SOFTWARE MODELING AND ANALYSIS

COURSE SYLLABUS

Code and name of specialty	121 Software Engineering	**Institute / faculty**	Faculty of Computer Science and Software Engineering
Program name	“Software Engineering”	**Department**	Software Engineering and Management Information Technologies
Type of program	Educational and Professional	**Language of instruction**	Ukrainian, English

LECTURER

Oleksandr Shmatko ***oleksandr.shmatko@khpi.edu.ua***

Ph.D., Ass. Prof. at the Department of Software Engineering and Management Information Technologies of NTU «KhPI». Prepared and published more than 70 publications, 3 collective monographs, 2 textbooks with the University stamp, 8 articles in publications indexed in Scopus. (h-index = 7, i10-index = 4 in Google Academy-https://scholar.google.com/citations?user=Wyv6esuaaaaaj&hl=ru; ORCID iD-https://orcid.org/0000-0002-2426-900x.
Leading lecturer of courses: *Software modeling and analysis (bachelors) (in English and Ukrainian), Advanced technologies and areas of development of intelligent software systems (Masters) (in English and Ukrainian), Modern technologies of web application development (PhD) (in Ukrainian)*

GENERAL DESCRIPTION OF THE COURSE

Summary	The course "Software modeling and analysis" is an academic course from the cycle of professional mandatory training in specialty 121"Software Engineering". It is taught in the seventh semester in the amount of 150 hours.(5 ECTS credits), in particular: lectures – 32 hours, laboratory works – 32 hours, independent work-84 hours. The course provides two content modules and one modular Control work. The discipline ends with an exam.
Course objectives	Training students in the methodology of System Analysis and modeling, which allow at the stage of creating software to solve the following main tasks: providing the necessary functionality of the software and adaptability to constantly changing conditions of its functioning; designing data objects implemented in the system; designing interface tools (screen forms, reports) that will ensure the execution of data requests; choosing a specific environment or technology for project implementation.
Types of classes and control	Lectures, laboratory classes. Continuous assessment – laboratory works, intermediate modular assessment, control work. Final assessment – exam.
Term	7

Student workload (credits) / Type of course	5/ Mandatory (elective)	**Lectures (hours)**	32	**Laboratory classes (hours)**	32	**Self-study (hours)**	84

Program competences	GC 01. Ability to abstract thinking, analysis and synthesis. GC 05. Ability to learn and master modern knowledge. GC 06. Ability to search, process and analyze information from various sources. PC14. Ability to participate in software design, including modelling (formal description) of its structure, behavior and functioning processes.

	PC15. Ability to develop architectures, modules and components of software systems. PC16. Ability to formulate and ensure software quality requirements in accordance with customer requirements, specifications and standards. PC17. Ability to adhere to specifications, standards, rules and recommendations in the professional field in the implementation of life cycle processes. PC19. Knowledge of information data models, the ability to create software for data storage, retrieval and processing. PC26. Ability to algorithmic and logical thinking.

Learning outcomes	Teaching and learning methods	Forms of assessment (continuous assessment CAS, final assessment FAS)
PO01. Analyze, purposefully search for and select the necessary information and reference resources and knowledge to solve professional problems, taking into account modern advances in science and technology.	Interactive lectures with presentations, discussions, laboratory classes, teamwork, case method, student feedback method, problem-based learning	Written individual assignments for laboratory works (CAS), assessment of knowledge in laboratory classes (CAS), express surveys (CAS), online tests (CAS), final/semester control in the form of a semester exam, according to the schedule of the educational process (FAS)
PO03. Know the basic processes, phases and iterations of the software life cycle.	Interactive lectures with presentations, discussions, practical exercises, teamwork, case study, research, project training	Written individual assignments for laboratory works (CAS), assessment of knowledge in laboratory classes (CAS), express surveys (CAS), online tests (CAS), final/semester control in the form of a semester exam, according to the schedule of the educational process (FAS)
PO05. Know and apply relevant mathematical concepts, methods of domain, system and object-oriented analysis and mathematical modelling for software development.	Interactive lectures with presentations, discussions, practical exercises, teamwork, case study, research, project training	Written individual assignments for laboratory works (CAS), assessment of knowledge in laboratory classes (CAS), express surveys (CAS), online tests (CAS), final/semester control in the form of a semester exam, according to the schedule of the educational process (FAS)
PO09. Know and be able to use methods and tools for collecting, formulating and analyzing software requirements.	Interactive lectures with presentations, discussions, laboratory classes, teamwork, case method, student feedback method, problem-based learning	Written individual assignments for laboratory works (CAS), assessment of knowledge in laboratory classes (CAS), express surveys (CAS), online tests (CAS), final/semester control in the form of a semester exam, according to the schedule of the educational process (FAS)
PO11. Choose source data for design, guided by formal methods of describing requirements and modelling.	Interactive lectures with presentations, discussions, laboratory classes, teamwork, case method, student feedback method, problem-based learning	Written individual assignments for laboratory works (CAS), assessment of knowledge in laboratory classes (CAS), express surveys (CAS), online tests (CAS), final/semester control in the form of a semester exam, according to the schedule of the educational process (FAS)
PO14. Put into practice the tools of domain analysis, design, testing, visualization, measurement and documentation of software.	Interactive lectures with presentations, discussions, laboratory classes, teamwork, case method, student feedback method, problem-based learning	Written individual assignments for laboratory works (CAS), assessment of knowledge in laboratory classes (CAS), express surveys (CAS), online tests (CAS), final/semester control in the form of a semester exam, according to the schedule of the educational process (FAS)
PO23. Be able to document and present the results of software	Interactive lectures with presentations, discussions, practical exercises, teamwork,	Written individual assignments for laboratory works (CAS), assessment of knowledge in laboratory classes (CAS), express surveys (CAS), online tests (CAS), final/semester control in the form of a semester exam, according to the schedule of the educational process (FAS)

