National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”

Department
Information systems and
technologies

JAVA PROGRAMMING LANGUAGE

Working program of the academic discipline (Syllabus)

Details of the academic discipline

Level of higher education	*First (undergraduate)*
Branch of knowledge	*12 Information technologies*
Specialty	*126 Information systems and technologies*
Educational program	*Information support of robotic systems*
Discipline status	*Selective*
Form of education	*Full-time/correspondence/distance*
Year of training, semester	*3rd year, autumn semester*
Scope of the discipline	*120 hours (4 credits)*
Semester control/ control measures	*Test*
Lessons schedule	*http://rozklad.kpi.ua/Schedules/ViewSchedule.aspx?v=45e550fd-ec44-4564-8338-a6c6ae7fec74*
Language of teaching	*Ukrainian*
Information about head of the course / teachers	Lecturer / Laboratory:Senior teacher of the Department of IST., PhD, S.P. Orlenko,orlenko_sergey@tk.kpi.ua
Placement of the course	In the Telegram group of disciplines and in Campus

Program of educational discipline

1. Description of the educational discipline, its purpose, subject of study and learning outcomes

1.1. This course is aimed at learning Java by people with a minimum level of programming knowledge and people who want to improve their knowledge of certain nuances of the language. The goal of the educational discipline is for students to master the means of programming, development and maintenance of applied applications using the Java programming language to perform the following tasks:

- practical use of the capabilities and tools of the Java programming language when solving applied problems;
- formation of skills for working with tools used in programming in the Java language;
- mastering the basics of Java programming.

1.2. The main tasks of the academic discipline are for students to obtain the following **knowledge:**

- basic tools of the Java programming language;
- the main instrumental software tools of the Java language used to solve applied problems;

skill:

- design and develop batch applications using the tools of the Java language to solve various problems;

- use technical documentation;
- conduct application testing using application libraries;

experience:

- using instrumental software tools of the Java language;
- carry out development using version control systems;
- creating applications using the Java language.

Also, according to the terms of the educational discipline program, after completing the credit module, students must master the following learning outcomes:

COMPETENCES:

- CS 1The ability to analyze the object of design or operation and its subject area
- CS 4The ability to design, develop and use the means of implementing information systems, technologies and information communications (methodical, informational, algorithmic, technical, software and others)
- CS 6The ability to use modern information systems and technologies (production, decision support, intelligent data analysis, and others), information protection and cyber security methods when performing functional tasks and duties
- CS 10Ability to select, design, deploy, integrate, manage, administer and support information systems, technologies, information communications, services and infrastructure of organizations

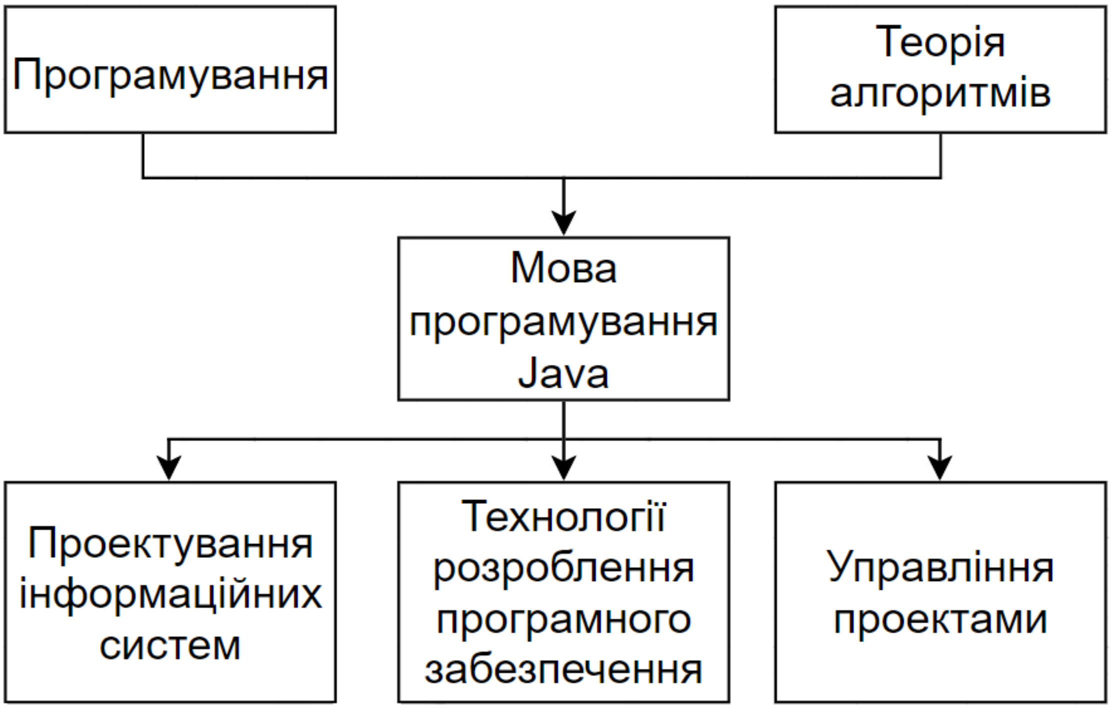
PROGRAM LEARNING OUTCOMES:

