Syllabus
Course Program

Software architecture and design

Specialty
121 – Software Engineering

Institute
Institute of Computer Science and Information Technology

Educational program
Software Engineering

Department
Software Engineering and Management Intelligent Technologies (321)

Level of education
Bachelor's level

Course type
Special (professional), Mandatory

Semester
5

Language of instruction
English, Ukrainian

Lecturers and course developers

Dmytro Dvukhhlavov

dmytro.dvukhhlavov@khpi.edu.ua
Ph.D., Associate Professor, Associate Professor of Software Engineering and Management Intelligent Technology Department.

Google Scholar: https://scholar.google.com/citations?user=OAzyFg8AAAAJ&hl
ORCID: https://orcid.org/0000-0002-3361-3212
Scopus: https://www2.scopus.com/authid/detail.uri?authorId=57211294555
Web of Science: https://www.webofscience.com/wos/author/record/E-8279-2019).
More about the lecturer on the department's website

Alyona Dvukhhlavova

Alyona.Dvukhhlavova@khpi.edu.ua
Senior Lecturer of Software Engineering and Management Intelligent Technology Department.

Google Scholar: https://scholar.google.com/citations?user=cSAk_9wAAAAJ
ORCID ID: https://orcid.org/0000-0002-0111-3010
More about the lecturer on the department's website

General information

Summary

Teaching the discipline ensures the readiness of the students to participate in addressing the issues of justification of the choice of software architecture, determining the composition and structure of its components, taking into account the established quality requirements. During the study of the discipline, a significant amount of time is devoted to the study of design patterns.
Teaching the discipline provides students with an understanding of the content of the software development process in IT companies, as well as the development of basic skills in software design and collaboration with the project code required to work in a team of software developers.

Course objectives and goals

The course objective to deepen knowledge about the software life cycle and standards governing the implementation of its stages, providing knowledge about software architecture, architectural styles of modern software, known software design templates, as well as frameworks describing software architecture, as well as developing skills in developing software solutions using some design templates, as well as developing skills in creating software models based on design templates in Enterprise Architect.

The course aims to deepen the knowledge system to deepen knowledge about the software life cycle and standards governing the implementation of its stages, the principles of organizing software development at the present stage of development of the IT industry, software development technologies, design strategies and methodologies and software development, on the presentation of software design results, on the principles of user interface design, as well as the development of software development skills based on a defined list of software requirements and skills of git version control system and github repository during software development. Mastering the discipline is an important element for the making diploma work.

Format of classes

Lectures, laboratory classes, self-study, consultations. Final control – an exam in each of the semesters.

Competencies

К01. Ability to think abstractly, analyze and synthesize.
К02. Ability to apply knowledge in practical situations.
К05. Ability to learn and master modern knowledge.
К06. Ability to search, process and analyze information from various sources.
К07. Ability to work in a team.
К13. Ability to identify, classify and formulate software requirements.
К14. Ability to participate in software design, including modeling (formal description) of its structure, behavior and processes of functioning.
К15. Ability to develop architectures, modules and components of software systems.
К17. Ability to comply with specifications, standards, rules and guidelines in the professional field when implementing life cycle processes.
К19. Knowledge of data information models, ability to create software for storing, extracting and processing data.
К23. Ability to implement phases and iterations of the life cycle of software systems and information technologies based on appropriate software development models and approaches.
К24. Ability to carry out the system integration process, apply change management standards and procedures to maintain the integrity, overall functionality and reliability of the software.
К25. Ability to reasonably choose and master the tools for software development and maintenance.
К26. Ability to think algorithmically and logically.

Learning outcomes

PLO01. Analyze, purposefully search and select information and reference resources and knowledge necessary for solving professional problems, taking into account modern achievements of science and technology.
PLO03. Know the basic processes, phases and iterations of the software life cycle.
PLO06. Ability to select and use a software development methodology appropriate to the task.
PLO07. To know and apply in practice the fundamental concepts, paradigms and basic principles of functioning of language, tools and computing tools of software engineering.
PLO10. Conduct a pre-project survey of the subject area, system analysis of the design object.
PLO11. Select input data for design, guided by formal methods of requirements description and modeling.
PLO12. Apply effective approaches to software design in practice.
PLO13. Know and apply methods of developing algorithms, designing software and data structures and knowledge.
PLO14. Apply in practice software tools for domain analysis, design, testing, visualization, measurement

and documentation of software.
PLO15. Motivated to choose programming languages and development technologies to solve the problems of creating and maintaining software.
PLO16. Have the skills of team development, coordination, design and production of all types of program documentation.
PLO17. Be able to apply methods of component software development.
PLO18. To know and be able to apply information technologies for data processing, storage and transmission.
PLO19. To know and be able to apply methods of software verification and validation.
PLO20. Know approaches to assessing and ensuring software quality
PLO23. Be able to document and present the results of software development.

