Syllabus
Course Program

Java advanced 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
5

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 for students to acquire the necessary level of knowledge about the standard capabilities of the Java language and platform for working with text data, the file system, user interaction mechanisms, graphic tools, databases, creating multi-threaded applications, as well as effective work with reflection mechanisms.

Course objectives and goals

Provide students with practical skills in the use of Java and tools of the Java Standard Edition platform for solving data processing problems, effective interaction with the environment and the use of multithreading.

Format of classes

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

Competencies

121 - Software Engineering
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.
PC20. Ability to apply fundamental and interdisciplinary knowledge to successfully solve software engineering problems.

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.
122 - Computer Science and Intelligent Systems
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.
PC3. Ability to think logically, build logical conclusions, use formal languages and models of algorithmic calculations, design, develop and analyze algorithms, evaluate their efficiency and complexity, solvability and insolvability of algorithmic problems for adequate modelling of subject areas and creation of software and information systems.
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
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.
PO13. Know and apply methods of algorithm development, software design and data and knowledge structures.
PO23. Be able to document and present the results of software development.
122 - Computer Science and Intelligent Systems
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.
PLO16. Perform parallel and distributed computing, apply numerical methods and algorithms for parallel structures, parallel programming languages in the development and operation of parallel and distributed software.

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

Object-oriented programming
Java advanced programming course

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 SE Platform. Working with Dates and Text. Localization
General features of the Java SE platform. Working with dates and times. Working with text in Java. Localization. Using the java.text package. Regular expressions. Using BigInteger and BigDecimal types. Testing in Java. Using JUnit.
Topic 2 Working with the File system. Advanced features of working with files
Use of build automation tools. Working with the file system. Using java.nio tools for reading and writing Data. Using the Java 8 Stream API. Working with JSON files. Serialization into XML and JSON files using XStream tools
Topic 3 Event-driven programming. Use of JavaFX tools.
Using Java to create GUI applications. Overview of JavaFX platform. The theoretical basis of creating JavaFX applications. Working with JavaFX visual components. Working with tabular data in JavaFX. Visual design of GUI applications.
Topic 4 Reflection. Class loaders. Creation of database applications
Using RTTI. Using class loaders. Reflection. Creating and using annotations. Using annotations in Java libraries and frameworks. Java tools for working with databases.
Topic 5 Metaprogramming. Working with threads
Scripting tools. Dynamic code generation. Working with threads. Synchronization. Use of thread-safe collections.

Topics of the workshops

Workshops are not provided within the discipline.

Topics of the laboratory classes

Topic 1 Working with dates and text. Localization
Topic 2 Advanced features of working with files
Topic 3 Development of GUI applications
Topic 4 Reflection. Creation of database applications
Topic 5 Metaprogramming. Multithreading

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 Horstmann C. S. Core Java, Volume II - Advanced Features (Core Series): 11th Edition, Pearson, 2019, 1040 p.

Additional literature

1 Kishori Sharan. Beginning Java 8 Fundamentals. Language Syntax, Arrays, Data Types, Objects, and Regular Expressions. – Apress, 2014. – 828.
2 Java 8 Programming: Black Book by DT Editorial Services. - Dreamtech Press, 2015. – 1052 p.
3 Urma Raoul-Gabriel, Fusco Mario, Mycroft Alan. Java 8 in Action: Lambdas, streams, and functional-style programming. –Manning Publications Co, 2015 – 497 p.

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