

Syllabus
Autumn semester of the 2025-2026 academic year
Educational program "6b06101 - Information Systems"

Subject ID and Name	Student's Independent Work (SIW)	Number of Credits			Loans general number	Independent work of the student under the guidance of the teacher (OBS)
		Lectures	Seminar classes	Laboratory work (Lab)		
100842 programming in Java	4	1,70	0,0	3,30	5	4

ACADEMIC INFORMATION ABOUT THE SUBJECT

Type of training	Cycle, component	Types of lectures	Types of seminar classes	Type and platform of the final control
Offline	Midterm, Final Exam	lecture-presentation, interactive discussion	laboratory work, code writing, project tasks	Written offline
Lecturer	Макетов Максат Толеужанович			
e-mail:	Maketov.Maxat@kaznu.kz			
Телефоны:	87777394467			
Assistant	-			
e-mail:	-			
Phone number:	-			

ACADEMIC PRESENTATION OF THE SUBJECT

Purpose of the Subject	Expected Learning Outcomes	Indicators for Achieving Learning Outcomes
To teach the principles and techniques of object-oriented programming in Java, and to train students to write efficient and optimal code in Java. To develop students' skills in using data structures, multithreading, and modern technologies within the Java ecosystem.	To understand and apply object-oriented programming in Java.	1.1 Students understand the main principles of object-oriented programming in Java (encapsulation, inheritance, polymorphism, abstraction).
		1.2 They effectively apply OOP to solve practical tasks.
	2. To be able to use data structures in Java.	2.1 Students understand Java data structures such as arrays, lists, maps, and trees.
		2.2 They can find efficient solutions by applying various data structures to practical problems.
	3. To master the fundamentals of multithreading in Java.	3.1 They grasp the concept of multithreading.
		3.2 They create multithreaded applications using Thread and Runnable objects.
		They correctly apply synchronization in multithreaded systems.
	4. To effectively use Exception Handling.	4.1 They understand the principles of handling errors and exceptions in Java.
		4.2 They protect code from errors by using try-catch-finally blocks and correctly implement exception handling to enhance program stability.
	5. To acquire skills in creating graphical user interfaces (GUI) in Java.	5.1 They can create graphical interfaces using JavaFX or Swing libraries, design user interfaces,

		and add visual elements such as buttons, text, and windows. 5.2 They enable a Java application to interact with a database (MySQL, PostgreSQL) using JDBC.
Prerequisites	1. Algorithms and Data Structures 2. Object-Oriented Programming 3. Mathematical Logic and Discrete Mathematics 4. Networks and Fundamentals of Information Security	
Postrequisites	1. Web Technologies 2. Mobile Application Programming 3. Database Management Systems (DBMS) – JDBC, SQL, NoSQL 4. Project Management and DevOps – Software Project Management	
Learning Resources	Literature: Primary and Supplementary 1. Beitman, M. (2021). Programming in Java. Binom Publishing. This book fully covers the basics of Java and provides practical examples and exercises. 2. Herbert Schildt (2019). Java: The Complete Reference. MIF Publishing. Schildt's book is aimed at deeper mastery of Java, covering object-oriented programming and modern features. 3. Kathy Sierra, Bert Bates (2020). Head First Java. Olimp Publishing. This book is very suitable for beginners in Java, explaining fundamentals clearly and providing many tests and exercises. 4. Safonov, D. (2022). Java for Professionals. DMK Press. A book demonstrating how to write high-level Java programs and efficient code, with user tips and solutions for common mistakes. 5. Robert C. Martin (2020). Clean Code: Creation, Analysis, and Refactoring. Williams Publishing. The book presents principles and best practices of clean coding, which can be effectively applied in Java. Research Infrastructure 1. Computer labs of the Department of Information Systems, Al-Farabi Kazakh National University 2. Artificial Intelligence and Big Data Laboratory Professional Scientific Databases 1. IEEE Xplore – IEEE Xplore 2. ACM Digital Library – ACM Digital Library 3. SpringerLink – SpringerLink 4. ScienceDirect – ScienceDirect 5. Web of Science – Web of Science Internet Resources 1. KazNU Electronic Library 2. Natural Language Toolkit (NLTK) 3. spaCy Documentation Software Tools 1. IntelliJ IDEA 2. JDK (Java Development Kit)	

