



PROGRAMMING. PART 2. FUNDAMENTALS OF WEB PROGRAMMING AND MOBILE APPLICATION DEVELOPMENT

Working program of the academic discipline (Syllabus)

Requisites of the Course

Cycle of Higher Education	<i>First (Bachelor)</i>
Field of Study	<i>12 Information Technologies</i>
Specialty	<i>121 Software Engineering</i>
Education Program	<i>Software Engineering of Multimedia and Information Retrieval Systems</i>
Type of Course	<i>Normative</i>
Mode of Study	<i>Full-time</i>
Year of Study, Semester	<i>2nd year, 4th semester</i>
ECTS workload	<i>Lectures: 36 hours, computer workshop: 18 hours, laboratory work: 18 hours, self-study: 78 hours</i>
Testing and Assessment	<i>Exam, modular control work, calendar control</i>
Class Schedule	<i>According to the schedule for the spring semester of the current academic year (schedule.kpi.ua)</i>
Language of Instructions	<i>English</i>
Course Instructors	Lecturer: <i>PhD, assistant, Severin Andrii, severinandrey97@gmail.com</i> Computer workshop: <i>PhD, associate professor, Zabolotnia Tetiana, tetiana.zabolotnia@gmail.com</i> Laboratory work: <i>PhD, assistant, Severin Andrii, severinandrey97@gmail.com</i>
Access to the Course	Google classroom: https://classroom.google.com/

Outline of the Course

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

The study of the "Programming. Part 2. Fundamentals of Web Programming and Mobile Application Development" discipline allows students to develop the competencies necessary for solving practical tasks of professional activity related to the development of software for network and mobile systems.

The **purpose** of studying the "Programming. Part 2. Fundamentals of Web Programming and Mobile Application Development" discipline is the formation of students' abilities to independently develop software for creating algorithms and data processing tools in network systems, as well as mobile applications based on the Android operating system.

The **subject** of the "Programming. Part 2. Fundamentals of Web Programming and Mobile Application Development" discipline is the technologies of developing software products for network and mobile systems using Java language.

The study of the "Programming. Part 2. Fundamentals of Web Programming and Mobile Application Development" discipline forms **professional competencies (PC)** in students necessary for solving practical problems of professional activity related to the development, improvement, and maintenance of intelligent information systems for processing multimedia data:

PC01 Ability to identify, classify and formulate software requirements.

PC02 Ability to participate in software design, including modelling (formal description) of its structure, behaviour, and functioning processes.

PC03 Ability to develop architectures, modules and components of software systems.

PC05 Ability to adhere to specifications, standards, rules, and recommendations in the professional field when implementing life cycle processes.

PC07 Knowledge of data information models, ability to create software for data storage, extraction 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 regarding the creation and maintenance of software and the recognition of the importance of a lifelong learning.

PC13 Ability to reasonably choose and master software development and maintenance tools.

PC14 Aptitude for algorithmic and logical thinking.

PC16 Ability to develop software of information retrieval systems.

PC17 Ability to develop software of multimedia systems.

Program learning outcomes (PLO) according to the educational program:

PLO01 To analyze, purposefully search for and select the information and reference resources and knowledge necessary for solving professional tasks, taking into account modern achievements of science and technology.

PLO03 To know the main processes, phases and iterations of the software life cycle.

PLO04 To know and apply professional standards and other regulatory documents in the field of software engineering.

PLO06 Ability to choose and use a software development methodology appropriate to the task.

PLO07 Know and apply in practice the fundamental concepts, paradigms and basic principles of the functioning of linguistic, instrumental and computational tools of software engineering.

PLO08 To be able to develop a human-machine interface.

PLO13 To know and apply methods of developing algorithms, designing software, data and knowledge structures.

PLO15 Reasonably choose programming languages and development technologies to solve the tasks of creating and maintaining software.

PLO18 To know and be able to apply information technologies for data processing, storage and transmission.

PLO23 To be able to document and present the results of software development.

PLO30 To be able to apply programming technologies for the development of software for multimedia and information retrieval systems.

PLO31 To know and be able to apply the principles of building retrieval systems, methods and algorithms for performing various types of information retrieval in them, criteria for evaluating the effectiveness of information retrieval.

