

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Львівський національний університет імені Івана Франка**  
**Факультет прикладної математики та інформатики**  
**Кафедра прикладної математики**

**Затверджено**

На засіданні  
кафедри прикладної математики  
факультету прикладної математики та  
інформатики  
Львівського національного університету  
імені Івана Франка  
(протокол № 1 від 31 серпня 2023 р.)



Завідувач кафедри

 Юрій ЯЩУК

**Силабус з навчальної дисципліни**  
**“Проектування об’єктно-зорієнтованих програм”,**  
**що викладається в межах ОПП Прикладна математика**  
**другого (магістерського) рівня вищої освіти для здобувачів з**  
**спеціальності 113 – прикладна математика**

Львів 2023 р.

<b>Назва дисципліни</b>	<b>Проектування об'єктно-зорієнтованих програм</b>
<b>Адреса викладання дисципліни</b>	Головний корпус ЛНУ ім. І. Франка м. Львів, вул. Університетська 1
<b>Факультет та кафедра, за якою закріплена дисципліна</b>	Факультет прикладної математики та інформатики Кафедра прикладної математики
<b>Галузь знань, шифр та назва спеціальності</b>	11 – математика та статистика 113 – прикладна математика
<b>Викладачі дисципліни</b>	Дяконюк Лілія Миколаївна, доцент кафедри прикладної математики
<b>Контактна інформація викладачів</b>	lilii.diakoniuk@lnu.edu.ua, <a href="https://ami.lnu.edu.ua/employee/diakoniuk">https://ami.lnu.edu.ua/employee/diakoniuk</a> Головний корпус ЛНУ ім. І. Франка, каб. 378. м. Львів, вул. Університетська, 1
<b>Консультації з питань навчання по дисципліні відбуваються</b>	Консультації в день проведення лекцій/лабораторних занять (за попередньою домовленістю).
<b>Сторінка курсу</b>	<a href="https://ami.lnu.edu.ua/course/objektno-zorijentovane-prohramuvannya-prykladna-matematyka">https://ami.lnu.edu.ua/course/objektno-zorijentovane-prohramuvannya-prykladna-matematyka</a>
<b>Інформація про дисципліну</b>	Дисципліна “Проектування об'єктно-зорієнтованих програм” є базовою дисципліною з спеціальності 113 – прикладна математика для освітньої програми магістрів Прикладна математика, яка викладається в 2-му семестрі в обсязі 4,5 кредити (за Європейською Кредитно-Трансферною Системою ECTS).
<b>Коротка анотація дисципліни</b>	Курс передбачає вивчення основних принципів проектування програмного забезпечення для великих програмних комплексів з використанням об'єктно-зорієнтованої парадигми.
<b>Мета та цілі дисципліни</b>	Метою вивчення вибіркової дисципліни “Проектування об'єктно-зорієнтованих програм” є освоєння студентами основних принципів розробки програмних проектів на об'єктно-зорієнтованій мові, створення архітектури програмного забезпечення з використанням шаблонів проектування, яка б давала можливість ефективно вносити зміни в структуру, уникати повторів коду, а також ускладнень, економно використовувати ресурси системи та забезпечувати надійність програмного забезпечення.
<b>Література для вивчення дисципліни</b>	<p style="text-align: center;"><b>Основна література</b></p> <ol style="list-style-type: none"> <li>1. Erich Gamma. Design Patterns: Elements of Reusable Object-Oriented Software / Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides (Gamma of Gang of Four).</li> <li>2. E. Freeman. Head First Design Patterns 2E. / E. Freeman, E. Robson. - O'Reilly Media -2020. - 641p.</li> <li>3. Jeffrey Chilberto. Hands-On Design Patterns with C# and .NET Core. / Jeffrey Chilberto, Gaurav Arora. - ISBN-139781789133646. – Packt Publishing. – 2019. – 410 p.3.</li> <li>4. A. Shvetc. Design patterns. – Refactoring.guru, 2021. - 410 p.</li> </ol>