development.	case study, research, project training	

ASSESSMENT AND GRADING

	score (points) for all types of learning activities	ECTS grading scale	The national grading scale		
Ranges of points corresponding to grades	90-100	A	excellent	**Allocation of grade points**	100% final assessment in the form of an exam (30%) and a current assessment (70%). 30% exam: semester exam, according to the schedule of the educational process 70% current rating: * 40% evaluation of tasks in laboratory work; * 30% intermediate control (2 modular Control works)
	82-89	B	good		
	74-81	C			
	64-73	D	satisfactory		
	60-63	E			
	35-59	FX	Unsatisfactory (with the exam retake option)		
	0-34Students are required to attend classes according to the schedule and adhere to the ethics of behavior. In case of absence, students will need to complete all tasks to compensate for missed classes. Participation in practical classes requires preliminary preparation and early processing of all the	F	Unsatisfactory (with mandatory repetition of the course)		

necessary materials for productive discussions during the lesson. Written assignments must be submitted before the deadline.					

Course policy	Students are required to attend classes according to the schedule and adhere to the ethics of behavior. In case of absence, students will need to complete all tasks to compensate for missed classes. Participation in practical classes requires preliminary preparation and early processing of all the necessary materials for productive discussions during the lesson. Written assignments must be submitted before the deadline.

COURSE STRUCTURE AND CONTENT

Topic 1	Basic concepts of software design technology. Information Systems Software lifecycle.	**Laboratory work 1**	Identify high-level requirements. Development of the vision document.	**Self-study**	Identify high-level requirements. Development of the "vision" document according to individual tasks.
Topic 2	Organization of development for Information Systems Software	**Laboratory work 2**	Identifying user requirements. Search for actors and use cases		Identifying user requirements. Search for actors and use cases. Formation and submission of relevant documentation
Topic 3	Basic concepts of organizational business modeling.	**Laboratory work 3**	Description of use cases (use-Case, UC)		Creating documents describing use cases (use-Case, UC)
Topic 4	Specification of functional requirements for Information Systems Software	**Laboratory work 4**	Developing a glossary		Developing a glossary for the selected subject area
Topic 5	Subject area modeling methodologies	**Laboratory work 5**	Detailed description of key UCS		Creating documents with a description of key UCS
Topic 6	Information support of Information Systems	**Laboratory work 6**	Analysis and specification of special requirements		Analysis and specification of special requirements
Topic 7	Modeling of information support	**Laboratory work 7**	SRS formation		Generating SRS for the selected subject area

Topic 8	Object-Oriented Analysis and design of Information Systems Software	**Laboratory work 8**	SRS verification		SRS verification for the selected subject area

RECOMMENDED READING

Compulsory

1. Wiggers, Carl I., Joy Beatty. (2016). Development of software requirements. BHV-Petersburg. Retrieved from http://www.twirpx.com/file/1073169/
2. Tabunshchyk, G. V., Kaplienko, T. I., Petrova, O. A. (2016). Design and modeling of software for modern information systems. Zaporozhye: Wild Field.
3. Introduction to Software Engineering and Software Lifecycle Management Guide to Software Engineering Base of Knowledge (SWEBOK). Retrieved from access mode: sorlik.blogspot.com/
4. Tabunshchyk, G. V., Kaplienko, T. I., Petrova, O. A. (2016). Design and modeling of software for modern information systems. Teaching. Manual. Zaporozhye.
5. Shishkov, B. (2020). Designing Enterprise Information Systems: Merging Enterprise Modeling And Software Specification. New York: Springer.
6. Dwyer Barry. (2016). Systems Analysis and Synthesis: Bridging Computer Science and Information Technology. Morgan Kaufmann.

Recommended

7. Elizabeth Hull, Ken Jackson, Dick Jeremy. (2017). Requirements Engineering. DMK Press.
8. I Jacobson I., Lawson H. Bud, Ng P.-W., McMahon P, E., Goedicke M. (2019). The Essentials of Modern Software Engineering: Free the Practices from the Method Prisons. Association for Computing Machinery and Morgan & Claypool Publishers.
9. Bubnov, A., Bubnov, S., Maikov, K. (2018). Development and analysis of software requirements. Course.
10. Mejia, J., Muñoz, M., Rocha, A., Quiñonez, Y. (Ed.). (2021). New Perspectives in Software Engineering: Proceedings of the 9th International Conference on Software Process Improvement (CIMPS 2020).
11. Rosen, C. (2020). Guide to Software Systems Development: Connecting Novel Theory and Current Practice. New York: Springer.
12. Standard for Software Verification and Validation Plans (ANSI / IEEE standard 1012-1986).
13. D'Andrade Brian. (2021). Software Engineering: Artificial Intelligence, Compliance, and Security. Nova Science Publishers.

Academic integrity

Graduate students are expected to adhere to the Code of Ethics of Academic Relations and Integrity" of NTU "KhPI".

The content of this syllabus is consistent with the course program.