PLO32 To know the main models of presenting of textual and multimedia information and methods of its pre-processing for use in the design of information retrieval systems.

PLO33 To know and be able to use software resources and libraries for processing text information and multimedia data in information retrieval systems.

2. Pre-requisites and post-requisites of the discipline (place in the structural and logical scheme of training according to the relevant educational program)

The successful study of the "Programming. Part 2. Fundamentals of Web Programming and Mobile Application Development" discipline is preceded by the study of the "Programming. Part 1. Object Oriented Programming and Design Patterns", "Fundamentals of Computer Systems and Networks", "Databases" disciplines of the curriculum of bachelor's training in specialty 121 Software Engineering.

The theoretical knowledge and practical skills obtained during the mastering of the "Programming. Part 2. Fundamentals of Web Programming and Mobile Application Development" discipline ensure the successful study of the "Multimedia Systems Software", "Information Retrieval Systems Software" disciplines, completion of pre-diploma practice, implementation of course projects and diploma projects in specialty 121 Software Engineering

3. Content of the course

The "Programming. Part 2. Fundamentals of Web Programming and Mobile Application Development" discipline involves the study of the following topics:

Topic 1. Object-oriented design using Java

Topic 2. Web programming using Java

Topic 3. Developing mobile applications for Android OS

Exam

4. Educational materials and resources

Basic literature:

1. Benjamin J. Evans, Jason Clark, David Flanagan. *Java in a Nutshell. 8th Edition: O'Reilly Media, Inc., 2023. – 479 p.*
2. Josh Juneau. *Java EE 8 Recipes (2nd Ed.) / Josh Juneau – Apress, 2018. – 792 p.*
3. Ian F. Darwin. *Android Cookbook: Problems and Solutions for Android Developers 2nd Edition / Ian F. Darwin – O'Reilly, 2017. – 772 p.*

Additional literature:

4. Herbert Schildt. *Java: The Complete Reference, Twelfth Edition: McGraw Hill, 2021.- 1280 p.*
5. Cay S. Horstmann. *Core Java, Volume I –Advanced Features, Eleventh Edition: Pearson, 2019. – 1040 p.*
6. Cay S. Horstmann. *Core Java, Volume I—Fundamentals, Eleventh Edition: Pearson, 2018. – 928 p.*
7. Google Java Style Guide. <https://google.github.io/styleguide/javaguide.html>
8. *Head First Java, 3rd Edition: A Brain-Friendly Guide / Kathy Sierra, Bert Bates, Trisha Gee – O'Reilly Media, 2022. – 752 p.*
9. C. Ullenboom. *Java: The Comprehensive Guide to Java Programming for Professionals. – Rheinwerk Computing, 2022. – 1128p.*
10. Bryan Sills, Brian Gardner, Kristin Marsicano, Chris Stewart. *Android Programming: The Big Nerd Ranch Guide, 5th Edition: Addison-Wesley Professional, 2022. – 688 p.*
11. Reto Meier. *Professional Android 4 Application Development, 3rd edition: Wrox, 2012. – 864 p.*
12. Kotlin. <https://kotlinlang.org/>.

13. *Android developer guides*. <https://developer.android.com/guide>.

14. *M.Fowler Refactoring: Improving the Design of Existing Code (2nd Edition) (Addison-Wesley Signature Series (Fowler)) 2nd Edition ISBN-13: 978-0134757599 – 2018. – 448 pp.*

15. *Clean Code: A Handbook of Agile Software Craftsmanship / Robert C. Martin. – Prentice Hall, 2009. – 464 p.*

16. *Clean Architecture: A Craftsman's Guide to Software Structure and Design (Robert C. Martin Series) 1st Edition / Robert C. Martin. – Prentice Hall, 2018. – 420 p.*

