Національний технічний університет України
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Department of computer systems software

COMPONENTS OF SOFTWARE ENGINEERING. PART 3. SOFTWARE ARCHITECTURE

Syllabus

Requisites of the Course

Cycle of Higher Education	***First cycle of higher education (Bachelor's degree)***
Field of Study	*12 Information Technologies*
Specialty	*121 Software engineering*
Education Program	*Software Engineering of Multimedia and Information Retrieval Systems*
Type of Course	*Normative*
Mode of Studies	*full-time*
Year of studies, semester	*2nd year, 4th semester*
ECTS workload	*36 hours for lectures, 18 hours for laboratory work, 96 hours for self-study.*
Testing and assessment	*Exam, modular control work, calendar control*
Course Schedule	*According to http://roz.kpi.ua/*
Language of Instruction	*English*
Course Instructors	Inna Saiapina, PhD, Assoc. Prof., saiapina@pzks.fpm.kpi.ua
Access to the course	Google classroom. To be provided to registered students.

Outline of the Course

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

The study of the "Components of software engineering. Part 3. Software architecture" allows students to develop the competencies necessary for solving practical tasks of professional activities related to the design, modeling and development of software for multimedia and information-search systems.

The purpose *to study the course "Components of software engineering. Part 3. Software architecture" is the formation of students' understanding of the basic principles and mechanisms and the ability to design, model and program implementation of multimedia and information-search systems and applications working independently or in a team.*

The subject *of the "Components of software engineering. Part 3. Software architecture" course are methods, methodologies and approaches to building a software architecture, its component modules and relationships between them.*

The study of the "Components of software engineering. Part 3. Software architecture» course contributes for students to the formation of general (GC) and professional competences (PC), necessary for solving practical problems of professional activity related to modeling, designing and software development:
PC01 *Ability to identify, classify and formulate software requirements.*
PC02 *Ability to participate in software design, including its structure, behavior and functioning processes modeling (formal description).*
PC03 *Ability to develop software systems architectures, modules and components.*
PC04 *Ability to formulate and ensure software quality requirements in accordance with customer requirements, specifications and standards.*
PC05*. Ability to follow specifications, standards, rules and recommendations in the professional field during the life cycle processes implementation.*

PC07 *Knowledge of information data models, the ability to create software for data storage, retrieval and processing.*
PC08 *Ability to apply fundamental and interdisciplinary knowledge to successfully solve software engineering problems.*
PC10 *Ability to accumulate, process and systematize professional knowledge about software creation and maintenance, and determination of the importance of lifelong learning.*
PC11*. Ability to implement phases and iterations of the life cycle of the software systems and information technology based on appropriate models and approaches to software development.*
PC12 *Ability to carry out the system integration process, apply change management standards and procedures to maintain software integrity, overall functionality and reliability.*
PC13 *Ability to reasonably select and master software development and maintenance tools.*
PC21*. Ability to identify, analyze and document software requirements for multimedia and information retrieval systems*
PC22*. Ability to create innovative startup projects, calculate basic technical and economic indicators and develop business models of multimedia software and information retrieval systems innovative startup projects that have commercial potential for investment.*
Studying the course "Components of software engineering. Part 3. Software architecture" contributes to students' formation of the following program learning outcomes (PLO) according to the educational program:
PLO01 *To analyze, purposefully search and select the necessary information and reference resources and knowledge to solve professional problems, taking into account modern advances in science and technology.*
PLO02 *To know the professional ethics code, understand the social significance and cultural aspects of software engineering and adhere to them in professional activities.*
PLO03 *To know the software life cycle basic processes, phases and iterations.*
PLO04 *To know and apply professional standards and other regulatory documents in the field of software engineering.*
PLO06 *To know and apply relevant mathematical concepts, domain methods, system and object-oriented analysis and mathematical modeling for software development.*
PLO07 *To know and to apply in practice the fundamental concepts, paradigms and basic principles of the functioning of language, instrumental and computational tools of software engineering.*
PLO08 *To know and to be able to develop a human-machine interface.*
PLO09 *To be able to use collecting, formulating and analyzing software requirements methods and tools.*
PLO10 *To conduct a pre-project survey of the subject area, system analysis of the design object.*
PLO11 *To select initial data for design, guided by formal methods of describing requirements and modeling.*
PLO13 *To know and apply methods of developing algorithms, designing software and data and knowledge structures.*
PLO14 *To apply in practice instrumental software tools for domain analysis, design, testing, visualization, measurement and documentation of software.*
PLO15 *To choose programming languages and development technologies to solve the problems of creating and maintaining software.*
PLO16 *To have the software development, design approval and all types of software documentation release skills.*
PLO17 *To be able to apply methods of component software development.*
PLO18 *To know and be able to apply information technology of processing, storage and transmission of data.*
PLO19 *To know and be able to apply software verification and validation methods.*
PLO20 *To know approaches to evaluation and quality assurance of software.*
PLO23 *To be able to document and present the software development results.*
PLO31 *To be able to identify, analyze and document software requirements for multimedia and information retrieval systems*

