic 1.3. Measures of complexity of algorithms.</strong> Task classes P and NP</td>
<td>10 2 4 4</td>
</tr>
<tr>
<td>Names of sections and topics</td>
<td>Number of academic hours</td>
</tr>
<tr>
<td>---------------------------------------------------------------------------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td></td>
<td>Total</td>
</tr>
<tr>
<td>Topic 1.4. Error. Classification, sources and rules for calculating errors. Significant cypher, rules for counting numbers according to Bradys, the general formula for error.</td>
<td>10</td>
</tr>
<tr>
<td>Test work 1</td>
<td>4</td>
</tr>
<tr>
<td>Topic 1.6. Turing machine. Functions calculated by Turing machine.</td>
<td>4</td>
</tr>
<tr>
<td>Topic 1.7. Normal Markov algorithms. Equivalence of different universal algorithmic models. Markov substitutions. Normal algorithms, which use words. Computed Markov function. The principle of Markov normalization. Coincidence of the class of all normally computed functions with the class of all functions computed by Turing.</td>
<td>5</td>
</tr>
<tr>
<td>SECTION 2. CALCULATION METHODS</td>
<td>0</td>
</tr>
<tr>
<td>Names of sections and topics</td>
<td>Number of academic hours</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td></td>
<td>Total</td>
</tr>
</tbody>
</table>
### Topic 2.9. Solving partial differential equations


### Topic 2.10. Iterative asynchronous methods

Method of asynchronous iterations. Method of asynchronous iterations with fixed points.

### Test work 2

4

### Final test

6

### Total in the semester:

150 36 36 78

---

<table>
<thead>
<tr>
<th>№</th>
<th>3/n</th>
<th>The title of the lecture topic and a list of key issues</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction. Methods and tasks of the course, connection with other disciplines.</td>
<td>The main sections and issues to be explored. Contents of independent and laboratory works. Features of laboratory work and individual tasks, requirements. <strong>The concept of algorithm, properties and methods of setting algorithms.</strong> Stages of problem solving, algorithm concept (algorithm definition, numerical algorithms, logical algorithms, sequential algorithms, parallel algorithms). General properties of algorithms (discreteness, mass, determinism, elementary steps, efficiency). Ways to set algorithms (verbal, graphical, pseudocode, software). Image of the algorithm in the form of a block diagram (rules of graphic design of the block diagram, linear algorithm, branching algorithm cyclic algorithm).</td>
</tr>
<tr>
<td>3</td>
<td>Error. Classification, sources and rules for calculating errors.</td>
<td>Mathematical model error, input data error, method error, rounding error. Absolute and relative error, the limit of absolute and relative error. Calculation with strict errors. Significant figure, calculation rules without strict consideration of errors. Rules for counting numbers according to Bradys, the general formula for error.</td>
</tr>
<tr>
<td>No</td>
<td>The title of the lecture topic and a list of key issues</td>
<td></td>
</tr>
<tr>
<td>----</td>
<td>----------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>№</td>
<td>The title of the lecture topic and a list of key issues</td>
<td></td>
</tr>
<tr>
<td>----</td>
<td>------------------------------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td><strong>Iterative asynchronous methods. Basic concepts.</strong> Method of chaotic iterations. Method of asynchronous iterations. Method of asynchronous iterations with fixed points</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td><strong>Turing machine.</strong> Functions calculated by Turing machine. Scope of the Turing machine. Mathematical model of the Turing machine. Tape. Reading head. Internal memory of the car. Control device. Turing machine program. The operation of the Turing machine. Examples of Turing machines operating in the alphabet {a, b}. An example of a Turing machine running in the alphabet {a, b, c}. Description of the class of functions calculated by Turing. Examples of functions calculated by Turing.</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td><strong>Normal Markov algorithms. Equivalence of different universal algorithmic models.</strong> The emergence of the theory of normal algorithms. Markov substitutions. Normal algorithms and their application to words. Normally calculated functions. An example of an algorithm for a computational function in the extended alphabet. The principle of Markov normalization. Coincidence of the class of all normally computed functions with the class of all functions computed by Turing.</td>
<td></td>
</tr>
</tbody>
</table>

**Self-study**

The main task of the cycle of laboratory classes is to provide students with the necessary practical skills to develop algorithms and software for solving computational problems on a PC; research of algorithms in terms of computational complexity, mastering the basic methods and algorithms of interpolation of functions, development of algorithms for solving nonlinear equations and systems of linear algebraic equations.