- PR 3 Use basic knowledge of informatics and modern information systems and technologies, programming skills, technologies for safe work in computer networks, methods of creating databases and Internet resources, technologies for developing algorithms and computer programs in high-level languages using object-oriented -oriented programming for solving design problems and using information systems and technologies
- PR 4 Conduct a system analysis of design objects and justify the choice of structure, algorithms and methods of information transmission in information systems and technologies
- PR 5 Argue the choice of software and technical means for creating information systems and technologies based on the analysis of their properties, purpose and technical characteristics, taking into account system requirements and operating conditions; have skills in debugging and testing software and technical means of information systems and technologies
- PR 6 Demonstrate knowledge of the modern level of information systems technologies, practical programming skills and the use of applied and specialized computer systems and environments with the aim of implementing them in professional activities
- PR 7 Justify the choice of technical structure and develop appropriate software that is part of information systems and technologies
- PR 14 To know modern programming languages and technologies for creating software for information systems and technologies

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

To master the course, basic knowledge of higher and discrete mathematics, procedural and object-oriented programming, theory of algorithms is required.

The discipline can serve as a basis for studying the following disciplines: "Technologies of software development", "Project management", "Design of information systems".


Програмування
Теорія алгоритмів
Мова програмування Java
Проектування інформаційних систем
Технології розроблення програмного забезпечення
Управління проектами

3. Content of the academic discipline

Chapter 1. Features of the Java programming language

1.1. History of development and application
1.2. The structure, functions and operation of the JVM

Chapter 2. The lexical structure of the language

2.1. Main operators
2.2. Types
2.3. arrays
2.4. Construction of classes
2.5. Processing deadlines
2.6. Regular expressions
2.7. Congestion resolution

Chapter 3. OOP in Java

3.1. Principles of OOP
3.2. Abstract classes
3.3. Interfaces
3.4. Nested, local and anonymous classes.
3.5. Access modifiers
3.6. Types of relationships between classes and interfaces
3.7. MVC architectural pattern
3.8. Creating and using packages

Chapter 4. Exceptions and generalized programming

4.1. Exceptional situations
4.2. Generalized programming in the Java language (Generics).

Chapter 5. Architecture and principles of software design

5.1. Software architecture
5.2. Software design principles and patterns

Chapter 6. Collections (Java Collections Framework)

6.1. Using collections
6.2. Hierarchy of collections

6.3. Realization of collections

Chapter 7. I/O streams

7.1 Flow organization
7.2. Serialization
7.3. Work with files
7.4. Work with inheritance

Chapter 8. Lambda expressions in Java

8.1. The concept of lambda expression
8.2. Functional interfaces
8.3. Fields of visibility

Chapter 9. Execution threads.