Educational content

5. Methods of mastering an educational discipline (educational component)

No.	Type of training session	Description of the training session
<i>Topic 1. Object-oriented design using Java</i>		
1	<i>Lecture 1. Course content, introduction to the discipline</i>	<i>Overview of course content. Approaches and technologies for developing web applications and mobile applications. Tasks on self-study: Item 6 No. 1.</i>
2	<i>Computer workshop 1. Choosing the topic of the software being developed, selecting and studying literature</i>	<i>Performing tasks according to the student's variant. Tasks on self-study: Item 6 No. 2.</i>
3	<i>Lecture 2. Programming in Java</i>	<i>Introduction to the Java API, the java.lang package, the concept of packages and imports, the static keyword, and base classes. Tasks on self-study: Item 6 No. 3.</i>
4	<i>Laboratory work 1. Developing an application using Java language (part 1)</i>	<i>Performing laboratory work tasks according to the student's variant. Tasks on self-study: Item 6 No. 4.</i>
5	<i>Lecture 3. Java Virtual Machine. Building and management of projects</i>	<i>The concept of the Java virtual machine, its features. Building and management of projects. Maven and Gradle Tasks on self-study: Item 6 No. 5.</i>
6	<i>Computer workshop 2. Preparation of technical specifications</i>	<i>Performing tasks according to the student's variant. Tasks on self-study: Item 6 No. 6.</i>
7	<i>Lecture 4. Unit testing in Java</i>	<i>The concept of unit tests. Phases of unit testing. Java libraries for writing unit tests. Tasks on self-study: Item 6 No. 7.</i>
8	<i>Laboratory work 1. Developing an application using Java language (part 2)</i>	<i>Performing laboratory work tasks according to the student's variant. Tasks on self-study: Item 6 No. 8.</i>
9	<i>Lecture 5. Architectural Patterns: MVC, MVP and MVVM</i>	<i>Architectural Patterns: MVC, MVP, and MVVM. Examples of their application. Tasks on self-study: Item 6 No. 9.</i>

10	<i>Computer workshop 3. Defining the structure of the software being developed, selecting design templates</i>	<i>Performing tasks according to the student's variant. Tasks on self-study: Item 6 No. 10.</i>
11	<i>Lecture 6. Client-server architecture</i>	<i>The concept of client-server architecture. Thick client and thin client, their features. Tasks on self-study: Item 6 No. 11.</i>
12	<i>Laboratory work 1. Developing an application using Java language (part 3)</i>	<i>Performing laboratory work tasks according to the student's variant. Tasks on self-study: Item 6 No. 12.</i>
<i>Topic 2. Web programming using Java</i>		
13	<i>Lecture 7. JDBC technology for working with databases</i>	<i>JDBC drivers and configuration. Interfaces and classes of the java.sql package. Query operators. Tasks on self-study: Item 6 No. 13.</i>
14	<i>Computer workshop 4. Preparation of the first version of the program structure diagram, class diagrams</i>	<i>Performing tasks according to the student's variant. Tasks on self-study: Item 6 No. 14.</i>
15	<i>Lecture 8. ORM technology for working with databases</i>	<i>The concept of ORM, its purpose. Advantages and disadvantages of using ORM. Hibernate as a standard ORM for the Java platform. Tasks on self-study: Item 6 No. 15.</i>
16	<i>Laboratory work 2. Development of a web application that implements CRUD operations (part 1)</i>	<i>Performing laboratory work tasks according to the student's variant. Tasks on self-study: Item 6 No. 16.</i>
17	<i>Lecture 9. HTTP Protocol</i>	<i>The concept of the HTTP protocol. Request format, methods, headers, response format, response statuses. HTTPS. Tasks on self-study: Item 6 No. 17.</i>
18	<i>Computer workshop 5. Finalization of the structure of modules and classes</i>	<i>Performing tasks according to the student's variant. Tasks on self-study: Item 6 No. 18.</i>
19	<i>Lecture 10. Servlets</i>	<i>Interfaces ServletContext, ServletConfig, ServletRequest, HttpServletResponse, HttpSession, HttpServletRequest, HttpSession, HttpSessionBindingInfo, HttpSessionBindingInfo, HttpSessionBindingInfo, HttpSessionBindingInfo. Request processing. Tasks on self-study: Item 6 No. 19.</i>
20	<i>Laboratory work 2. Development of a web application that implements CRUD operations (part 2)</i>	<i>Performing laboratory work tasks according to the student's variant. Tasks on self-study: Item 6 No. 20.</i>
21	<i>Lecture 11. HTML markup language. JSP technology</i>	<i>HTML concepts. Basic HTML elements. JSP document. Standard action elements. JSTL tag library. Expression Language. Tasks on self-study: Item 6 No. 21.</i>
22	<i>Computer workshop 6. Development of</i>	<i>Performing tasks according to the student's variant.</i>