**Profile for laboratory work**
<table>
<thead>
<tr>
<th>№</th>
<th>The title of laboratory work</th>
<th>Number of acad. hours</th>
</tr>
</thead>
</table>
| 1  | Laboratory work № 1. The concept of algorithm. Setting algorithms in the form of block diagrams.  
- Section 1. Topic 1.1,  
- Section 1. Topic 1.2. | 2                     |
| 2  | Laboratory work № 2. Computational complexity of sorting algorithms.  
- Section 1. Topic 1.3,  
- Section 1. Topic 1.4. | 4                     |
| 3  | Laboratory work № 3. Interpolation of functions. Interpolation polynomials  
- Section 2. Topic 2.1,  
- Section 2. Topic 2.2. | 4                     |
| 4  | Laboratory work № 4. Solving nonlinear equations on a computer.  
- Section 2. Topic 2.3,  
- Section 2. Topic 2.4. | 4                     |
| 5  | Laboratory work № 5. Solving systems of linear algebraic equations.  
- Section 2. Topic 2.5,  
- Section 2. Topic 2.6. | 4                     |

**Profile for lectures**

| №  | The title of the lecture topic and a list of key issues  
(List of teaching tools, references and tasks for self-study) |
|----|----------------------------------------------------------------|
| 1  | **Introduction. Methods and tasks of the course, connection with other disciplines.**  
(Presentation 1 and textbook on the website [http://amodm.pp.ua](http://amodm.pp.ua))  
References  
DSTU. Schemes of algorithms, programs, data and systems. Symbols and rules of execution  
DSTU. Schemes of algorithms and programs. Rules of execution.  
DSTU. Schemes of algorithms and programs. Symbols are graphic |
| 2  | **Measures of complexity of algorithms. Problem classes P and NP.**  
(Lecture 2 and Presentation 2 on the website http://amodm.pp.ua)  
References  
| 3  | **Error. Classification, sources and rules for calculating errors.**  
Tasks for self-study. Sources of errors and their classification. Rounding numbers on a computer.  
Algorithms of calculations. Errors in calculating the values of functions. Inverse tasks of the theory of errors.  
(Lecture 3 and Presentation 3 on the website [http://amodm.pp.ua](http://amodm.pp.ua))  
References  
<table>
<thead>
<tr>
<th>No</th>
<th>The title of the lecture topic and a list of key issues (List of teaching tools, references and tasks for self-study)</th>
</tr>
</thead>
</table>
| 4 | **Interpolation and the problem of interpolation. Generalized polynomials. Interpolation by algebraic polynomials.**  
**Tasks for self-study.** Aitken interpolation polynomial. Finite high order differences. Interpolation formulas of Gauss, Sterling, Bessel, features of application.  
(Lecture 4 and Presentation 4 on the website [http://amodm.pp.ua](http://amodm.pp.ua))  
**References**  
| 5 | **Interpolation and the problem of interpolation (continued). Newton's interpolation polynomial.**  
**Tasks for self-study.** Spline interpolation, construction of algorithms, application feature.  
(Lecture 5 and Presentation 5 on the website [http://amodm.pp.ua](http://amodm.pp.ua))  
**References**  
Fraser D.C. Newton's Interpolation Formulas. – Dover Publications. –2006. –272 p |
| 6 | **Methods of numerical differentiation and integration.**  
**Tasks for self-study.** Quadrature formulas for calculating integrals, estimating coefficients, estimating integration errors. Estimation of stability of quadrature formulas to errors of calculation function values. Analysis of error behavior in the interpolation interval. Quadrature formulas (Gaussian formulas).  
(Lecture 6 and Presentation 6 on the website [http://amodm.pp.ua](http://amodm.pp.ua))  
**References**  
| 7 | **Methods for solving nonlinear equations.**  
**Tasks for self-study.** Comparison of methods for solving nonlinear equations by the number of iterations.  
(Lecture 7 and Presentation 7 on the website [http://amodm.pp.ua](http://amodm.pp.ua))  
**References**  
| 8 | **Solving systems of linear algebraic equations.**  
(Lecture 8 and Presentation 8 on the website [http://amodm.pp.ua](http://amodm.pp.ua))  
**References**  
| 9 | **Iterative methods for solving systems of algebraic equations.**  
**Tasks for self-study.** Solving systems of linear algebraic equations by the method of choosing the principal element, estimating the rate of convergence of the iterative process of solving SLAE.  
(Lecture 9 and Presentation 9 on the website [http://amodm.pp.ua](http://amodm.pp.ua))  
**References**  
<p>| 10 | <strong>Solving systems of nonlinear algebraic equations.</strong> |</p>
<table>
<thead>
<tr>
<th>No</th>
<th>Tasks for self-study.</th>
<th>References</th>
</tr>
</thead>
</table>
| 11 | **Numerical solution of differential equations.**  
| 12 | **Numerical methods for solving the boundary value problem for the ordinary differential equation (ODE).**  
| 13 | **Solving partial differential equations.**  
Tasks for self-study. Formulate and solve a boundary value problem in a two-dimensional quadratic domain with boundary conditions of the first kind for an elliptic equation.  
| 14 | **Solving partial differential equations (continued).**  
Tasks for self-study. Construct a computational template for an elliptical domain.  
| 15 | **Iterative asynchronous methods.**  
Tasks for self-study. To form a difference scheme of calculations by the method of asynchronous iterations on a cross-shaped template.  
(Lecture 15 and Presentation 15 on the website [http://amodm.pp.ua](http://amodm.pp.ua)) | |
<table>
<thead>
<tr>
<th>No</th>
<th>Title of the lecture topic and a list of key issues (List of teaching tools, references and tasks for self-study)</th>
</tr>
</thead>
</table>

**Policy and Assessment**

5. **Course policy**

During classes in the "Algorithms and methods of calculations" academic discipline students must follow certain disciplinary rules:

- it is forbidden to be late for classes;
- at the entrance of the teacher, as a sign of greeting, persons studying at “KPI Igor Sikorsky” must stand up;
- no extraneous conversations or other noise that interferes with the classes;
- leaving the classroom during the lesson is allowed only with the permission of the teacher.
- it is not allowed to use mobile phones and other technical means without the permission of the teacher.

