

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Львівський національний університет імені Івана Франка**  
**Факультет електроніки та комп'ютерних технологій**  
**Кафедра радіоелектронних і комп'ютерних систем**

**Затверджено**

На засіданні  
кафедри радіоелектронних і комп'ютерних  
систем  
факультету електроніки та комп'ютерних  
технологій  
Львівського національного університету  
імені Івана Франка  
(протокол № 1/24 від 28.08.2023 р.)

Завідувач кафедри:

  
\_\_\_\_\_ Ігор ОЛЕНИЧ

**Силабус з навчальної дисципліни**  
**“Патерни проектування”,**  
**що викладається в межах ОПП**  
**“Інженерія програмного забезпечення”**  
**першого (бакалаврського) рівня вищої освіти для здобувачів з**  
**спеціальності 121 – Інженерія програмного забезпечення**

Львів 2023 р.

<b>Назва дисципліни</b>	Патерни проектування
<b>Адреса викладання дисципліни</b>	Корпус факультету електроніки та комп'ютерних технологій, Львівський національний університет імені Івана Франка, вул. Драгоманова 50, м. Львів, 79005
<b>Факультет та кафедра, за якою закріплена дисципліна</b>	Факультет електроніки та комп'ютерних технологій Кафедра радіоелектронних і комп'ютерних систем
<b>Галузь знань, шифр та назва спеціальності</b>	12 – інформаційні технології 121 – Інженерія програмного забезпечення
<b>Викладачі дисципліни</b>	Сінкевич О.О., доктор філософії з комп. н., доцент кафедри радіоелектронних і комп'ютерних систем
<b>Контактна інформація викладачів</b>	oleh.sinkevych@lnu.edu.ua <a href="https://electronics.lnu.edu.ua/employee/o_sinkevych">https://electronics.lnu.edu.ua/employee/o_sinkevych</a>
<b>Консультації з питань навчання по дисципліні відбуваються</b>	Консультації в день проведення лекцій/лабораторних занять (за попередньою домовленістю): ауд. 102, корпус факультету електроніки та комп'ютерних технологій, вул. Драгоманова 50, м. Львів
<b>Сторінка курсу</b>	<a href="https://electronics.lnu.edu.ua/course/paterny-proektuvannia-ipz">https://electronics.lnu.edu.ua/course/paterny-proektuvannia-ipz</a> <a href="https://e-learning.lnu.edu.ua/course/view.php?id=4405">https://e-learning.lnu.edu.ua/course/view.php?id=4405</a>
<b>Коротка анотація дисципліни</b>	Дисципліна “Патерни проектування” є нормативною дисципліною з спеціальності 121 – Інженерія програмного забезпечення для освітньої програми “Інформаційні системи та технології”, яка викладається в 3-му семестрі в обсязі 6-ох кредитів (за Європейською Кредитно-Трансферною Системою ECTS).
<b>Інформація про дисципліну</b>	Курс розроблено для ознайомлення студентів з основними поняттями та патернами проектування об'єктно-орієнтованих систем, класифікацією та особливостями патернів проектування та підходами до їх використання при розробці програмних засобів.
<b>Мета та цілі дисципліни</b>	<i>Мета:</i> одержання студентами знань про основні види патернів проектування; об'єднання об'єктно-орієнтованого коду у контексті породжувальних, структурних і поведінкових патернів; процес побудови структурних одиниць патернів; стандарти оформлення об'єктно-орієнтованого коду; механізми взаємодії складної структури класів. <i>Цілі:</i> навчити студентів реалізовувати і застосування патерни проектування; ознайомити студентів з основними видами патернів: структурного, породжувального і поведінкового; забезпечити розуміння проблематики та своєчасності застосування конкретних патернів на багатьох прикладах; показати нюанси і схеми конкретних патернів і дати навички для їх використання у проектування типових рішень в межах об'єктно-орієнтованої парадигми.
<b>Література для вивчення дисципліни</b>	Основна література: 1) Erich Gamma, John Vlissides, Richard Helm, Ralph Johnson. Design patterns, software engineering, object-oriented programming. Addison-Wesley, 1994. – 335 p. 2) Ерік Фрімен. Патерни проектування. Фабула, 2020. – 672 с. 3) Sean Bradley. Design Patterns In Python. Common GoF (Gang of Four)