PLO32 *To be able to develop and analyze full cycle models for multimedia and information retrieval systems software creation.*
PLO33 *To be able to organize a software product management complete cycle.*
PLO34 *To be able to create innovative startup projects of designing multimedia and information-search systems software that have commercial potential for investment.*
PLO35 *To be able to develop and analyze business models of innovative startup projects of developing multimedia and information retrieval systems software that have commercial potential for investment.*
PLO36 *To be able to manage the creation and implementation of software projects in accordance with international standards.*
PLO38 *To be able to apply programming technologies for multimedia and information retrieval systems software development.*

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

To the successful study of the course "Components of software engineering. Part 3. Software architecture" precedes the study of the course "Components of Software Engineering" of the bachelor's study curriculum in the specialty 121 Software engineering.

Theoretical knowledge and practical skills, received during the study of the course "Components of software engineering. Part 3. Software architecture" contribute to the assimilation of material from the courses "Components of software engineering. Part 4. Software quality and testing", "Components of software engineering. Course work", "Software security", "Bachelor Thesis" and "Pre-diploma practice" of the bachelor's study curriculum in the specialty 121 Software engineering.

3. **Content of the course**

Course "Components of software engineering. Part 3. Software architecture" involves the study of the following topics:

Topic 1. Introduction. Basic definitions and concepts.

Topic 2. Architecture planning and design.

Topic 3. Object analysis and modeling.

Topic 4. Design patterns.

Topic 5. Architectural styles and patterns.

Topic 6. Applied and theoretical methods of programming.

Topic 7. Software verification and validation.

Modular control work.

Exam

4. **Coursebooks and teaching resources**

Main literature:

1. *Educational and methodological materials on the subject "Components of software engineering. Part 3. Software architecture".*
   Use to master the practical skills of the discipline. The materials are in Google classroom; access is to be provided to registered students.

Additional literature:

2. *L. Bass, P. Clements, R. Kazman. Software Architecture in Practice. Addison-Wesley, 2021. 497 p.*
3. *M. Richards, N. Ford. Fundamentals of Software Architecture, O'Reilly Media, 2021.*
4. *M. Richards. Software Architecture Patterns. O'Reilly Media, 2015. 55 p.*

5. R. Martin. Clean Architecture: A Craftsman's Guide to Software Structure and Design. Pearson, 2017. 432 p.
6. B. Rumpe. Agile Modeling with UML. Springer, 2017, 388 p. DOI 10.1007/978-3-319-58862-9
7. E. Freeman, E. Robson. Head First Design Patterns: Building Extensible and Maintainable Object-Oriented Software. O'Reilly Media, 2021, 669 p.

Educational content

5. Methodology for mastering the course (educational component)

No	Type of training session	Description of the training session
Topic 1. Introduction. Basic definitions and concepts		
1	Lecture 1. Introduction. Basic definitions and concepts. (2 ac.h.)	Introduction. Information on the educational process organization. Useful resources. Academic integrity. Evaluation system. What this course will teach you. Life cycle of software development. The concept of software architecture. Tasks on self-study: item 6, number 1.
Topic 2. Architecture planning and design		
2	Lecture 2. Life cycle standards and models (2 ac.h.)	Cascade model. Incremental model. Spiral model. Evolutionary model. Basic software development methodologies. Agile SCRUM. Extreme Programming, SAFE, Kanban. Tasks on self-study: item 6, number 2.
3	Laboratory work 1. (2 ac.h.)	Analysis and formation of software requirements Assignment on self-study: item 6, number 3.
4	Lecture 3. Architecture and system quality attributes (2 ac.h.)	Quality attribute scenarios. Readiness. Modifiability, Productivity, Security. Controlability. Assignment to self-study: item 6, number 4.
Topic 3. Object analysis and modeling		
5	Lecture 4. Object-oriented analysis and design (2 ac.h.)	Object-oriented approach and basic principles. Compromises in requirements and design. Advantages of object-oriented architecture. Tasks on self-study: item 6, number 5.
6	Laboratory work 2. (2 ac.h.)	Fundamentals of software modeling. Tasks on self-study: item 6, number 6
7	Lecture 5. Object-oriented modeling (2 ac.h.)	UML class diagrams. Specialized UML diagrams. Concepts and elements of DDD. Cohesion as an internal characteristic of the module. Coupling as an external characteristic of the module. Tasks on self-study: item 6, number 7.
Topic 4. Design patterns		
8	Lecture 6. Design patterns (4 ac.h.)	The concept of pattern design. Types of patterns. Generating patterns. Structural patterns. Behavioral patterns. Tasks on self-study: item 6, number 8.
9	Laboratory work 3. (2 ac.h.)	Modeling of the system behavior at the logical level (development of the state diagram) Tasks on self-study: item 6, number 9.
Topic 5. Architectural styles and patterns		
10	Lecture 7. Architectural patterns	Types of architectural patterns. Multilayer architecture. Event-driven architecture. Microkernel architecture.