Student workload

The total volume of the course is:
in 5 semester – 120 hours (4 ECTS credits): lectures – 16 hours, laboratory classes – 32 hours, self-study – 72 hours;
in 6 semester – 120 hours (4 ECTS credits): lectures – 32 hours, laboratory classes – 32 hours, self-study – 56 hours.

Course prerequisites

Data Models and Structures
Object-Oriented Programming

Features of the course, teaching and learning methods, and technologies

Teaching and learning methods
The main method of teaching during lectures is the explanatory-illustrative method. To intensify cognitive activity, students' speeches and organization of discussions on certain issues of lectures are provided. Execution of laboratory works involves the creation by students of a solution model in the form of a set of UML diagrams and program code for the implementation of typical program functionality based on known design templates. For the implementation of the student determines his own subject area. There are no ready-made algorithms for solving, which encourages the manifestation of creative activity of students.

Forms of assessment
Assimilation of the theory is tested in the form of a rapid survey during lectures (CAS), a survey or automated testing at the beginning of laboratory work (CAS).
Control of mastering the material for self-study involves the preparation and defense of abstracts on individual topics (2 abstracts) (CAS).
The practical skills test is tested on individual case studies (CAS). Final control is carried out at the defense of the course project and during semester exams (FAS). At the exam in the 5th semester, testing on theoretical questions is expected. The exam in the 6th semester involves testing theoretical questions and creating a model of a small business process and a behavioral UML model according to an individual task in a limited time.

Program of the course

Topics of the lectures

5 semester
Topic 1.
The concept of "software architecture". The main provisions of the IEEE 1471 standard and other regulatory documents of software engineering. Frameworks for describing software architecture (software) ("4 + 1", RM-ODP, SOMF).
Topic 2.
Requirements for modern software architecture. The essence of using templates to ensure modern software requirements. Classification of design patterns (Design pattern by GoF). Typical description of

design templates. Detailed description of popular design templates GRASP responsibility allocation templates.
Topic 3.
Architectural styles. Definition of architectural style. An overview of modern architectural styles.
Topic 4.
Programming paradigms.

6 semester
Topic 5.
A modern point of view at the software development process. SCRUM framework: purpose and principles of application during development. Contents CD / CI (continuous integration / continuous delivery). Software to automate software development.
Topic 6.
Design as a component of the software development life cycle. Basis of definition and regulations. Notations and presentation of design results.
Topic 7.
UML as a notation to represent design results. Classification of UML diagrams. Use UML diagrams to describe the behavior and structure of software. UML charting software.
Topic 8.
Strategies and methodologies for software development.
Topic 9.
User interface design.

Topics of the workshops

Workshops are not provided within the discipline.

Topics of the laboratory classes

5 semester
Topic 1: Study of the Enterprise Architect installation process and its application to create basic UML diagrams
Topic 2. Research of design pattern BRIDGE implementation features
Topic 3. Research of design pattern COMPOSITE implementation features
Topic 4. Research of design pattern COMMAND implementation features
Topic 5. Research of design patterns SINGLETONE and PROXY implementation features
Topic 6. Research of design pattern FACTORY METHOD implementation features
Topic 7. Research of design pattern TEMPLATE METHOD implementation features

6 semester
Topic 8: Installing and configuring the Git system. Work in Git-Bash
Topic 9. Branch management using Git
Topic 10. Using the GitHub repository
Topic 11. Formalized representation of business processes and software requirements for the use of CASE-system
Topic 12. Designing system behavior for different categories of users based on UML diagrams
Topic 13. Design of the scheme for data storage
Topic 14. Designing the structure of software components and its deployment scheme

Self-study

5th semester
Additional materials (videos, articles) for independent study by students are recommended. Their topics presented below.
Installing Enterprise Architect. Creating projects and models in Enterprise Architect. Enterprise Architect toolbars. Create class, sequence, and activity diagrams in Enterprise Architect. Assignment of design templates. Typical situations, advantages and disadvantages of using design templates. Principles of

application of GRASP responsibilities distribution templates. Overview of implementations of the application of responsibilities distribution templates. Detailed description of architectural styles. Analysis of the implementation of style in well-known software products. Modern views on approaches to software design.