Academic Policy of the Subject	The subject's academic policy is defined by the Academic Policy and the Academic Integrity Policy of Al-Farabi Kazakh National University. The documents are available on the main page of the Univer IS. Integration of Science and Education: The research work of students, master's students, and doctoral candidates represents an in-depth extension of the learning process. It is conducted directly in departments, laboratories, the university's research and design units, and student scientific and technical associations. Independent work at all levels of education is aimed at developing research skills and competencies based on acquiring new knowledge using modern research and information technologies. The research university instructor integrates the results of research activities into lectures, seminar (practical) classes, laboratory sessions, and syllabus topics, aligning them with the relevance of course lessons and assignments. Attendance: The deadlines for each assignment are indicated in the subject's implementation calendar (schedule). Failure to meet deadlines will result in loss of points. Academic Integrity: Practical/laboratory sessions and independent work develop the learner's autonomy, critical thinking, and creativity. Plagiarism, falsification, use of cheat sheets, and
--------------------------------	---

copying at any stage of task completion are strictly prohibited. During theoretical instruction and examinations, maintaining academic integrity is regulated not only by the main policies but also by documents such as the "Rules for Conducting Final Assessment," "Guidelines for Conducting End-of-Semester Assessments for the Fall/Spring Semester of the Current Academic Year," and the "Rules for Verifying the Copying of Students' Test Documents."

Key Principles of Inclusive Education: The university's educational environment is designed as a safe space where all students receive support and fair treatment from instructors and peers, regardless of gender, racial/ethnic background, religious beliefs, socio-economic status, physical health, and other factors. Everyone needs peer support and friendship. Success for all students lies in what they are capable of achieving rather than in what is impossible. Diversity enriches all aspects of life.

All learners, especially those with disabilities, can receive advisory support via phone/email at baikuvekov_meirzhan3@live.kaznu.kz or through MS Teams video call: <https://us04web.teams.us/j/73674837609?pwd=AID37HhIShnBYtJOgU7qqmDQ9oK4na.I.>

MOOC Integration (Massive Open Online Course): When a MOOC is integrated into the subject, all students are required to register. The deadlines for completing MOOC modules must strictly follow the course schedule.

Attention! The deadline for each assignment is indicated in the subject's implementation calendar (schedule) and in MOOC. Failure to meet deadlines will result in the loss of points.

INFORMATION ON TEACHING, LEARNING, AND ASSESSMENT

Letter grading system corresponding to the numeric four-point scale for evaluating students' academic achievements				Assessment Methods
Letter Grade Evaluation on Numeric	Equivalent of Points	Percentage Content	Traditional Grading System	Criterion-Based Assessment – a process that weighs the actual learning outcomes achieved against the expected learning outcomes based on clearly defined criteria. It is based on both formative and summative assessment. Formative Assessment – a type of assessment conducted during daily learning activities. It serves as an ongoing indicator, providing immediate interaction between the student and the instructor. It helps identify student capabilities, detect difficulties, achieve optimal results, and allows the instructor to timely adjust the educational process. Performance is evaluated during lectures, seminars, and practical sessions (discussions, quizzes, debates, round tables, laboratory work, etc.), as well as class participation. Acquired knowledge and competencies are assessed. Summative Assessment – a type of assessment conducted after completing a section according to the course program. It is carried out 3–4 times per semester during independent work (SIW). It evaluates the mastery of expected learning outcomes using descriptors. This allows determination and recording of the level of subject mastery at a specific stage. Learning outcomes are assessed.
A	4,0	95-100	Very Good	
A-	3,67	90-94		
B+	3,33	85-89		
B	3,0	80-84	Good	
B-	2,67	75-79		
C+	2,33	70-74		
C	2,0	65-69		
C-	1,67	60-64		
D+	1,33	55-59	Satisfactory	
D	1,0	50-54		
FX	0,5	25-49		
F	0	0-24	Unsatisfactory	

Course Implementation Calendar (Schedule). Teaching and Learning Methods.

