Syllabus
Course Program

Object-Oriented Programming

Specialty
121 – Software Engineering

Institute
Institute of Computer Science and Information Technology

Educational program
Software Engineering

Department
Software Engineering and Management Intelligent Technologies (321)

Level of education
Bachelor's level

Course type
Special (professional), Mandatory

Semester
3

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

Course objectives and goals

Acquisition of the necessary knowledge on mastering modern technologies of analysis, design and object-oriented development of software systems.

Format of classes

Lectures, laboratory classes, self-study, consultations. Final control in the form of an exam.

Competencies

К01. Ability to think abstractly, analyze and synthesize.
К02. Ability to apply knowledge in practical situations.
К05. Ability to learn and master modern knowledge.
К06. Ability to search, process and analyze information from various sources.
К07. Ability to work in a team.
К14. Ability to participate in software design, including modeling (formal description) of its structure, behavior and processes of functioning.
К15. Ability to develop architectures, modules and components of software systems.

К22. Ability to accumulate, process and systematize professional knowledge of software development and maintenance and recognize the importance of lifelong learning.
К23. Ability to implement phases and iterations of the life cycle of software systems and information technologies based on appropriate software development models and approaches.
К25. Ability to reasonably choose and master the tools for software development and maintenance.

Learning outcomes

PLO01. Analyze, purposefully search and select information and reference resources and knowledge necessary for solving professional problems, taking into account modern achievements of science and technology.
PLO04. To know and apply professional standards and other regulatory documents in the field of software engineering.
PLO05. To know and apply relevant mathematical concepts, methods of domain, system and object-oriented analysis and mathematical modeling for software development.
PLO07. To know and apply in practice the fundamental concepts, paradigms and basic principles of functioning of language, tools and computing tools of software engineering.
PLO08. Be able to develop a human-machine interface.
PLO13. Know and apply methods of developing algorithms, designing software and data structures and knowledge.
PLO14. Apply in practice software tools for domain analysis, design, testing, visualization, measurement and documentation of software.
PLO16. Have the skills of team development, coordination, design and production of all types of program documentation.
PLO17. Be able to apply methods of component software development.
PLO23. Be able to document and present the results of software development.

Student workload

The total volume of the course is 120 hours (4 ECTS credits): lectures – 32 hours, laboratory classes – 32 hours, self-study – 56 hours.

Course prerequisites

Fundamentals of programming
Theory of algorithms
Introductory practice at the "Innovation Campus"

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 exam, according to the schedule of the educational process (FAS).

Program of the course

Topics of the lectures

Topic 1: Client OOP Bases. The Base Syntax of C# Programming Language. Reference Types
Programming paradigms. Origins and Benefits of Object-Oriented Approach. Components of an Object-Oriented Methodology.. .NET Platform and C# Programming Language. Bases of C# Language Syntax. Namespaces. Classes. Properties. Working with Arrays. Indexers. Strings. Console Output and Input.

Topic 2. Additional features of classes. Inheritance and polymorphism. Structures and enumerations
Object initializers. Static classes. Operator overloading. Inheritance. Polymorphism. Interfaces. Anonymous types and records. Structures and enumerations. Tuples. Methods that extend existing types. Nested types. Pattern matching.
Topic 3. Working with exceptions and files. Generics and collections
Exception handling. Initial information about working with files. Working with XML documents. Using serialization. Creation and use of generics. Standard generic classes and methods. Creating custom container types. Creation and use of class libraries.
Topic 4. Partial classes. Event-driven programming. Functional and declarative programming
Dynamic type. Partial classes and methods. Working with the file system. delegates. events. concepts of Functional and declarative programming. The implementation of functional and declarative programming in C#. GUI Application development using Windows Presentation Foundation technology.
Topic 5: Applying design patterns in C# applications
Design patterns summary. Classification of design patterns. Examples of creational patterns. Examples of structural patterns. Examples of behavioral patterns. Solving design problems using patterns.

Topics of the workshops

Workshops are not provided within the discipline.

Topics of the laboratory classes

Topic 1. OOP Bases. The Base Syntax of C# Programming Language
Topic 2. Inheritance and Polymorphism
Topic 3. Generic Programming
Topic 4. Event-Driven Programming. Functional and Declarative Programming
Topic 5. Design Patterns

Self-study

The curriculum includes the completion of coursework (CW). At the beginning of the semester, students choose the topics of the course work from the list or propose their own topics and agree them with the teacher. The CW is completed during the semester and is defended during the test week or examination session.
Students are recommended with additional materials for self-study and processing.

Course materials and recommended reading

Key literature

1. Troelsen A. Japikse P. Pro C# 9 with .NET 5: Foundational Principles and Practices in Programming: 10th edition, Apress, 2021, 1411 p.
2. Albahari J. C# 9.0 in a Nutshell: The Definitive Reference: 1st Edition, O'Reilly Media, 2021, 1060 p. Nagel C. Professional C# 7 and .NET Core 2.0: 7th Edition, Wrox, 2018, 1440 p.
3. Booch G. Rumbaugh J, Jacobson I. The Unified Modeling Language User Guide (Object Technology Series): 2nd Edition, Addison-Wesley Professional, 2005, 494 p.
4. Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software: 1st Edition, Addison-Wesley Professional, 1994, 540 p.

Additional literature

1. . Farrell J. Microsoft Visual C#: An Introduction to Object-Oriented Programming: 007 Edition, Joyce Cengage Learning; 007 edition, 2017, 784 p.
2. Cardoso A. F. M. Implementing Design Patterns in C# and .NET 5: Build Scalable, Fast, and Reliable .NET Applications Using the Most Common Design Patterns. BPB Publications, 2021, 290 p.
3. Weisfeld M. Object-Oriented Thought Process, The Developer's Library: 5th Edition, 2019, 240 p.

Assessment and grading

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

100% of the final grade consists of the results of the assessment in the form of an exam (40%) and current assessment (60%):
- 5 laboratory works (5% each);
- coursework (35%).

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	11.04.2023	Head of the department Ihor HAMAIUN
	08.06.2023	Guarantor of the educational program Uliya LITVINOVA