Syllabus
Course Program

Java basic programming course

Specialty
121 – Software Engineering
122 – Computer Science

Institute
Institute of Computer Science and Information Technology

Educational program
Software Engineering
Computer Science and Intelligent Systems

Department
Software Engineering and Management Intelligent Technologies (321)

Level of education
Bachelor's level

Course type
Special (professional), Elective

Semester
4

Language of instruction
English, Ukrainian

Lecturers and course developers

Lev Ivanov

lev.ivanov@khpi.edu.ua
Senior Lecturer of Software Engineering and Management Intelligent Technologies Department

Google Scholar: https://scholar.google.com/citations?user=B8fggLEAAAAJ
More about the lecturer on the department's website

General information

Summary

The task of the discipline is to acquire the necessary level of knowledge about the components of the object-oriented model and the syntax of the Java object-oriented programming language, as well as the application of the object-oriented approach, methods and techniques for creating programs using the Java programming language.

Course objectives and goals

Mastering the syntax of the Java object-oriented programming language, as well as the application of methods and techniques for creating programs using the tools of the Java language and Java platform.

Format of classes

Lectures, laboratory classes, consultations, self-study. Final control in the form of a credit.

Competencies

121 - Software Engineering
GC01. Ability to abstract thinking, analysis and synthesis.
GC02. Ability to apply knowledge in practical situations.
GC06. Ability to search, process and analyze information from various sources.
PC15. Ability to develop architectures, modules and components of software systems.
PC19. Knowledge of information data models, the ability to create software for data storage, retrieval and processing.

PC23. Ability to implement phases and iterations of the life cycle of software systems and information technology based on appropriate models and approaches to software development.
PC25. Ability to reasonably select and master software development and maintenance tools.
PC26. Ability to algorithmic and logical thinking..
122 - Computer Science and Intelligent Systems
GC1. Ability to abstract thinking, analysis and synthesis.
GC2. Ability to apply knowledge in practical situationsGC6. Ability to learn and master modern knowledge.
GC7. Ability to search, process and analyze information from various sources.
PC8. Ability to design and develop software using different programming paradigms: generalized, object-oriented, functional, logical, with appropriate models, methods and algorithms of calculations, data structures and management mechanisms.

Learning outcomes

121 - Software Engineering
PO03. Know the basic processes, phases and iterations of the software life cycle
PO04. Know and apply professional standards and other regulatory documents in the field of software engineering
PO08. Be able to develop a human-machine interface
PO14. Put into practice the tools of domain analysis, design, testing, visualization, measurement and documentation of software.
122 - Computer Science and Intelligent Systems
PLO1. Apply knowledge of the fundamental forms and laws of abstract-logical thinking, the basics of the methodology of scientific knowledge, forms and methods of extraction, analysis, processing, and synthesis of information in the subject area of computer science.
PLO5. Design, develop and analyze algorithms for solving computational and logical problems, evaluate the efficiency and complexity of algorithms based on the use of formal models of algorithms and computational functions.
PLO9. Develop software models of subject areas, choose a programming paradigm from the standpoint of convenience and quality of its application to implement methods and algorithms that solve problems in the computer science field.To use the methodology of system analysis of objects, processes and systems for the tasks of analysis, forecasting, management and design of dynamic processes in macroeconomic, technical, technological and financial objects.

Student workload

The total volume of the course is 150 hours (5 ECTS credits): lectures - 32 hours, laboratory classes - 32 hours, self-study - 86 hours.

Course prerequisites

Fundamentals of programming
Algorithmization and programming
Object-oriented programming

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

Teaching and learning methods:
interactive lectures with presentations, discussions, laboratory classes, teamwork, case method, student feedback, problem-based learning.
Forms of assessment:
written individual assignments for laboratory work (CAS), assessment of knowledge in laboratory classes (CAS), express surveys (CAS), online tests (CAS), final/semester control in the form of a semester test, according to the schedule of the educational process (FAS).

Program of the course

Topics of the lectures

Topic 1: Java architecture and syntax. Data types. Expressions, operations and statements. Packages
Java programming language and Java platform. The general structure of the Java program. Identifiers, keywords and reserved words. Comments. Data types. Expressions, operations and statements. Packages and functions. Console input and output
Topic 2 Reference types. Arrays. Classes
References. Arrays. Arrays as parameters and result of functions. Standard functions for working with arrays. Definition of classes. Encapsulation. Using standard classes. Strings. Wrapper classes. Starting Java applications from the command line
Topic 3 Using inheritance and polymorphism in Java. Interfaces
Composition of classes. Inheritance. Sealed classes. Abstracts. Polymorphism. Interfaces. Comparison of objects. Nested classes. Default implementation of interface methods. Working with functional interfaces. Object cloning, equivalence checking, and hash codes.
Topic 4 Working with records, enumerations, generics and collections
Records. Enumerations. Generics. Container classes and interfaces. Working with lists. Work with queues and stacks. Static methods of the Collections class. Algorithms. Working with sets and associative arrays. Internal organization of sets and associative containers. Creating custom containers
Topic 5 Working with exceptions and I/O streams
Exception handling. Input and output streams. Character streams. Working with byte streams. Binary serialization of objects. Working with archives. Using the XML language.

Topics of the workshops

Workshops are not provided within the discipline.

Topics of the laboratory classes

Topic 1: Java architecture and syntax
Topic 2 Working with reference types
Topic 3 Using inheritance and polymorphism in Java
Topic 4 Working with generics and collections
Topic 5 Working with exceptions and files in Java

Self-study

Individual assignments are not provided in the curriculum.
Students are recommended with additional materials (videos, articles) for self-study and processing.

Course materials and recommended reading

Key literature

1. Bloch J. Effective Java: 3rd Edition, Addison Wesley, 2017, 412 p.
2. Schildt H. Java: A Beginner's Guide: 8th Edition, McGraw-Hill Education, 2018, 684 p.
3. Schildt H. Java: The Complete Reference: 11th Edition, McGraw-Hill Education, 2018, 1208 p.
4. Horstmann C. S. Core Java Volume I – Fundamentals: 11th Edition, Prentice Hall 2018, 889 p.
5. Eckel B. Thinking in Java 4th Edition: Pearson, 2006, 1150 p.
6. Deitel P., Deitel H. Java How to Program, Early Objects: 11th Edition, Pearson, 2017, 1296 p..

Additional literature

1.Horstmann C. S. Core Java SE 9 for the Impatient: 2nd Edition Addison-Wesley Professional, 2017, 576 p.
2. Deitel P., Deitel H. Java How To Program, Late Objects: 11th Edition, Pearson, 2017, 1248 p.
3. Ратушняк Т. В. Програмування мовою JAVA: практикум: навчальний посібник. Державна фіскальна служба України, Університет державної фіскальної служби України. – Ірпінь, 2017. – 212 с.

Assessment and grading

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

100% Final assessment as a result of Final test (30%) and Continuous assessment (70%).
40% Final test
60% Continuous assessment:
Laboratory work №1 (12%)
Laboratory work №2 (12%)
Laboratory work №3 (12%)
Laboratory work №4 (12%)
Laboratory work №5 (12%)

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 08.06.2023 Head of the department
Ihor HAMAIUN

08.06.2023 Guarantors of the educational programs
Andrii KOPP
Uliya LITVINOVA