Аптабы	Topic Title	Number of Hours	Maximum Score
MODULE 1: Basic Structures and OOP			
1	1-Lecture: Introduction to Java. Architecture of JVM, JDK, JRE.	1	
2	1-Lab: Setting up the Java environment (IntelliJ IDEA/Eclipse). Writing "Hello, World!".	1	
	2-Lecture: Data types, variables, operators.	1	
3	2-Lab: Solving problems using variables and arithmetic operations.	1	
	3-Lecture: Arrays and Strings (String, StringBuilder).	1	
4	3-Lab: Using arrays: finding the largest/smallest element.	1	
	4-Lecture: Methods and parameters. Recursion.	1	
	4-Lab: Working with strings: text analysis and reversal.	1	
	1-Independent Work Guidance: Tips for completing IW 1. "Exploring Object-Oriented Programming in Java."		0
5	5-Lecture: Fundamentals of Object-Oriented Programming (OOP): class, object, encapsulation.	1	0
	5-Lab: Creating classes, using objects. Examples of encapsulation.	2	15
	1-Independent Work: Submission of IW 1 "Exploring Object-Oriented Programming in Java."		20
MODULE 3: Inheritance, Interfaces, and Collections			
6	6-Lecture: Inheritance and Polymorphism.		
	6-Lab: Creating class hierarchies using inheritance (Student, Master Student).	2	15
	2-Independent Work Guidance: Tips for completing IW 2. "Exploring Data Structures in Java."		0
7	7-Lecture: Abstract Classes and Interfaces.		
	7-Lab: Using Interfaces (Comparable, Iterable).	2	15
	2-Independent Work: Submission of IW 2 "Exploring Data Structures in Java."		20
8	8-Lecture: Exception Handling.	1	0
	8-Lab: Exception Handling: examples using try-catch-finally.	2	15
Midterm Assessment 1			
9	9-Lecture: Java Collections Framework (List, Set, Map).	1	0
	9-Lab: Working with collections: solving problems using ArrayList, HashMap, TreeSet.	2	10
	3-Independent Work Guidance: Tips for completing IW 3. "Fundamentals of Multithreading in Java."		
10	10-Lecture: Generics and Lambda Expressions.	1	0
	10-Lab: Generics and Lambda: creating functional interfaces.	2	10
	3-Independent Work: Submission of IW 3 "Fundamentals of Multithreading in Java."		15
MODULE 3: Advanced Features and Design in Java			
11	11-Lecture: Working with the File System (File I/O, NIO).	1	0
	11-Lab: Reading/writing files (txt, csv).	2	10
12	12-Lecture: Multithreading (Multithreading, Executor).	1	0
	12-Lab: Creating threads: Thread, Runnable.	2	10
13	13-Lecture: Basics of JavaFX (GUI Applications).	1	0
	13-Lab: Creating a simple calculator using JavaFX.	2	10
	4-Independent Work Guidance: Tips for completing IW 4. "Working with Databases in Java: Connecting to a Database via JDBC."		
14	14-Lecture: Databases and JDBC.	1	0
	14-Lab: Connecting to MySQL/PostgreSQL via JDBC.	2	10
	4-Independent Work: Submission of IW 4 "Working with Databases in Java: Connecting to a Database via JDBC."		15
15	15-Lecture: Preparing the Course Project. Modern Java Ecosystem (Overview of Spring Boot).	1	0
	15-Lab: Project: small information system (e.g., "Student List" CRUD application).	2	10
Midterm Assessment 2			
Final control (exam)			100
A set for the subject			100
			100