9.1. Threads and the JVM
9.2. A lot of current
9.3. Interaction of flows.
9.4. Synchronization
9.5. Organization of competitive access

Chapter 10. Reflection. Internalization and logging.

10.1. Reflection
10.2. Internalization
10.3. Logging in

Chapter 11. SLE and unit testing

11.1. Version control systems
11.2. Modular testing

Chapter 12. Databases

12.1. SQL database
12.2. NoSQL database
12.3. Working with databases in Java

4. Educational materials and resources

Basic literature:

- K. Arnold, J. Gosling, D. Holmes. Java programming language. 3rd ed.-M: "Williams", 2010.-624 p.
- K.S. Horstmann, H. Cornell. Library of the professional. Java. Volume 1. Fundamentals, 7th ed. -M: "Williams", 2008.-896 p. K.S. Horstmann, H. Cornell. Library of the professional. Java 2. Volume 2. Subtleties of programming, 7th ed. - M: "Williams", 2007. - 1168 p.
- Fischer T. Java Pocket reference book. M. LLC "Vilms", 2008, 224 p.
- Karabyn P. The Java programming language. Creation of interactive applications for the Internet, M. "Informative book", 2001. - 224p.
- S.A. Kravchuk, Shonin V.A. Fundamentals of programming in Java. - K.: Norita-plus, 2007. - 280 p.

Additional literature:

- P. Naughton, G. Schildt Java. - St. Petersburg: BHV-Petersburg, 2007. - 1072 p.
- B. Eckel. The philosophy of Java. - St. Petersburg: Peter, 2006. – 880 p.
- Gudman S., Hidetniemy S. Introduction to the development and analysis of algorithms. - M.: Mir, 1981. - 368p.

- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Introduction to the algorithms. - The MIT Press, 2001. - 1180p.
- Library of the professional. Java 2. Volume 1. - M: "Williams" Publishing House, 2003. - 848 p.
- Library of the professional. Java 2. Volume 2. - M: Izdatelsky dom "Williams", 2002. - 1120 p.

Educational content

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

Names of sections and topics	Number of hours			
	In total	including		
		Lectures	Laboratory	SRS
1	2	3	5	6
Chapter 1. Features of the Java programming language		**2**		**4**
1.1. History of development and application		1		
1.2. The structure, functions and operation of the JVM		1		
Chapter 2. The lexical structure of the language		**4**	**2**	**4**
2.1. Main operators		0.5		
2.2. Types		0.5		
2.3. arrays		0.5	1	
2.4. Construction of classes		0.5	1	
2.5. Processing deadlines		0.5		
2.6. Regular expressions		0.5		
2.7. Congestion resolution		1		
Chapter 3. OOP in Java		**6**	**2**	**12**
3.1. Principles of OOP		1.5		
3.2. Abstract classes		0.5		
3.3. Interfaces		0.5		
3.4. Nested, local and anonymous classes.		1		
3.5. Access modifiers		0.5		
3.6. Types of relationships between classes and interfaces		0.5		
3.7. MVC architectural pattern		1	2	
3.8. Creating and using packages		0.5		
Chapter 4. Exceptions and generalized programming		**2**	**2**	**6**
4.1. Exceptional situations		1	1	