	<i>algorithms and software interface</i>	<i>Tasks on self-study: Item 6 No. 22.</i>
23	<i>Lecture 12. Authentication, pagination and localization of web applications</i>	<i>Built-in authentication and authorization capabilities in java. Approaches to pagination. The concept of localization and internationalization. Approaches to implementing localization in web applications. Tasks on self-study: Item 6 No. 23.</i>
24	<i>Laboratory work 2. Development of a web application that implements CRUD operations (part 3)</i>	<i>Performing laboratory work tasks according to the student's variant. Tasks on self-study: Item 6 No. 24.</i>
<i>Topic 3. Developing mobile applications for Android OS</i>		
25	<i>Lecture 13. Introduction to mobile application development</i>	<i>Operating systems for mobile application development. Approaches to mobile application development. Overview of programming languages for mobile platforms. Tasks on self-study: Item 6 No. 25.</i>
26	<i>Computer workshop 7. Demonstration of the first version of the developed software</i>	<i>Performing tasks according to the student's variant. Tasks on self-study: Item 6 No. 26.</i>
27	<i>Lecture 14. Android SDK. Android Studio. Android Project Structure</i>	<i>Android platform architecture. Android application components. Android SDK versions, their compatibility. Integrated mobile application development environment Android Studio. Android project structure. Tasks on self-study: Item 6 No. 27.</i>
28	<i>Laboratory work 3. Development of a mobile application that implements CRUD operations (part 1)</i>	<i>Performing laboratory work tasks according to the student's variant. Tasks on self-study: Item 6 No. 28.</i>
29	<i>Lecture 15. Building a user interface. Basic widgets</i>	<i>Basics of creating UI design. Basic widgets: TextView, ImageView, EditText, ScrollView, Button, RadioButton, ToggleButton, ImageButton, ProgressBar, Toast, AlertDialog, ProgressDialog. Working with menus. Menu interface, ContextMenu, SubMenu classes. Tasks on self-study: Item 6 No. 29.</i>
30	<i>Computer workshop 8. Finalization of work, testing of developed software</i>	<i>Performing tasks according to the student's variant. Tasks on self-study: Item 6 No. 30.</i>
31	<i>Lecture 16. Navigation. Fragments</i>	<i>Navigation principles. Intent class. Navigation without data transmission. Navigation with data transmission. Fragments. Fragment, FragmentActivity, and FragmentManager classes. Tasks on self-study: Item 6 No. 31.</i>

32	Laboratory work 3. Development of a mobile application that implements CRUD operations (part 2)	Performing laboratory work tasks according to the student's variant. Tasks on self-study: Item 6 No. 32.
33	Lecture 17. Adapters. Working with the database	Purpose and types of adapters. ListView and ArrayAdapter classes. Using adapters and collections based on the Map interface together. Adapters for composite collections. BaseAdapter class. Working with a database in Android. SQLite API and Room technologies. Tasks on self-study: Item 6 No. 33.
34	Computer workshop 9. Writing and formatting an explanatory note	Performing tasks according to the student's variant. Tasks on self-study: Item 6 No. 34.
35	Laboratory work 3. Development of a mobile application that implements CRUD operations (part 3)	Performing laboratory work tasks according to the student's variant. Tasks on self-study: Item 6 No. 35.
<i>Modular control work</i>		

6. Self-study

The "Programming. Part 2. Fundamentals of Web Programming and Mobile Application Development" discipline is based on self-study for classes on theoretical and practical topics.