SUMMARY ASSESSMENT RUBRICATOR
CRITERIA FOR EVALUATING LEARNING OUTCOMES

Study of Object-Oriented Programming in Java (20% of 1AB 100%)

Criterion	"Very good" 20-25 %	"Good" 15-20%	"Satisfactory" 10-15%	"Unsatisfactory" 0-10%
Practical part of programming	In Java, specific and complete examples have been created that cover all aspects of Object-Oriented Programming. The quality of the code is high, there are no errors.	In Java, the main aspects of Object-Oriented Programming are implemented, but some parts of the code are incomplete or contain errors.	Only the basic principles of Object-Oriented Programming are indicated, errors in the code are common, some components do not work.	The use of Object-Oriented Programming in Java does not exist or is very limited. The code is full of errors, functional flaws are noticeable.
Project structure and design	The project structure is clear and logically organized. All classes, objects, and methods are well constructed and mutually compatible.	The project structure is clear, but the use of some classes and objects is incorrect or redundant.	The project structure is complex, but working. Since there is no connection between classes and objects, the code is difficult to understand.	The project structure is missing or incorrectly created, and the relationship between classes and objects is not clear.
Documentation and explanations	The documentation is complete, all code parts and methods are explained, and UML diagrams and other technical information are included.	Documentation exists, but some methods and parts of the code are not explained or have insufficient diagrams.	Documentation is less or very general, there are no comments to the code.	There is no or very little documentation. It is not clear for what purpose the code was written.

Study of data structures in Java (20% of 100% of 1AB)

Criterion	"Very good" 20-25 %	"Good" 15-20%	"Satisfactory" 10-15%	"Unsatisfactory" 0-10%
Practical part of programming	In Java, all types of data structures are implemented with specific and complete examples. The quality of the code is high, there are no errors, all structures are used efficiently.	Several types of data structures have been used in Java, but some are incorrect or incomplete.	Only a few basic data structures are shown, but some of them are incorrect or incomplete.	In Java, Data Structures were not used correctly or only one structure was used. There are many errors in the code.
Project structure and design	The project structure is logical, clearly organized, and all data structures are used in a coherent way. The organization and structure of the code is optimal.	The project structure is clear, but some structures are incorrectly selected or redundant.	The structure of the project is unclear, some structures are not interconnected or incorrectly formed.	The project structure is missing or incorrectly organized. The relationship of data structures is unclear.
Documentation and explanations	The documentation is complete, all code parts and data structures are explained, and UML diagrams and other technical information are included.	Documentation exists, but some code parts and data structures are not explained or have insufficient diagrams.	Documentation is less or very general, there are no comments to the code.	There is no or very little documentation. It is not clear for what purpose the code was written.

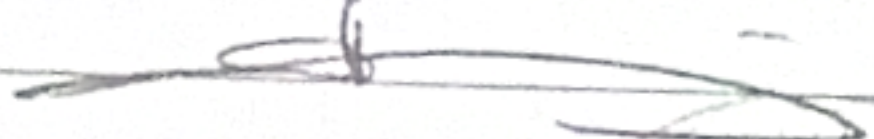
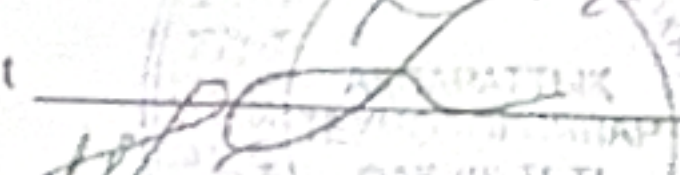
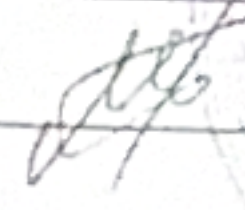
Basics of multithreaded programming in Java (20% of 2AB 100%)