4.2. Generalized programming in the Java language (Generics).		1	1	
Chapter 5. Architecture and principles of software design		**2**		**6**
5.1. Software architecture		1		
5.2. Software design principles and patterns		1		
Chapter 6. Collections (Java Collections Framework)		**4**	**2**	**4**
6.1. Using collections		0.5		
6.2. Hierarchy of collections		0.5		
6.3. Realization of collections		3	2	
Chapter 7. I/O streams		**4**	**2**	**6**
7.1 Flow organization		1		
7.2. Serialization		1	1	
7.3. Work with files		1	1	
7.4. Work with inheritance		1		
Chapter 8. Lambda expressions in Java		**2**	**2**	**4**
8.1. The concept of lambda expression		1	2	2
8.2. Functional interfaces		0.5		
8.3. Fields of visibility		0.5		
Chapter 9. Execution threads.		**4**	**4**	**12**
9.1. Threads and the JVM		1		
9.2. A lot of current		0.5	2	
9.3. Interaction of flows.		0.5		
9.4. Synchronization		1		
9.5. Organization of competitive access		1	2	
Chapter 10. Reflection. Internalization and logging.		**2**	**2**	**4**
10.1. Reflection		1	0.5	
10.2. Internalization		0.5	0.5	
10.3. Logging in		0.5	1	
Chapter 11. SLE and unit testing		**2**		**6**
11.1. Version control systems		1		
11.2. Modular testing		1		
Chapter 12. Databases		**2**		**4**
12.1. SQL database		0.5		
12.2. NoSQL database		0.5		
12.3. Working with databases in Java		1		

Hours in general	**120**	**36**	**18**	**66**

6. Independent work of student

The main tasks of the cycle of practical classes (computer workshops): assimilation of knowledge by students on the main topics of the discipline and their consolidation; formation of students' practical skills and abilities.

No. z/p	Name of the subject of the lesson
1	Laboratory work 1. Work with loops, arrays and strings in Java.
2	Laboratory work 2. Working with classes.
3	Lab 3. Using OOP and the MVC pattern.
4	Laboratory work 4. Use of generalized programming.
5	Lab 5. I/O streams and serialization.
6	Laboratory work 6. Java Collections Framework
7	Laboratory work 7. Lambda expressions in Java.
8	Laboratory work 8. Multithreading.
9	Laboratory work 9. Organization of competitive access.
10	Laboratory work 10. Reflection, Logging, Internalization.

Policy and control

7. Policy of academic discipline (educational component)

The organization of the educational process and evaluation of learning outcomes are regulated by the Regulation on the Organization of the Educational Process at the National Technical University of Ukraine "Ihor Sikorskyi Kyiv Polytechnic Institute". According to which attendance at lectures and computer workshops is mandatory (except when there is a valid reason, for example, illness or permission of the dean's office).

Grading policy: each grade is assigned in accordance with the RSO developed by the teacher and announced to students in advance; if the student does not complete all the tasks (computer workshops) provided for in the working curriculum, he will not be admitted to the credit. If the student cannot attend the classes, he must complete the computer practicals on his own and make a defense in the class or, according to the schedule, at the teacher's consultation.

Policy of academic conduct and integrity: conflict situations should be openly discussed with the teacher, it is necessary to be mutually tolerant, respect the opinion of the other. Any form of dishonest work is unacceptable.

The student must complete all practical tasks independently using the appropriate methodical instructions, recommended literature and acquired knowledge and skills.

Inadmissible tips during the defense of computer workshops, on credit. Norms of academic ethics: discipline; observance of subordination; honesty; responsibility; working in the classroom with mobile phones turned off. During the protection of computer practicals, the student can use his own laptops. However, smartphones, tablets or computers should not be used during lectures and discussions of laboratory tasks. If you use your laptop or phone for audio or video recording, you must get permission from the instructor in advance.

Observance of the academic integrity of students and teachers is regulated by the code of honor of the National Technical University of Ukraine "Kyiv Polytechnic Institute", Regulations on the organization of the educational process at KPI named after Igor Sikorsky.

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

Semester certification is conducted in the form of credit. A 100-point rating system and a university rating scale are used to evaluate learning outcomes.

Tests are carried out as part of the current control of students' knowledge, according to the list of the main sections of the course. The main purpose of conducting control tests is to check students' knowledge.

The student's rating in the discipline consists of the points he receives for performing and defending computer practicals.

Weighted points for the computer workshop – 10 points.

The maximum number of points for all computer workshops is equal to 10 points · 10 = 100 points.

