Syllabus
«Expert systems»

Lecturer: *Oleksandr Gozhyi*
Doctor of Technical Sciences, Professor, Professor of the Intelligent Information Systems Department

Expected learning outcomes
As a result of studying the discipline, the student
must know:

- features of knowledge engineering technology;
- technology of design and development of expert systems (ES);
- formalization methods and strategies for solving intellectual tasks in ES;
- methods of obtaining and structuring knowledge in ES;
- models of knowledge presentation and methods of logical inference in ES.

***must be able to*:**

- to formulate tasks and develop algorithms for solving them during ES design;
- carry out methodological substantiation of the research in the design of ES;
- apply knowledge presentation methods and decision-making strategies when designing ES;
- apply modern methods and programming languages of the highest level when designing ES.
- choose and apply methods of formalization of knowledge for building ES;
- use the main technological methods of design and development of ES.

Prerequisites
According to the educational program, it is necessary to first acquire

The scope of the discipline: 5,5 ECTS credits (16 hours of lectures, 48 hours of practice).

Purpose: familiarization with the main technologies and methods of creating expert systems as one of the main areas of development of intelligent information systems; formation of knowledge on technologies for the development of expert systems in various applications, studying their structural and functional organization, the form of presentation and properties of knowledge bases, methods of processing knowledge, acquiring skills in storing and processing data and knowledge, developing the ability to apply knowledge in practice. Studying this course provides an increase in professional competence in the field of intellectual technologies, contributes to the expansion of professional horizons and the ability to navigate trends and directions of development of modern information technologies.

The content of the discipline

Topic 1. Expert systems. Evolution of intelligent information systems. Subject and content of the course. Connection of the course with other disciplines. Classification of IIS. Expert systems. Characteristic features. Terms of use. Problem areas of the ES. Using expert systems to support decision-making.
Topic 2. Architecture and presentation of knowledge in ES. Components of ES. Forms of organization and presentation of knowledge in ES. Visual and problematic knowledge. Declarative and procedural forms of knowledge presentation. Languages of presentation of knowledge in ES. The solution search space. Logical and heuristic methods of decision-making in ES. Reasoning based on deduction, induction and analogy. Unclear derivation of knowledge. A generalized scheme for solving problems in the ES. Acquisition of knowledge. Testing the correctness of knowledge. Generation of explanations. Dialogic user interaction with ES, use of templates, menus, natural language.
Topic 3. Production, frame and object-oriented methods of developing knowledge bases. Simple and complex rules. Static and dynamic knowledge bases. Conjunctive and

knowledge in the following disciplines: "Higher mathematics", "Fundamentals of programming", "Object-oriented programming", "Probability theory and mathematical statistics", "Algorithms and data structures", "System analysis", "Methods and systems of artificial intelligence", "Organization of databases and knowledge".

Consequences

Competences, knowledge and skills acquired within the framework of studying this discipline can be applied to obtain reasonable research results and increase the scientific level of qualification work**.**

Technical support

Laboratory work on the discipline is carried out in computer classes using the software environment for the development of CLIPS external systems (*C Language Integrated Production System*)**.**

Deadline Policy

Works that are submitted late without good reason will be assigned a lower grade**.**

Academic Integrity Policy

Provides independent performance of practices. Write-off (including using mobile devices) is prohibited. The work is not credited in case of detection of plagiarism or write-off.

disjunctive forms of representations of prerequisites when designing rules. Frames. Output on frames. ES user interface.

Topic 4. ES design. Determination of the purpose and scope of ES application, classes of tasks to be solved and types of knowledge used. Selection of experts and engineers by knowledge, allocation of resources. Parameterization of solved tasks: goals, limitations, hypotheses, concepts, initial data. Development of prototypes, development and modification of the project. Participants in the design process: visual experts, knowledge engineers, end users, their interaction. The paradox of knowledge engineering. ES development tools: programming languages, knowledge representation languages, generators, shells, design automation tools.

Topic 5. Means of acquiring and explaining knowledge. Development of an intelligent interface. Inductive method of acquiring knowledge. The choice of the form of interaction of end users with the ES. Intelligent editors. Application of graphic means of input and output. Morphological, syntactic, semantic analysis of requests and synthesis of output messages. Designing help, hints, explanations.

Topic 6. Methods of derivation in ES: direct and reverse. Confidence coefficients. Probable output in the ES. Probabilistic and fuzzy methods of handling uncertainty in the design of rules. Strategies of direct, reverse and mixed chains of logical derivation of knowledge. Interaction of a set of rules. Application of meta rules.

Topic 7. Methods of overcoming uncertainties in ES. Uncertainties in the presentation of knowledge. Probabilistic methods of overcoming uncertainties. Method of confidence coefficients. Deduction according to the Demster-Schäffer rule.

Topic 8. CLIPS expert system programming language. Introductory description of practical concepts that are necessary for the formation of ES using CLIPS. CLIPS support capabilities for 3 types of programming: rule-based, object-oriented, and procedural. A detailed description of the components of ES, which is based on rules: a list of facts, a knowledge base, a logical inference machine. Development of a pattern matcher. Modular design, enforcement management and rule effectiveness.

Topic 9. ES development using CLIPS. Testing the accuracy of problem solving by experts. Selection of test cases. Complete verification of the solution space. Study of accuracy indicators. Testing consumer qualities of ES by potential users. Task execution time, user-friendliness of the interface, help tools and explanations. Application of testing tools: tracing and explanations, semantic analysers, control points of statistics collection, restructuring.

Evaluation criteria of laboratory works / practices / individual works / reports / projects

Maximum number of points – a student with high quality independently completed the entire amount of work, answers all questions related to the completed work, and makes additional calculations offered by the teacher. The teacher has no complaints about the implementation and requirements for the performance of the work.
70%-99% of the maximum number of points – the student completed all tasks with sufficient quality, but in the process of work he made some mistakes, which, after being pointed out by the teacher, he corrected himself. He answers some questions incorrectly. Additional calculations proposed by the teacher are done with some effort. Not all requirements for performance of work are met.
Approximately 40%-69% of the maximum number of points – the student independently completed all the work, but the quality of the implementation is insufficient (calculation errors, not all work requirements are met). The answers to questions about the performance of work are not quite clear. There are mistakes in the answers.
Approximately 1%-39% of the maximum number of points – the student did not complete all the work independently, while the quality of the implementation was insufficient (errors in calculations, does not comply with the requirements for the design of the work). He does not answer questions about the performance of work clearly. There are gross mistakes in the answers.
0 points – the student did not complete the entire amount of work, or did it with gross errors. He has problems with calculations, does not know the theoretical material, the software implementation does not meet the requirements.