SIC ISO 9001 CERTIFIED
SLC. MS.040. LSO9001. 1693

IEU INTERNATIONAL EUROPEAN UNIVERSITY

SYLLABUS

INTERNATIONAL EUROPEAN UNIVERSITY

BUSINESS ARS EST

IEU INTERNATIONAL EUROPEAN UNIVERSITY

EUROPEAN SCHOOL OF BUSINESS

Software modeling and analysis

2024

SYLLABUS

Discipline

Software modeling and analysis

Lecturer (s)

Borys Shevchuk, Associate Professor at the Department of Information Technology, PhD in Pedagogy, Associate Professor

Discipline page

https://business.ieu.edu.ua/kafedry/kafedra-informatsiinykh-tekhnolohii#zzz-063

Consultations

Online: Monday 3:10 p.m. – 4:30 p.m; Friday 3:10 p.m. – 4:30 p.m.

Offline -

Contact number

(093) 806 84 58

E-mail

borys_shevchuk@ieu.edu.ua

Discipline page

https://dist.ieu.edu.ua/course/view.php?id=635

Form of final control	Test	Fail test	Exam
	□	□	☒

SYLLABUS

1 Brief discipline annotation

The Software modeling and analysis academic discipline is designed for Bachelors of the knowledge area: 12 Information Technology, specialty: 121 Software Engineering. It is one of the professional disciplines for future software developers.

2 Background for studying discipline

The program is based on the knowledge acquired while studying such disciplines as Software architecture and design, Fundamentals of programming, Object-oriented programming.

3 Goal and objectives of the discipline

The **goal** of the Software modeling and analysis discipline is to study methods of object-oriented analysis and UML graphical tools and to acquire practical skills in the purpose and construction of the main types of canonical diagrams.
Key objectives of the discipline: to provide students with competencies in object-oriented analysis and modeling in the UML visual language in order to take part in the collaborative development, maintenance, design and documentation of object-oriented software systems. This course aims to teach students how to use basic UML diagrams and apply this knowledge at all stages of conceptual, logical and physical design of software application architecture using CASE tools,

4 Learning outcomes

PLO 6. Ability to select and use the appropriate software development methodology.
PLO 11. To select design input data using formal methods of requirements description and modeling.
PLO 17. To be able to apply component-based software development techniques.
After learning the discipline, students should
know:

- content of basic concepts, subject matter and methods of the course;
- basic design patterns;
- the history of the creation and development of the Unified Modeling Language (UML);
- capabilities, advantages and disadvantages of various object-oriented analysis techniques and graphical notations;
- the appearance and purpose of the main graphical tools of the UML language, the main types of canonical diagrams;
- how to analyze the subject area and describe it using the UML language;
- the main stages, principles and styles of constructing complex software systems;
- software life cycle, stages in different life cycle models;

be able to:

- analyze the requirements for the developed software;
- estimate labor intensity and choose adequate approaches to software development;
- design software architecture using visual modeling tools;
- prepare program documentation;
- apply tools for object-oriented analysis and graphical representation in UML (IBM Rational Rose);
- design components of an architectural solution;
- formulate software requirements;
- analyze software architecture for compliance with quality attributes.

5 Credits

4 credits

6 Discipline structure

No.	Modules and topics	Amount of hours (full-time mode of study)				
		Total	including			
			Lectures	*Practical classes*	*Laboratory classes*	*Independent work*
MODULE I. THEORETICAL AND PRACTICAL FOUNDATIONS OF MODELING AND ANALYSIS						
Content module I. Introduction to the Software modeling and analysis discipline						
1.	*Topic 1.* Introduction to the Software modeling and analysis discipline. General approaches to software modeling	18	2	4	-	12
Total per content module I:		*18*	*2*	*4*	-	*12*
Content module II. Fundamentals of software modeling and analysis						
2.	*Topic 2.* Fundamentals of structural modeling	30	6	8	-	16
3.	*Topic 3. Fundamentals* of software behavior modeling	28	4	8	-	16
4.	*Topic 4.* Fundamentals of event modeling	22	2	6	-	14
5.	*Topic 5.* Fundamentals of software architecture modeling	22	2	6		14
Total per content module II:		*102*	*14*	*28*	-	*60*
Total hours per course:		***120***	***16***	***32***	**-**	***72***

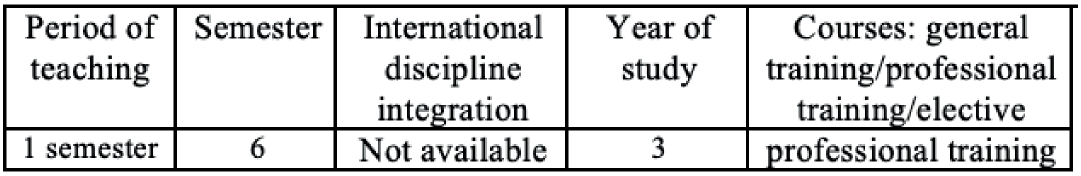
7 List of obligatory tasks

1. Requirements analysis and construction of a target model of the subject area
2. Structural modeling of software. ER methodologies.
3. Structural modeling of software. Bachmann notation.
4. Structural modeling of software. Methodology of the IDEF0 functional modeling
5. Structural modeling of software. Methodology of the DFD functional modeling
6. Construction of use case diagrams (precedent model)
7. Construction of a class diagram in UML. Model of the subject area
8. Construction of interaction diagrams to identify the distribution of responsibilities between objects
9. Construction of state and activity diagrams as a tool for analyzing complex transactions and class operations in system design

8 List of selective tasks

1. Proof of identities with sets. The principle of duality.
2. Problems with countable and uncountable sets.
3. Problems with functional relations.
4. Functions and features of the use of basic logic elements. Switches. Multiplexers. Shift devices. Selector multiplexers. Encoding and decoding devices. Encoders. Decoders.
5. Optimization problems on graphs.
6. Problems on algebraic structures.
7. Formula of inclusions and exclusions. Recurrence relations.