	databases (4 ac.h.)	*Architecture of microservices and others. Advantages, disadvantages and application examples.* *Tasks on self-study: item 6, number 10.*
11	*Laboratory work 4. (4 ac.h.)*	*A study of software design architectural patterns.* *Tasks on self-study: item 6, number 11*
12	*Lecture 8. Monolithic architecture (2 ac.h.)*	*Monolithic and distributed architecture. Big Ball of Mud. Misconceptions in software development.* *Tasks on self-study: item 6, number 12.*
13	*Lecture 9. Multi-layer software architecture (2 ac.h.)*	*Logical structure of multilayer architecture. The concept of levels and layers. Multi-layered architecture design patterns. Designing layers.* *Assignment on self-study: p. 6, No. 13.*
14	*Lecture 10. Service-oriented architecture (2 ac.h.)*	*Web services, their composition. REST architecture, design principles.* *Tasks on self-study: item 6, number 14.*
15	*Laboratory work 5. (4 ac.h.)*	*Modeling of the system behavior at the logical level (development of sequence and cooperation diagrams)* *Tasks on self-study: item 6, number 15*
16	*Lecture 11. Microkernel and Pipeline architecture (2 ac.h.)*	*Characteristics, basic principles, examples of use.* *Tasks on self-study: item 6, number 16*
17	*Lecture 12. Event-driven architecture and space-based architecture (2 ac.h)*	*Characteristics, basic principles, examples of use.* *Tasks on self-study: item 6, number 17*
Topic 6. Applied and theoretical methods of programming		
18	*Lecture 13. Applied programming (2 ac.h.)*	*The main types of application programming and their features. Principles of DRY, KISS, YAGNI. SOLID.* *Tasks on self-study: Item 6, No. 18.*
29	*Lecture 14. Theoretical programming (2 ac.h.)*	*The main types of theoretical programming and their features.* *Tasks on self-study: Item 6, No. 19*
20	*Laboratory work 6. (4 ac.h.)*	*Development of the physical representation model of an informational system (development of a diagram of components and deployment)* *Tasks on self-study: Item 6, No. 20*
Topic 7. Software verification and validation		
21	*Lecture 15. Software verification and validation (2 ac.h.)*	*Approach to requirements scenario validation. Verification of object models and composition of components.* *Tasks on self-study: item 6, number 21.*
22	*Modular control work (2 acc. hours)* *Tasks on self-study: item 6, number 22*	

6. Self-study work of a student

The course "Components of software engineering. Part 3. Software architecture " is based on self-study preparations for classroom classes on theoretical and practical topics.

No	*The name of the topic submitted for independent processing*	*Number of hours*	*literature*

1	*Preparation for the lecture 1*	*2*	*1-7*
2	*Preparation for lecture 2*	*2*	*1, 6*
3	*Preparation for laboratory work 1*	*5*	*1, 2*
4	*Preparation for the lecture 3*	*2*	*2*
5	*Preparation for the lecture 4*	*2*	*1,*
6	*Preparation for laboratory work 2*	*5*	*1-3, 5, 6*
7	*Preparation for the lecture 5*	*2*	*1-3, 5*
8	*Preparation for the lecture 6*	*2*	*1, 7*
9	*Preparation for laboratory work 3*	*5*	*1-3, 5, 6*
10	*Preparation for the lecture 7*	*2*	*1, 3 – 5*
11	*Preparation of laboratory work 4*	*5*	*1, 3 – 5*
12	*Preparation for the lecture 8*	*2*	*1, 3-5*
13	*Preparation for the lecture 9*	*2*	*1, 3-5*
14	*Preparation for lecture 10*	*2*	*1, 3, 4*
15	*Preparation of laboratory work 5*	*5*	*1-8, 10, 11*
16	*Preparation for lecture 11*	*2*	*1, 8, 9*
17	*Preparation for lecture 12*	*2*	*1, 8, 9*
18	*Preparation for lecture 13*	*2*	*1, 4*
19	*Preparation for lecture 14*	*2*	*1, 4*
20	*Preparation of laboratory work 6*	*5*	*1, 4-6, 7, 10, 12*
21	*Preparation for modular control work*	*8*	*1-12*
22	*Preparation for the exam*	*30*	*1-12*