Критерий	"Very good" 20-25 %	"Good" 15-20%	"Satisfactory" 10-15%	"Unsatisfactory" 0-10%
Practical part of programming	Multithreaded programming in Java is implemented with complete and specific examples. Proper flow management, synchronization, deadlock prevention, and other flow management methods have been used effectively.	Several aspects of multithreaded programming have been implemented in Java, but some thread management is incorrect or incomplete.	Only the main aspects of multithreaded programming are specified, the threads are not properly managed, and there are errors.	Multithreaded programming in Java is used incorrectly, there are many errors in the code and problems with thread management.
Project structure and design	The project structure is clear and logically organized. The work of the threads is correctly distributed, synchronization problems are solved, deadlock and other problems are prevented.	The project structure is clear, but some flow control or synchronization methods have been used incorrectly.	The project structure is complex, but the code is difficult to understand because there is no connection between threads. Synchronization problems are not solved.	The project structure is incorrect or the flow is not organized properly.
Documentation and explanations	The documentation is complete, all flows and synchronization methods are explained, and UML diagrams and other technical information are included.	Documentation exists, but some threads and synchronization methods are not explained or have insufficient diagrams.	Documentation is less or very general, there are no comments to the code.	There is no or very little documentation. It is not clear for what purpose the code was written.

Working with databases in Java: connecting to databases using JDBC (20% of 2AB 100%)

Критерий	"Very good" 20-25 %	"Good" 15-20%	"Satisfactory" 10-15%	"Unsatisfactory" 0-10%
Practical part of programming	The Java program provides examples of full and correct connection to the database using JDBC, execution of SQL queries, obtaining and processing data. The quality of the code is high, there are no errors.	Using JDBC, code is written based on connecting to a database and executing SQL queries, but some methods are incomplete or have errors.	Communication with the database and execution of requests via JDBC is carried out only at the basic level. There are errors or functional flaws in the code.	Establishing a connection to the database via JDBC is not implemented correctly, there are many errors in the code.
Structure and design	The project structure is clear, communication with the database is organized efficiently. All components (connection, queries, results) are correctly distributed, and optimal solutions are used.	The project structure is clear, but the connection to the database does not cover some problems or some pieces of code are redundant.	The project structure is complex, but some parts of the code are incompatible or redundant.	The project structure is incorrect or the database communication system is incorrectly organized.
Documentation and explanations	The documentation is complete, all parts of the code and methods are explained, the connection process.	There is documentation, but some parts of the code are not explained.	The documentation is more or less general, there are no explanations for why the code was written.	There is no or very little documentation. It is not clear for what purpose the code was written.

Criterion	"Great" 16 - 20%	"Fine" 9-15%	"Satisfactorily" 5-8%	"Unsatisfactory" 0-4%
Working with extended network topology and static routing	Understand the degree of compliance, relevance, and reliability of routing protocols. Knowledge and understanding of extended network topology and static routing	Understanding the degree of relevance, relevance, and reliability of routing protocols. Knowledge of extended network topology and static routing	Limited understanding of the consistency, relevance, and validity of extended network topology and static routing	Superficial understanding/lack of understanding of the degree of compliance, relevance, and reliability of routing protocols. Lack of knowledge of extended network topology and static routing
Designing skills	Clear and clear presentation of extended network topology and static routing	There are small logical errors in the extended network topology and static routing	A large number of logical and syntax errors in the extended network topology and static routing	No network topology and static routing
Writing a report	The writing demonstrates clarity, conciseness, and accuracy.	The writing demonstrates clarity, conciseness, and correctness. Mostly no errors.	There are some key errors in the writing, and the clarity needs improvement.	The writing is unclear, and it is difficult to follow the content. Lots of errors in the text

Dean  Imankulov T.S.
 Chair of the Academic Committee
 on the Quality of Teaching and Learning  Buribayev Zh.A.
 Head of Department  Shormakova A.N.
 Lecturer  Maketov M.