No.	Topic for self-studying	Number of hours	Literature
1	Preparation for lecture 1	1	1-5; 9; 10
2	Preparation for computer workshop 1	1	14-16
3	Preparation for lecture 2	1	1; 4-8; 14-15
4	Preparation for laboratory work 1 (part 1)	2	1; 4-8; 14-15
5	Preparation for lecture 3	1	1; 4-8; 14-15
6	Preparation for computer workshop 2	1	14-16
7	Preparation for lecture 4	1	1; 4-8; 14-15
8	Preparation for laboratory work 1 (part 2)	2	1; 4-8; 14-15
9	Preparation for lecture 5	1	1; 4-8; 16
10	Preparation for computer workshop 3	1	14-16
11	Preparation for lecture 6	1	1; 4-8; 16
12	Preparation for laboratory work 1 (part 3)	2	1; 4-8; 14-16
13	Preparation for lecture 7	1	2; 5; 9; 14-16
14	Preparation for computer workshop 4	1	14-16

15	Preparation for lecture 8	1	2; 5; 9; 14-16
16	Preparation for laboratory work 2 (part 1)	2	2; 5; 9; 14-16
17	Preparation for lecture 9	1	2; 5; 9; 14-16
18	Preparation for computer workshop 5	1	14-16
19	Preparation for lecture 10	1	2; 5; 9; 14-16
20	Preparation for laboratory work 2 (part 2)	2	2; 5; 9; 14-16
21	Preparation for lecture 11	1	2; 5; 9; 14-16
22	Preparation for computer workshop 6	1	14-16
23	Preparation for lecture 12	1	2; 5; 9; 14-16
24	Preparation for laboratory work 2 (part 3)	2	2; 5; 9; 14-16
25	Preparation for lecture 13	1	3; 5; 10-16
26	Preparation for computer workshop 7	1	14-16
27	Preparation for lecture 14	1	3; 5; 10-16
28	Preparation for laboratory work 3 (part 1)	2	3; 5; 10-16
29	Preparation for lecture 15	1	3; 5; 10-16
30	Preparation for computer workshop 8	1	14-16
31	Preparation for lecture 16	1	3; 5; 10-16
32	Preparation for laboratory work 3 (part 2)	2	3; 5; 10-16
33	Preparation for lecture 17	1	3; 5; 10-16
34	Preparation for computer workshop 9	1	14-16
35	Preparation for laboratory work 3 (part 3)	2	3; 5; 10-16
36	Preparation for modular control work	4	1-16
37	Preparation for exam	30	1-16

Policy and Control

7. Policy of academic discipline (educational component)

Attending lectures is mandatory.

Attending laboratory work classes may be occasional and as needed for consultation/protection of laboratory work.

Rules of behavior in classes: activity, respect for those present, turning off phones.

Adherence to the policy of academic integrity.

Rules for the protection of laboratory work: the work must be performed in accordance with the assigned tasks and according to the option chosen by the student.

Incentive points are awarded for:

- creative approach to laboratory work. The maximum number of incentive points for one laboratory work is 2 points, the maximum per semester is 5 points.

8. Types of control and rating system for evaluating learning outcomes

*During the semester, students perform **3 laboratory works**. The maximum number of points for each laboratory work: 14 points.*

Points are awarded for:

*- the quality of performance of laboratory work: 0-7 points;
- answer during the defense of laboratory work: 0-5 points;
- timely presentation of work for defense: 0-2 points.*

Criteria for evaluating the quality of performance:

*7 points – the work was done qualitatively, in full;
5-6 points – the work is done qualitatively, in full, but has shortcomings;
3-4 points – the work is done in full, but contains minor errors;
1-2 points – the work done in full, but contains significant errors;
0 points – the work is not fully completed.*

Criteria for evaluating response:

*5 points – the answer is complete, well-reasoned;
3-4 points – the answer is correct, but has flaws or minor errors;
1-2 points – the answer contains significant errors;
0 points – there is no answer or the answer is incorrect.*

Criteria for evaluating the timeliness of submission of work for defense:

*2 points – the work is submitted for defense no later than the specified deadline;
0 points – the work is submitted for defense after the specified deadline.*

The maximum number of points for performing and defending laboratory work:

14 points × 3 laboratory works = 42 points.

*The **modular control work** consists of 1 theoretical question and 1 practical task.*

The maximum number of points for a theoretical question is 3 points.

Criteria for evaluating each theoretical question:

*3 points – the answer is correct, complete, and well-reasoned;
2 points – the answer is generally correct, but has shortcomings;
1 point – the answer contains significant errors;
0 points – there is no answer or the answer is incorrect.*

The maximum number of points for a practical task is 5 points.