	<p>Design Patterns Implemented In Python, 2021. – 238 p.</p> <p>4) Robert Martin. Clean Code: A Handbook of Agile Software Craftsmanship. Pearson Education (US), 2008. – 464 p.</p> <p>5) John F. Dooley. Software Development, Design and Coding. Apress, 2017. – 330 p.</p> <p>6) Швець О. Refactoring and design patterns. <i>Refactoring and Design Patterns</i>. URL: <a href="https://refactoring.guru/">https://refactoring.guru/</a>.</p> <p>7) Kasun Indrasiri. Design Patterns for Cloud Native Applications: Patterns in Practice Using APIs, Data, Events, and Streams. O'Reilly Media, 2021. – 314 с.</p> <p>8) J Geewax. API Design Patterns. Manning Publications Co, 2021. - 480 с.</p> <p>Додаткова література:</p> <p>9. Bewis T. C# Design Pattern Essentials. – NY: Ability First Limited, 2012. – 264 p.</p> <p>10. Maciaszek L. Requirements analysis and systems design / L. Maciaszek. – Ontario: Pearson Education Canada, 2007. - 656 p.</p> <p>11. Richards M. Fundamentals of Software Architecture. An Engineering Approach / M. Richards, N. Ford. - Sebastopol, CA: O'Reilly Media, 2020. - 422 p.</p> <p>12. Brown S. Visualise, document and explore your software architecture. Software Architecture for Developers / Simon Brown. - Leanpub, 2019.</p> <p>13. Eyskens S. Software Architecture for Busy Developers / S. Eyskens. - Birmingham: Packt Publishing, 2021. - 174 p.</p>
<p><b>Обсяг курсу</b></p>	<p>Загальний обсяг: 180 годин. Аудиторних занять: 64 год., з них 32 год. лекційних та 32 годин лабораторних робіт. Самостійної роботи: 116 год.</p>
<p><b>Очікувані результати навчання</b></p>	<p>У результаті вивчення даного курсу студент буде:</p> <p><b>знати:</b></p> <p>що таке патерн проектування і з чого він складається; класифікацію патернів, історію їх виникнення та фундаментальні джерела; принципи проектування та якості правильної архітектури коду; інтерфейси класів; принцип SOLID; коли і як застосовувати конкретну реалізацію патерна.</p> <p><b>вміти:</b></p> <p>правильно оформляти об'єктно-орієнтований код; застосовувати засвоєний матеріал для створення об'єктно-орієнтованих програм; використовувати Python для програмної реалізації патернів; реалізувати просте модульне тестування; визначати, коли необхідно використовувати патерн певного типу.</p> <p>Після вивчення даного курсу «Патерни проектування» здобувачі набудуть таких Загальних та Фахових компетентностей та Програмних результатів навчання:</p> <p>ЗК01. Здатність до абстрактного мислення, аналізу та синтезу.</p> <p>ЗК02. Здатність застосовувати знання у практичних ситуаціях.</p> <p>ЗК06. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.</p> <p>ЗК07. Здатність працювати в команді.</p> <p>ЗК10. Здатність діяти соціально відповідально та свідомо.</p> <p>ФК13. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення.</p> <p>ФК14. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури,</p>