6th semester
Students execute a course project, during which the questions presented below are worked out. Choice of the task for automation and definition of object of automation. Description of business processes of the a object of automation. Determination of functional and non-functional requirements for the designed software. Development of SRS for the software being created. Designing a domain model of subject area objects. Development of use case diagrams and their refinement by developing scenarios or interaction diagrams. Entity-Relation diagram design. Reasonable choice of database management system. Reasonable choice of software architecture. Development and description of the structure of software components and its deployment scheme. Development of components and deployment diagram.

Course materials and recommended reading

Key literature

1. Sommerville Ian. Software Engineering (6th Edition) / https://www.academia.edu/6826193/Ian_Sommerville_Software_Engineering_6th_Edition.
2. Freeman Eric, Robson Elisabeth. Head First Design Patterns: Building Extensible and Maintainable Object-Oriented Software 2nd Edition. – O'Reilly Media, 2020. – 672 p.
3. Larman Craig. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, 3rd Edition. – Pearson, 2005. – 736 p.
4. Git - Documentation (git-scm.com) // https://git-scm.com/doc.
5. Грицюк Ю. І. Аналіз вимог до програмного забезпечення. Навчальний посібник. 2018. – 456 с.
6. Піхлер Р. Agile продукт-менеджмент за допомогою Scrum. Фабула, 2019. – 128 с.
7. Мартін Роберт. Чистий код. – Фабула, 2019. – 416 с.

Additional literature

Standards
6. ISO/IEC/IEEE 12207 Systems and software engineering — Software life cycle processes.
7. IEEE 610-1990. IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries.
8. ISO 25010. Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models.
9. IEEE 1471. Recommended Practice for Architectural Description of Software-Intensive Systems.
10. ISO/IEC/IEEE 42010. Systems and software engineering — Architecture description.
11. IEEE/ISO/IEC 29148-2018. Systems and software engineering – Life cycle processes – Requirements engineering.
12. IEEE 1016. IEEE Recommended Practice for Software Design Descriptions.

Internet resources
13. https://refactoring.guru/ua/design-patterns.
14. Dr. Winston W. Rovce. Managing the development of large software systems // http://www-scf.usc.edu/~csci201/lectures/Lecture11/royce1970.pdf.
15. Microsoft Solutions Framework (MSF). Team Model // https://www.microsoft.com/en-us/download/details.aspx?id=3214.
16. Manifesto for Agile Software Development // http://agilemanifesto.org/
17. UML modeling tools for Business, Software, Systems and Architecture // https://www.sparxsystems.com/
18. https://www.smart-it.com/ru/2021/08/12-best-software-development-methodologies-with-pros-and-cons/
19. https://refactoring.guru/ua/design-patterns.

20. https://sparxsystems.com

Assessment and grading

Criteria for assessment of student performance, and the final score structure

The grade for the discipline consists of the points obtained during the study during the semester and the points awarded for passing the exam.
During the 5 semester, a student can receive up to 70 points for:
- assimilation of theory (topics of independent work) (up to 30 points);
- performance of the 1st laboratory work (4 points);
- performance of 2-7 laboratory works (6 points for each).
A student can get up to 30 points for passing the exam.
During the 6 semester, a student can receive up to 80 points for:
- assimilation of theory (up to 15 points);
- performance of LW#1-3 (Git Practice) (up to 15 points);
- performance of of LW#4-7 (Practice in design SW) and passing course project (up to 50 points).
A student can get up to 20 points for passing the exam.

Grading scale

Total points	National	ECTS
90–100	Excellent	A
82–89	Good	B
75–81	Good	C
64–74	Satisfactory	D
60–63	Satisfactory	E
35–59	Unsatisfactory (requires additional learning)	FX
1–34	Unsatisfactory (requires repetition of the course)	F

Norms of academic integrity and course policy

The student must adhere to the Code of Ethics of Academic Relations and Integrity of NTU "KhPI": to demonstrate discipline, good manners, kindness, honesty, and responsibility. Conflict situations should be openly discussed in academic groups with a lecturer, and if it is impossible to resolve the conflict, they should be brought to the attention of the Institute's management.
Regulatory and legal documents related to the implementation of the principles of academic integrity at NTU "KhPI" are available on the website: http://blogs.kpi.kharkov.ua/v2/nv/akademichna-dobrochesnist/

Approval

Approved by 11.04.2023

Head of the department
Ihor HAMAIUN

Guarantor of the educational program
Uliya LITVINOVA