Criteria for evaluating the practical tasks:

*5 points – the answer is correct;
3-4 points – the answer is generally correct, but has shortcomings;
1-2 points – the answer contains significant errors;
0 points – there is no answer or the answer is incorrect.*

The maximum number of points for modular control work:

3 points × 1 theoretical question + 5 points × 1 practical task = 8 points.

The rating scale for the discipline is equal to:

$R = R_C = R_{lab. works} + R_{modular control work} + R_{exam} = 42 points + 8 points + 50 points = 100 points.$

Calendar control: conducted twice a semester as a monitoring of the current state of fulfillment of the syllabus requirements.

At the first certification (7th week), the student receives "Passed" if his current rating is at least 7 points (50% of the maximum number of points that the student can receive before the first certification).

At the second certification (13th week), the student receives "Passed" if his current rating is at least 14 points (50% of the maximum number of points that the student can receive before the second certification).

Semester control: exam.

Conditions for admission to semester control:

A prerequisite for a student's admission to the exam is a semester rating (R_S) of at least 30 points.

After passing the exam, a grade is assigned according to the table (Table of correspondence of rating points to grades on the university scale).

The exam task consists of 3 questions - 2 theoretical and 1 practical. The answer to each theoretical question is worth 15 points, and the answer to a practical question is worth 20 points.

Evaluation criteria for a theoretical question:

13–15 points – the answer is correct, complete, well-argued;

8–12 points – in general, the answer is correct, but it has flaws;

1–7 points – there are significant errors in the answer;

0 points – there is no answer or the answer is incorrect.

Evaluation criteria for a practical question:

16–20 points – the answer is correct, complete, well-argued;

9–15 points – in general, the answer is correct, but it has flaws;

1–8 points – there are significant errors in the answer;

0 points – there is no answer or the answer is incorrect.

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

Score	Grade
100-95	Excellent
94-85	Very good
84-75	Good
74-65	Satisfactorily
64-60	Enough
Less than 60	Unsatisfactorily
Admission conditions not met	Not allowed

9. Additional information on the discipline (educational component)

The list of questions submitted for semester control is given in Appendix 1.

Working program of the academic discipline (syllabus):

Is designed by PhD, assistant, Severin Andrii

Adopted by Computer Systems Software Department (protocol № 11 from 30.04.2025)

Approved by the Faculty Board of Methodology (protocol № 11 from 23.05.2025)

Appendix 1. List of questions submitted for semester control

- 1. The concept of a Java virtual machine, its features.*
- 2. Building and management of projects.*
- 3. The concept of unit tests. Java libraries for writing unit tests.*
- 4. Architectural Patterns: MVC, MVP, and MVVM.*
- 5. The concept of client-server architecture.*
- 6. JDBC technology.*
- 7. ORM technology.*
- 8. The concept of the HTTP protocol. Request format, methods, headers, response format, response statuses.*
- 9. Interfaces ServletContext, ServletConfig, ServletRequest, HttpServletRequest, ServletResponse, HttpServletResponse. Request processing.*
- 10. Standard action elements. JSP document. JSTL tag library. Expression Language.*
- 11. Built-in authentication and authorization capabilities in java.*
- 12. Approaches to pagination.*
- 13. The concept of localization and internationalization. Approaches to implementing localization in web applications.*
- 14. Operating systems for mobile application development. Approaches to mobile application development.*
- 15. Android platform architecture. Android application components.*
- 16. Integrated development environment for mobile applications Android Studio. Structure of an Android project.*
- 17. Basics of UI Design. Examples of Basic Widgets.*
- 18. Working with menus. Menu interface, ContextMenu, SubMenu classes.*
- 19. Navigation principles. Intent class. Navigation.*
- 20. Fragments. Fragment, FragmentActivity, and FragmentManager classes.*
- 21. Purpose and types of adapters. ListView and ArrayAdapter classes.*
- 22. Using adapters and collections based on the Map interface together.*
- 23. Adapters for composite collections. BaseAdapter class.*
- 24. Working with a database in Android. SQLite API and Room technologies.*