	<p>поведінки та процесів функціонування.</p> <p>ФК15. Здатність розробляти архітектури, модулі та компоненти програмних систем.</p> <p>ФК16. Здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами замовника, технічним завданням та стандартами.</p> <p>ФК17. Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу.</p> <p>ФК18. Здатність аналізувати, вибирати і застосовувати методи і засоби для забезпечення інформаційної безпеки (в тому числі кібербезпеки).</p> <p>ФК19. Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних.</p> <p>ФК27. Здатність використовувати для розробки програмного забезпечення перспективні засоби та технології, зокрема, науки про дані, штучного інтелекту, IoT, вбудованих систем тощо.</p> <p>ФК28. Володіння методами, серверними технологіями та інструментальними засобами проектування веб-застосувань.</p> <p>ФК29. Здатність здійснювати розробку програмного забезпечення використовуючи різні методології програмування (в тому числі паралельне, об'єктно-орієнтоване, функціональне програмування та інші)</p> <p>ПРН04. Знати і застосовувати професійні стандарти і інші нормативно-правові документи в галузі інженерії програмного забезпечення.</p> <p>ПРН05. Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення.</p> <p>ПРН06. Вміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення.</p> <p>ПРН07. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.</p> <p>ПРН12. Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення.</p> <p>ПРН14. Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення.</p> <p>ПРН17. Вміти застосовувати методи компонентної розробки програмного забезпечення.</p> <p>ПРН20. Знати підходи щодо оцінки та забезпечення якості програмного забезпечення.</p> <p>ПРН25. Вміти розробляти програмно-апаратні рішення з використанням технологій програмування вбудованих систем.</p> <p>ПРН27. Вміти обирати оптимальні алгоритми та технології розробки програмного забезпечення.</p>
<b>Ключові слова</b>	Патерни проектування, об'єктно-орієнтоване програмування, SOLID, принципи проектування, рефакторинг.
<b>Формат курсу</b>	Очний
<b>Теми</b>	<i>Див. Схема курсу</i>
<b>Підсумковий</b>	Залік у кінці семестру

<b>контроль, форма</b>	
<b>Пререквізити</b>	Для вивчення даного курсу студентам потрібні базові знання з курсів: <ul style="list-style-type: none"> <li>- основи програмування;</li> <li>- теорія алгоритмів;</li> <li>- алгоритмізація і програмування;</li> <li>- об'єктно-орієнтоване програмування.</li> </ul>
<b>Навчальні методи та техніки, які будуть використовуватися під час викладання курсу</b>	Інформаційні методи (лекція, бесіда, ілюстрація, демонстрація); дедуктивні методи на основі узагальнень; евристичні методи (проблемна лекція); інтерактивні методи (дискусія).
<b>Необхідне обладнання</b>	Для проведення лекційних занять: комп'ютер (мінімальні характеристики: процесор Intel Core i3(4 ядра/8 потоків), 8ГБ оперативної пам'яті, 50ГБ вільного місця на диску, доступ до мережі Internet, засоби мультимедіа (в т.ч. проектор).  Для проведення лабораторних занять: Комп'ютер (мінімальні характеристики: процесор Intel Core i3(4 ядра/8 потоків), 8ГБ оперативної пам'яті, 50ГБ вільного місця на диску. Необхідне програмне забезпечення включає в себе ОС Windows 10, або Ubuntu 22.04 LTS, середовище розробки MS Visual Studio/PyCharm/Vim/VS Code, компілятор мови програмування C++/Python.
<b>Критерії оцінювання (окремо для кожного виду навчальної діяльності)</b>	Оцінювання проводиться за 100-бальною шкалою. Бали нараховуються за наступним співвідношенням: <ul style="list-style-type: none"> <li>• Змістовий модуль 1: 20% семестрової оцінки за виконане модульне завдання, максимальна кількість балів 20.</li> <li>• Змістовий модуль 2: 20% семестрової оцінки за виконані лабораторні роботи, максимальна кількість балів 20.</li> <li>• Лабораторні роботи: 60% семестрової оцінки, максимальна кількість балів 60.</li> </ul> <p>Підсумкова максимальна кількість балів 100.</p> <p><b>Змістовий модуль</b> – самостійна робота студента оформлена у вигляді робочої програми та звіту – робота друкованим текстом, рекомендованим обсягом до 5 сторінок (шрифт Times New Roman, 14). Звіт включає в себе детальний розгляд завдання до модуля, приведення прикладів та лістингів коду програм, огляду технологій. Код програм повинен бути обов'язково прокоментований та пояснений, необхідно також продемонструвати його роботу у разі, якщо в якості прикладу наводяться не окремі елементи технології, а суцільна програма. Фінальна версія звіту у .pdf форматі разом з кодом завантажується на віддалений Git-репозиторій.</p> <p><b>Академічна доброчесність:</b> Очікується, що роботи студентів будуть оригінальними дослідженнями чи міркуваннями. Списування та втручання в роботу інших студентів становлять, але не обмежують, приклади можливої академічної недоброчесності. Виявлення ознак</p>

академічної недоброчесності в написанні завдань є підставою для її незарахування викладачем, незалежно від масштабів плагіату чи обману. Жодні форми порушення академічної доброчесності не толеруються.