Policy and Assessment

7. Course policy

- *Attending lectures is mandatory.*
- *Attending laboratory classes can be occasional and as needed to defend laboratory work.*
- *Rules of behavior in classes: activity, respect to the others, turning off phones.*
- *Adherence to the policy of academic integrity.*
- *Rules for the laboratory work defence: the work must be done according to the student's variant, which is determined by his number in the group list, or the topic and subject area approved by the teacher.*
- *The rules for assigning bonus and penalty points are as follows.*

Bonus points are awarded for:
- activity in lectures and laboratory classes. The maximum number of points for all classes is 5 points.
Penalty points are calculated for:
- plagiarism The performed laboratory work does not correspond to the task option, the identity of laboratory work reports among different works (number of points: 5 points).

8. Monitoring and grading policy

During the semester, students perform 6 laboratory works. The maximum number of points for laboratory work: 5 points.

Points are awarded for:
- quality of laboratory work (report): 0-2 points;
- survey (test) during the defense of laboratory work: 0-2 points;
- timely submission of work for defense: 0-1 point.

Criteria for evaluating the quality of laboratory work (report):
2 points – the work is done qualitatively, in full;
0-1 point – the work is incomplete or contains errors.

Evaluation criteria for the survey on the protection of laboratory work:
2 points – the answer is complete, well-argued;
1 point – there are errors in the answer;
0 points - there is no answer or the answer is incorrect.

If the laboratory work is re-submitted for examination, the total maximum grade for the laboratory work is reduced by 1 point.

The maximum number of points for performing and defending laboratory work:
R_L= 6 laboratory works × 5 points = 30 points.

The task for the modular control work consists of 14 test questions - 8 questions with one correct answer and 6 questions with several correct answers. Each question with one correct answer is valued at 1 point, each question with several correct answers are valued at 2 points.
Evaluation criteria for each test question with one correct answer:
1 point – the answer is correct;
0 points - there is no answer or the answer is incorrect.

Evaluation criteria for each multiple-choice test question:
2 points – all correct answers and no incorrect answers are selected;
1 point – at least 50% of all correct answers are chosen;
0 points – no answer or all answers are incorrect.

The maximum number of points for a modular control work:
R_{MKR} = 1 point × 8 test questions with one correct answer + 2 points × 6 questions with several correct answers = 20 points.
The rating scale for the course is equal to:
$R = R_c + R_{exam} = R_L + R_{MKR} + R_{exam}$ = 30 points + 20 points + 50 points = 100 points.

Calendar control: is conducted twice a semester as a monitoring of the current state of fulfillment of the syllabus requirements.
At the first certification (8th week), the student receives "credited" if his current rating is at least 8 points (50% of the maximum number of points a student can receive before the first certification).
At the second certification (14th week), the student receives "passed" if his current rating is at least 15 points (50% of the maximum number of points a student can receive before the second certification).

Semester control: exam
Conditions for admission to semester control:
With a semester rating (R_c) of at least 30 points and the enrollment of all laboratory work, the student is admitted to the exam. After passing the exam, a grade is assigned according to the table (Table of correspondence of rating points to grades on the university scale).

A necessary condition for admission to the exam is the performance and defense of laboratory work.

The exam work contains 3 questions: 2 theoretical and 1 practical-oriented. The answer to each theory question is worth 15 points, and the answer to a practical question is worth 20 points.

Evaluation criteria for each theoretical question:
13-15 points – the answer is correct, complete, well-argued;
9-12 points – the answer is correct, detailed, but the reasoning is incomplete;
5-8 points - in general, the answer is correct, but there are errors;
1-4 points – there are significant errors in the answer;
0 points - there is no answer or the answer is incorrect.

Criteria for evaluating a practically-oriented question:

18-20 points – the task was completed correctly, a full thorough explanation of the chosen solutions for the task was provided;
14-17 points – tasks and explanations of the chosen solutions are completed at a basic level, but not all features are taken into account;
9-13 points – the performance of the task and/or the reasoning of the chosen decisions contains a number of inaccuracies or the reasoning of the decisions is absent;
5-8 points – when completing the task, significant mistakes were made, leading to a false result;
1-4 points – the task has been completed, but the correct answers have not been received;
0 points - there is no answer or the answer is incorrect.

Table of correspondence of rating points to grades on the university scale:

Scores	*Rating*
100-95	Perfectly
94-85	Very good
84-75	Fine
74-65	Satisfactorily
64-60	Enough
Less than 60	Unsatisfactorily
Admission conditions not met	Not allowed

Syllabus of the course
Is designed by Inna Saiapina, Ph.D., Assoc. Prof.
Adopted by Computer Systems Software Department (protocol № 12 from 26.04.23)

Approved by the Faculty Board of Methodology (protocol № 10 from 26.05.23)