The maximum number of points for work is reduced if:

- the student does not know part of the theoretical material – 1..– 4 points;
- preliminary preparation for work was not completed –1..–4 points;
- untimely submission of laboratory work – 1..– 4 points;

The condition for the first attestation is to obtain at least 21 points and to complete three laboratory works. The condition for the second attestation is to obtain at least 50 points and to pass seven works.

A necessary condition for admission to the credit is the enrollment of all laboratory work, as well as a starting rating of at least 50 points.

In order for the student to receive the appropriate grades, his rating grade R is translated according to the table:

R	**rating**
95...100	Perfectly
85...94	Very good
75...84	Fine
65...74	Satisfactorily
60...64	Enough
50...59	Unsatisfactorily
R < 50 or other acceptance conditions are not met	Not allowed

9. Additional information on the discipline (educational component)

List of questions for assessment

1. Basic properties of the Java programming language.

2. The main components of the Java virtual machine. Their purpose.

3. How is platform independence achieved in Java technology?

4. The main differences between the Java language and C++.

5. Destruction of objects in Java. Conditions of destruction. Purpose of the finalize method.

6. Primitive data types. Conversion of primitive data types between themselves. Types of wrappers. Automatic casting of types.

7. Classification of data types in the Java language. Detail each type.

8. Describe the comparison methods used in the Java language.

9. Basic classes for processing and parsing lines in the Java programming language.

10. The main operators of program execution management.

11. Interrupt/return operators in the Java language.

12. Packages and Namespaces.

13. Class components in the Java language. Describe each component.

14. Inheritance in Java.

15. Enumeration types. Main properties of the type. Type composition.

16. Local and anonymous classes.

17. Access Modifiers. Specify the modifiers, objects of their application and their effect.

18. Modifier final. Specify the objects of application and describe the effect of the modifier.

19. Inheritance. The mechanism of the prohibition of inheritance.

20. Encapsulation and polymorphism.

21. Abstract classes.

22. Modifier static. Specify the objects of application and the effect of the modifier. Which keywords cannot be used inside static methods.

23. Purpose of the overload mechanism. To which software elements can overloading be applied?

24. Interfaces. Mechanism of inheritance and interfaces.

25. Nested classes. Types of nested classes and their differences.

26. Polymorphism in Java. Which modifiers prohibit overlapping methods.

27. Handling of exceptional situations in the Java programming language.

28. The syntax of the exception handling block in Java. What are the possible options for using sections of the exception handling block?

29. Compiler-Checked Exceptions. What rules must be followed if a compiler-checked exception is triggered in the body of a method?

30. Classification of exception situations in the Java language. Hierarchy of the main classes of exceptional situations.

31. Main types of exceptional situations. Hierarchy of exceptional situations.

32. Support for generalized programming in the Java language.

33. The main types of I/O streams in Java and their purpose. Basic classes for handling I/O streams.

34. State and describe the classes of I/O streams that involve additional data processing or provide convenient read/write methods.

35. Basic collection classes in the Java programming language. Assignment of different types of collections.

36. The main algorithms included in the Collections Framework.

37. Methods of sorting collections.

38. Basic implementations of sets. Purpose and differences.

39. Basic implementations of lists. Purpose and differences.

40. Basic implementations of association arrays (Map). Purpose and differences.

41. Ensuring data integrity when multiple threads work on the same data simultaneously.

42. Scheduler of execution threads in Java. Appointment of a scheduler. Algorithm applied by the scheduler.

43. Algorithm for delimiting access of execution threads to data using the wait and notify methods.

44. Algorithm for creating and starting the execution stream.

45. The synchronized operator. Appointment. To which elements does the operator apply?

46. How is thread interaction implemented in Java concurrent?

47. What blocks are implemented in Java concurrent. What are the advantages of data blocking over synchronized?

Working program of the academic discipline (syllabus):

FoldedArt. Lecturer, PhD, Serhii Petrovych Orlenko

Approvedby the Department of Information Systems and Technologies (protocol No. 13 dated June 15, 2022)

AgreedMethodical commission of the faculty (protocol No. 11 dated 07.07.2022)