

ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ

Тип дисципліни	Обов'язкова
Рівень вищої освіти (бакалаврський)	Перший
Мова викладання	Українська, Англійська
Семестри	другий, третій
Обсяг кредитів ЄКТС	12
Форма здобуття освіти	Очна (Денна)

Результати навчання. Відповідно до Стандарту вищої освіти із зазначеної спеціальності та освітньої програми дисципліна має забезпечити:

компетентності: Здатність розв'язувати складні спеціалізовані завдання або практичні проблеми інженерії програмного забезпечення, що характеризуються комплексністю та невизначеністю умов, із застосуванням теорій та методів інформаційних технологій. Здатність до абстрактного мислення, аналізу та синтезу; Здатність застосовувати знання у практичних ситуаціях. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення. Здатність до алгоритмічного та логічного мислення.

програмні результати навчання: Аналізувати, цілеспрямовано шукати і вибрати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки; знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення; уміти вибирати та використовувати відповідну задачі методологію створення програмного забезпечення; знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення; мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення

Зміст навчальної дисципліни. Об'єктно-орієнтований підхід при конструюванні програмного забезпечення. Модульність в програмуванні. Методи доменного, системного, об'єктно-орієнтованого аналізу та математичного моделювання в інженерії програмного забезпечення. Поняття класів та об'єктів в ООП. Абстрактні класи. Структури та перерахування. Концепція наслідування. Інтерфейси. Множинне наслідування. Делегати. Функціональний тип даних. Події. Універсальність. Класи з родовими параметрами. Декларативність. Коректність та стійкість програмних систем.

Запланована навчальна діяльність: лекції – 70 год., лабораторні заняття – 87 год., самостійна робота – 263 год, разом – 360 год.

Форми (методи) навчання: лекції (з використанням методів проблемного навчання і візуалізації); лабораторні заняття (з використанням методів комп'ютерного моделювання, практикумів), самостійна робота.

Форми оцінювання результатів навчання: захист лабораторних робіт, усне опитування, тестування.

Форма семестрового контролю: залік, іспит.

Навчальні ресурси:

1. Порівняльний аналіз парадигм програмування при розробці програмних систем на основі штучного інтелекту/ ВМ Кривий, ОМ Яшина, ГІ Радельчук, СМ Лисенко// «Measuring and computing devices in technological processes» 2021, Issue 1. – p.62-65
2. Дібрівний О.А. Вступ до об'єктно орієнтованого програмування С#: Навчальний посібник/ О.А.Дібрівний, В.В. Гребенюк– К.: Державний університет телекомунікацій, 2018, - 190 с.

3. Васильєв О.М. Програмування в PYTHON. Теорія і практика/ О.М.Васильєв, - К. Ліра-К, 2023, - 462 с.
4. David Flanagan JavaScript: The Definitive Guide: Master the World's Most-Used Programming Language/Flanagan David - O'Reilly Media 2020. - 704 p.
5. Joel Murach ASP.NET Core MVC/Murach Joel, Dalamater Mary - Mike Murach & Associates 2020. - 780 p.
6. Mark Seemann. Code That Fits in Your Head: Heuristics for Software Engineering/ Seemann Mark - Addison-Wesley Professional 2021. - 406 p.
7. Модульне середовище для навчання MOODLE. Доступ до ресурсу : <https://msn.khmnu.edu.ua>
8. Електронна бібліотека університету. Доступ до ресурсу : http://lib.khmnu.edu.ua/asp/php_f/page_lib.php.
9. Репозитарій ХНУ. Доступ до ресурсу: <http://elar.khmnu.edu.ua/jspui/?locate=uk>

Викладачі: доктор тех. наук, професор Лисенко С.М., канд. техн. наук, доцент Форкун Ю.В.

OBJECT-ORIENTED PROGRAMMING

Type of discipline	Mandatory
Level of higher education	First (bachelor's)
Language of instruction	Ukrainian, English
Semesters	second, third
ECTS credits	12
Course study mode	Full-time (Daytime)

Learning outcomes. According to the Standard of Higher Education in the specified specialty and educational program, the discipline must provide:

Competences: Ability to solve complex, specialised tasks or practical problems in software engineering, characterised by complexity and uncertainty of conditions, using information technology theories and methods. Ability to abstract thinking, analysis, and synthesis. Ability to apply knowledge in practical situations. Ability to reasonably choose and master the toolkit for software development and maintenance. Ability for algorithmic and logical thinking.

Programmatic Learning Outcomes: To analyse, purposefully search for, and select the necessary information, reference resources, and knowledge for solving professional tasks, considering modern scientific and technical achievements. To understand and apply relevant mathematical concepts, domain and system methods, object-oriented analysis, and mathematical modelling for software development. To select and utilise a software development methodology appropriate for the task. To understand and apply in practice the fundamental concepts, paradigms, and basic principles of functioning linguistic, instrumental, and computational tools of software engineering. To make informed decisions when choosing programming languages and development technologies to address the tasks of creating and maintaining software.

The content of the discipline. Object-oriented approach to software design. Modularity in programming. Methods of domain, system, object-oriented analysis and mathematical modeling in software engineering. The concept of classes and objects in OOP. Abstract classes. Structures and enumerations. The concept of inheritance. Interfaces. Multiple inheritance. Delegates. Functional data type. Events. Versatility. Classes with generic parameters. Declarativeness. Correctness and stability of software systems.

Planned educational activities: lectures – 70 hours, laboratory classes – 87 hours, individual work – 263 hours, total – 360 hours.

Forms (methods) of education: lectures (using the methods of problem-based learning and visualization); laboratory classes (using computer modeling methods, workshops), independent work.

Forms of assessment of learning outcomes: defense of laboratory work, oral questioning, testing.

Form of semester control: credit, exam.

Learning Resources:

1. Порівняльний аналіз парадигм програмування при розробці програмних систем на основі штучного інтелекту/ ВМ Кривий, ОМ Яшина, ГІ Радельчук, СМ Лисенко// «Measuring and computing devices in technological processes» 2021, Issue 1. – p.62-65

2. Дібрівний О.А. Вступ до об'єктно орієнтованого програмування С#: Навчальний посібник/ О.А.Дібрівний, В.В. Гребенюк– К.: Державний університет телекомунікацій, 2018, - 190 с.

3. Васильєв О.М. Програмування в PYTHON. Теорія і практика/ О.М.Васильєв, - К. Ліпа-К, 2023, - 462 с.

4. David Flanagan JavaScript: The Definitive Guide: Master the World's Most-Used Programming Language/Flanagan David - O'Reilly Media 2020. - 704 p.

5. Joel Murach ASP.NET Core MVC/Murach Joel, Dalamater Mary - Mike Murach & Associates 2020. - 780 p.

6. Mark Seemann. Code That Fits in Your Head: Heuristics for Software Engineering/ Seemann Mark - Addison-Wesley Professional 2021. - 406 p.

7. Modular learning environment MOODLE. Access to the resource:
<https://msn.khmnu.edu.ua>

8. Electronic Library of the University. Access to the resource :
http://lib.khmnu.edu.ua/asp/php_f/page_lib.php.

9. KhNU Repository. Access to the resource: <http://elar.khmnu.edu.ua/jspui/?locate=uk>

Lecturers: Doctor of Technical Sciences, Professor Lysenko S.M., Cand. Tech. Doctor of Physical Sciences, Associate Professor Forkun Y.V.