	<p style="text-align: center;"><b>Інформаційні ресурси</b></p> <ul style="list-style-type: none"> <li>• <a href="http://www.dofactory.com">http://www.dofactory.com</a></li> <li>• <a href="http://sourcemaking.com/design_patterns">http://sourcemaking.com/design_patterns</a></li> <li>• <a href="http://www.citforum.ru/SE/project/pattern/">http://www.citforum.ru/SE/project/pattern/</a></li> <li>• <a href="https://refactoring.guru/uk/design-patterns">https://refactoring.guru/uk/design-patterns</a></li> <li>• <a href="https://echo.lviv.ua/dev/5394">https://echo.lviv.ua/dev/5394</a></li> <li>• <a href="https://www.journaldev.com/1827/java-design-patterns-example-tutorial">https://www.journaldev.com/1827/java-design-patterns-example-tutorial</a></li> <li>• <a href="https://www.dofactory.com/net/design-patterns">https://www.dofactory.com/net/design-patterns</a></li> <li>• <a href="https://dotnettutorials.net/course/dot-net-design-patterns/">https://dotnettutorials.net/course/dot-net-design-patterns/</a></li> <li>• <a href="https://www.freecodecamp.org/news/solid-principles-explained-in-plain-english/">https://www.freecodecamp.org/news/solid-principles-explained-in-plain-english/</a></li> <li>• <a href="https://medium.com/backticks-tildes/the-s-o-l-i-d-principles-in-pictures-b34ce2f1e898">https://medium.com/backticks-tildes/the-s-o-l-i-d-principles-in-pictures-b34ce2f1e898</a></li> </ul>
<b>Обсяг курсу</b>	Загальний обсяг: 135 годин. Аудиторних занять: 64 год., з них 16 год. лекцій та 32 години лабораторних робіт. Самостійної роботи: 87 год.
<b>Очікувані результати навчання</b>	<p>Після завершення цього курсу студент буде</p> <p><b>Знати:</b></p> <ul style="list-style-type: none"> <li>- Основні парадигми ОЗП.</li> <li>- Переваги і недоліки принципів Solid.</li> <li>- Мову UML.</li> <li>- Шаблони проектування.</li> <li>- Пастки програмування.</li> </ul> <p><b>Вміти:</b></p> <ul style="list-style-type: none"> <li>- Будувати різні види UMLдіаграм.</li> <li>- Обґрунтовувати вибрану схему архітектури програмного забезпечення.</li> <li>- Використовувати шаблони проектування.</li> <li>- Уникати стандартних пасток в проектуванні, які приводять до проблемних ситуацій.</li> <li>- Реалізовувати шаблони на обраній мові, з врахуванням специфіки реалізації.</li> </ul>
<b>Ключові слова</b>	Шаблони проектування, UML-діаграми, архітектура програмного забезпечення.
<b>Формат курсу</b>	Очний Проведення лекцій, лабораторних робіт і консультацій.
<b>Теми</b>	Подано нижче у таблиці Схема курсу “Проектування об’єктно-зорієнтованих програм”.
<b>Підсумковий контроль, форма</b>	Залік
<b>Пререквізити</b>	Для вивчення курсу студенти потребують базових знань з курсів: <ul style="list-style-type: none"> <li>- Об’єктно-орієнтоване програмування</li> <li>- Програмне забезпечення.</li> <li>- Алгоритми обчислювальних процесів</li> <li>- Архітектури комп’ютера</li> <li>- Бази даних</li> </ul>
<b>Навчальні методи та техніки, які будуть використовуватися під час викладання курсу</b>	Презентації, лекції Індивідуальні завдання Груповий та індивідуальний проект
<b>Необхідне об-</b>	Комп’ютер із програмним забезпеченням, інтернет зв’язок.

<p><b>ладнання</b></p> <p><b>Критерії оцінювання (окремо для кожного виду навчальної діяльності)</b></p>	<p>Оцінювання проводиться за 100-бальною шкалою. Бали нараховуються за наступним співвідношенням:  індивідуальні завдання: 40% семестрової оцінки; максимальна кількість балів – 40.  розробка проєктів: 60% семестрової оцінки; максимальна кількість балів – 60.  Підсумкова максимальна кількість балів 100.</p> <p><b>Письмові роботи:</b> Очікується, що студенти оформлять діаграми та створять програми на одній з запропонованих мов (c#, java, c++) та захистять виконану роботу. Груповий проєкт передбачає публічний захист з презентацією та обґрунтуванням обраної архітектури, демонстрацією роботи програми.</p> <p><b>Академічна доброчесність:</b> Очікується, що програмні роботи студентів будуть створені ними самостійно. Виявлення ознак академічної недоброчесності в письмовій роботі студента є підставою для її незарахування викладачем, незалежно від масштабів плагіату чи обману. Відвідання занять є важливою складовою навчання. Очікується, що всі студенти відвідають лекційні та лабораторні заняття курсу. Для студентів передбачені терміни здачі індивідуальних завдань. Невчасний захист індивідуальних завдань є підставою до зменшення отриманих балів. Для отримання залікового підсумкового контролю мусять бути виконані весь пакет завдань, незалежно від кількості отриманих балів за попередні завдання.</p> <p><b>Література.</b> Студентам буде наданий рекомендований список літератури та інформаційних джерел та проведено їх аналіз. Водночас, заохочуватиметься використання також й іншої літератури та джерел, яких немає серед рекомендованих.</p> <p><b>Політика виставлення балів.</b> Враховуються бали набрані за захист індивідуальних завдань, виконання домашніх робіт, а також за активність роботи на заняттях.</p> <p>Жодні форми порушення академічної доброчесності не толеруються.</p>
<p><b>Опитування</b></p>	<p>Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенню курсу.</p>