Laboratory works are handed over personally with preliminary check of the theoretical knowledge, which is necessary for performance of laboratory work. Verification of practical results includes verification of code and performance of test tasks.

During the training, the teacher has the right to receive up to 5 incentive points for early performance of laboratory work, for creative approach in performing an individual task or for active participation in the discussion of issues related to the topic of the lecture or practical lesson.

For the performance and delivery of laboratory work after the specified deadline, for a significant number of missed classes, or for violation of the rules of conduct in the classroom, the teacher may assign up to 5 penalty points.
When conducting control measures and performing laboratory work, graduate students must follow the rules of academic integrity. If a significant percentage of write-offs or plagiarism is detected, the teacher may refuse to accept the work and demand a fair implementation of the curriculum.

6. Monitoring and grading policy

At the first class the students are acquainted with the grading policy which is based on Regulations on the system of assessment of learning outcomes [https://document.kpi.ua/files/2020_1-273.pdf](https://document.kpi.ua/files/2020_1-273.pdf)

Types of control in the "Algorithms and methods of calculations" discipline include:

- Laboratory works: laboratory works
- Current control: testing by closed tests.
- Final test: closed test or interview with a teacher.
- Bonuses.

Conditions of admission to semester control: semester rating more than 40 points.

At the first class the students are acquainted with the grading policy which is based on Regulations on the system of assessment of learning outcomes [https://document.kpi.ua/files/2020_1-273.pdf](https://document.kpi.ua/files/2020_1-273.pdf) The student's rating in the course consists of laboratory work scores (R1), current test scores (R2), final test scores (R4) and bonus scores (R4)

\[ R_s = R_1 + R_2 + R_3 + R_4 = 100 \text{ scores} \]

The maximum average weight score is equal to:

\[ R_1 = 5 \text{ labs} \times 10 \text{ points} = 50 \text{ points} \]

\[ R_2 = 5 \text{ current test points} \times 5 \text{ points} = 25 \text{ points} \]

\[ R_3 = \text{Final test} = 25 \text{ points} \]

\[ R_4 = \text{from 0 to 10 points for outstanding achievements in learning and completing additional tasks} \]

According to the university regulations on the monitoring of students' academic progress ([https://kpi.ua/document_control](https://kpi.ua/document_control)) there are two assessment weeks, usually during 7th/8th and 14th/15th week of the semester, when students take the Progress and Module tests respectively, to check their progress against the criteria of the course assessment policy.

The students who finally score the required number of points (≥60) can:
- get their final grade according to the rating score;
- perform personal test or to pass an interview with the teacher in order to increase the grade.

Students whose final performance score is below 60 points but more than 30 are required to complete personal test or to pass an interview with the teacher. If the final grade is lower than the grade, which the student gets for his semester activity, a strict requirement is applied - the student's previous rating is canceled and he receives a grade based on the results of the Fail/Pass test. Students whose score is below 30 are not allowed to take the Fail/Pass Test.

The final performance score or the results of the Fail/Pass Test are adopted by university grading system as follows:
### Score and Grade Table

<table>
<thead>
<tr>
<th>Score</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>100-95</td>
<td>Excellent</td>
</tr>
<tr>
<td>94-85</td>
<td>Very good</td>
</tr>
<tr>
<td>84-75</td>
<td>Good</td>
</tr>
<tr>
<td>74-65</td>
<td>Satisfactory</td>
</tr>
<tr>
<td>64-60</td>
<td>Sufficient</td>
</tr>
<tr>
<td>Below 60</td>
<td>Fail</td>
</tr>
<tr>
<td>Course requirements are not met</td>
<td>Not Graded</td>
</tr>
</tbody>
</table>

7. **Additional information about the course**

Teaching the academic discipline "Algorithms and methods of calculations" for the field of "Computer Engineering" has its own specifics, which is due to the fact that the scope of methods of machine data analysis is constantly expanding. Widespread informatization leads to the accumulation of huge amounts of data in research, production, transport, and health care. Forecasting, control, and decision-making tasks often require machine data analysis and machine learning, as previously such tasks were either not posed at all or solved by precise methods that required large computing power due to the complexity of the respective algorithms.

**Syllabus of the course**

*Is designed by teacher* Doc.Sci, professor Mykhailo Novotaerskyi

*Adopted by Department of Computing Technics* (protocol № , date )

*Approved by the Faculty Board of Methodology* (protocol № , date )