9 Discipline features



Period of teaching	Semester	International discipline integration	Year of study	Courses: general training/professional training/elective
1 semester	6	Not available	3	professional training

10 Hardware and software

Personal computer (PC), Windows OS (7, 8.1, 10, 11), office software packages (Microsoft: Word, Excel Project; Adobe Acrobat), online services.

11 Assessment system and requirements

As part of discipline teaching, one carries out the current and final control of students' knowledge. The final grade is given according to the total rating of students.

The results of the current control of students' knowledge is assessed in general between 0 and 60 points.

Students are admitted to the final control if they fulfil the requirements of the training program and obtain at least 36 points for the current learning activity.

Final assessment of students' knowledge is conducted in the form of exam.

The maximum amount of scores that can be obtained during the exam is 40 points.

The overall points of the discipline are 100. The total grade for the discipline is given according to the national and European scale.

12 Absence policy

Teaching of the discipline is based on cutting-edge educational technologies aimed at increasing the level of students' interest in the course, providing theoretical and practical knowledge of the discipline.

To activate the learning and cognitive activity of students, the discipline includes the consolidation of knowledge obtained at the lecture and acquisition of practical skills in lecture topics during laboratory classes.

13 Absence policy

Points are not given for missed lectures. If students miss a laboratory work, they should perform the homework before the next laboratory work and present the results to the lecturer.

Students who have missed classes without valid reasons and have not participated in current control activities are not admitted to the final semester control. In this case, a mark 'non-admission' is put in the exam record on the day of the exam. Repeated taking of the exam of the discipline is appointed in case of accomplishing all types of educational, independent (individual) work stipulated by the working program of the academic discipline and is carried out according to the approved schedule of academic failure liquidation.

14 Policy of late task performance

Tasks submitted later are assessed with a lower grade.

15 Academic integrity policy

Participants in the educational process rely on the academic integrity principles. One should provide references to sources of information when using someone else's ideas, statements, data, as well as verified information.

16 Recommended sources of information

Primary:

1. Petryk M.R., Petryk, O.Y. Software modeling: study guide. Ternopil: Ternopil Ivan Puluj National Technical University, 2015. – 200 p.
2. Lavrishcheva K.M. Software engineering. – Kyiv. – 2008. – 319 p.
3. Study guide for the Software development technologies discipline for students in the 123 Computer Engineering specialty. – Poltava: PoltNTU, 2017. – 218 p.
4. K.Y. Zolotko, D.V. Krasnoshapka, S.F. Siryk, Methodical recommendations for the accomplishment of laboratory work in the Design of software systems discipline. – Dnipro, 2018. – 27 p.
5. Methodical guidelines for the laboratory workshop on the Software design discipline for students of 6.050103 Software Engineering area / Compiled by A.M. Akymenko, I.V. Bohdan. – Chernihiv: Chernihiv Institute of Information Business and Law, 2016. – 34 p.
6. Software product development technology [Electronic resource]: laboratory workshop for full-time and part-time students of the 6.050101 Computer Science area / compiled by. V.A. Lytvynov, M.V. Hladka, O.A. Khlobystova: NUFT, 2014. – 86 p.
7. Alistair Coburn. Modern methods of describing the functional requirements of the system.
8. Fowler Scott D. UML in a nutshell. Application of the standard object modeling language: Translated from English. – M.: Mir, 1999. – 191 p.
9. Smirnov V.V. Technology of designing software systems. Laboratory works / V.V. Smirnov, N.V. Smirnova. – Kirovohrad: KNTU, 2012. – 53 p.
10. Smirnov V.V. Technology of designing software systems. Lectures / V.V. Smirnov, N.V. Smirnova. – Kirovohrad: KNTU, 2012. – 95 p.

Internet resources:

1. Definition of application architecture using Rational Software Architect. Available at: https://www.ibm.com/developerworks/ru/library/r-define-application-architecture-rational-software-architect-2/index.html.
2. E-learning system of International European University. Software architecture and design course. Available at: https://dist.ieu.edu.ua/course/view.php?id=632.
3. C++ Tutorial. – Available at: https://www.tutorialspoint.com/cplusplus/index.htm.
4. Java Tutorial. – Available at: https://www.tutorialspoint.com/java/index.htm.
5. Matplotbib – library's official website. Available at: https://matplotlib.org.
6. UML Tutorial. – Available at: https://www.tutorialspoint.com/uml/index.htm.

17 Tips on successful study during the course

Note: examine lecture materials and perform laboratory tasks synchronously with the curriculum. Thus, your abilities and insistence will be the key to success!