## Схема курсу “Проектування об’єктно-зорієнтованих програм”

Тиж.	Тема, план, короткі тези	Форма діяльності (заняття)	Література, Ресурси в інтернеті	Завдання, год.	Термін виконання
1	<b>Тема 1.</b> Огляд курсу. Особливості розробки великих програмних комплексів.	Лекція (2 год.)	[1,3,4,6,7]	Опрацювання лекційного матеріалу (2 год.)	1 тиждень
2	<b>Тема 2.</b> Огляд парадигм програмування з визначенням місця ОЗП.	Лекція (2 год.)	[1-5,7,11,12]	Опрацювання лекційного матеріалу (2 год.)	1 тиждень
	<b>Тема 1-2.</b> Розв’язування задач з демонстрацією переваг та недоліків різних підходів. Обговорення особливостей групових проєктів. Розподіл підзадач для виконання.	Лабораторне (4 год)	[1-5,7,11,12]	Опрацювання матеріалу (4 год.)	2 тижні
3	<b>Тема 3.</b> Мова UML. Типи діаграм та особливості їх використання.	Лекція (2 год.)	[5,7,8]	Опрацювання лекційного матеріалу (4 год.)	2 тижні
	<b>Тема 3.</b> Побудова діаграм взаємозв’язків між класами, діаграми послідовностей викликів та діаграми прецедентів	Лабораторне (4 год)	[1-5,7,11,12]	Створення діаграм для групового проєкту (8 год)	2 тижні
4	<b>Тема 4. Класифікація шаблонів проєктування.</b> Принципи SOLID. Основне призначення шаблонів проєктування.	Лекція (2 год.)	[5,8]	Опрацювання лекційного матеріалу (4 год)	2 тижні
	<b>Тема 4.</b> Розробка програм з використанням принципів SOLID. Огляд можливих суперечностей та наслідків порушень.	Лабораторне (4 год.)	[1-5,7,11,12]	Виконання домашнього завдання 1 (4 год.) Створення класів для групового проєкту (6 год)	2 тижні

5	<b>Тема 5.</b> Породжуючі шаблони.	Лекція (2 год.)	[5,7,8]	Опрацювання лекційного матеріалу (4 год.)	2 тижні
6	<b>Тема 6.</b> Реалізація шаблонів Одинак, Абстрактна фабрика, Фабричний метод	Лабораторне (4 год.)	[5,8,9-12]	Виконання домашнього завдання, (4 год.) Робота над груповим проєктом (6 год.)	2 тижні
7	<b>Тема 7.</b> Поведінкові шаблони.	Лекція, (2 год.)	[1-5,7,11,12]	Опрацювання лекційного матеріалу (4 год.)	2 тижні
	<b>Тема 7.</b> Розв'язування задач з використанням шаблонів Ланцюг обов'язків, Відвідувач, Посередник, спостерігач.	Лабораторне (4 год.)	[2,9-12]	Виконання домашнього завдання, (4 год.) робота над груповим проєктом (6 год.)	2 тижні
8.	<b>Тема 8.</b> Структурні шаблони.	Лекція (2 год.)	[5,7,8]	Опрацювання лекційного матеріалу (4 год.)	2 тижні
	<b>Тема 8.</b> Робота з шаблонами Декоратор, Фасад, Міст. Особливості поєднання різних шаблонів.	Лабораторне (6 год.)	[5,7,8]	Виконання домашнього завдання, (4 год.), робота над груповим проєктом. (5 год.)	3 тижні
9	<b>Тема 9.</b> Презентація групових проєктів. Обговорення переваг і недоліків обраної методики проєктування.	Лекція (2 год.)	[1-4,6]	Підготовка звіту. Огляд стандартних пасток проєктування. (4 год.)	2 тижні

	<b>Тема 9.</b> захист групових проєктів.	Лабораторне (4 год)		Аналіз виконання роботи іншої команди(8 год)	2 тижні
--	------------------------------------------	---------------------	--	----------------------------------------------	---------