**Відвідання занять** є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції та лабораторні заняття курсу. Студенти повинні інформувати викладача про неможливість відвідати заняття. У будь-якому випадку студенти зобов'язані дотримуватися термінів визначених для виконання всіх видів робіт, передбачених курсом.

**Література.** Уся література, яку студенти не зможуть знайти самостійно, буде надана викладачем виключно в освітніх цілях без права її передачі третім особам. Студенти заохочуються до використання також й іншої літератури та джерел, яких немає серед рекомендованих.

**Політика виставлення балів.** Враховуються бали, набрані при поточному контролі та бали за виконання лабораторних робіт. При цьому обов'язково враховуються присутність на заняттях та активність студента під час практичного заняття; недопустимість пропусків та запізнь на заняття; користування мобільним телефоном, планшетом чи іншими мобільними пристроями під час заняття в цілях не пов'язаних з навчанням; списування та плагіат; несвоєчасне виконання поставленого завдання і т. ін.

**Оцінювання лабораторних робіт** (8 лабораторних робіт, максимальна кількість балів: 60) відбувається шляхом оцінки роботи студента під час проведення лабораторної роботи в аудиторії та захисту звіту по виконаній лабораторній роботі (0-7.5 відносних балів за одну роботу).

Бали оцінювання лабораторних робіт нараховуються за наступним співвідношенням:

7-7.5 – студент в повному обсязі володіє навчальним матеріалом, має повне розуміння розглянутої теми, надає правильні відповіді на запитання по темі, код програми функціонує відповідно до завдання;

6 – студент достатньо розуміє розглянутий матеріал та принципи написаного ним коду програми, присутні неточності та незначні помилки у відповідях на запитання по темі, код програми функціонує відповідно до завдання (або з несуттєвими недоліками);

5 - студент достатньо розуміє розглянутий матеріал та принципи написаного ним коду програми, присутні неточності та незначні помилки у відповідях на запитання по темі, код програми функціонує відповідно до завдання, проте з помірними недоліками;

4 - студент не досить добре розуміє розглянутий матеріал та написаний ним код програми, вагається та надає неточні/не конкретні відповіді на запитання по темі, код програми функціонує неточно, або з помірними недоліками;

3 – студент не досить добре розуміє розглянутий матеріал та написаний ним код програми, вагається та надає неточні/не конкретні відповіді на запитання по темі, код програми функціонує неточно, або з вагомими недоліками;

2 – студент погано розуміє розглянутий матеріал та написаний ним код програми, студент в більшості надає помилкові відповіді на питання по темі, код програми функціонує з суттєвими недоліками;

1 - студент погано розуміє розглянутий матеріал та написаний ним код

	<p>програми, код програми не функціонує належним чином;  0 - студент зовсім не засвоїв розглянутий матеріал, написаний ним код програми не відповідає темі/не функціонує взагалі.</p> <p><b>Оцінювання змістових модулів</b> (2 змістових модулів, 20 балів за кожний) — за результатами написаних студентом звіту та коду програми.  Бали оцінювання змістових модулів нараховуються за наступним співвідношенням:  16-20 - розглянута тема відтворюється в повному обсязі, правильно, обґрунтовано, логічно, містить аналіз і систематизацію, аргументовані висновки. Засвідчено глибоке володіння матеріалом. Наведений код повністю робочий та відповідає темі. Можуть бути присутні несуттєві помилки та невідповідності;  12-16 - відтворюється значна частина розглянутої теми. Виявлено знання і розуміння основних положень навчальної дисципліни, проте присутні неточності та/або невідповідності основній темі. Наведений код частково робочий, проте в загальному відповідають темі;  8-12 - відстежується загальне розуміння розглянутої теми. Виявлені множинні неточності та невідповідності, пояснення наведеного коду відсутні, код функціонує із значними неточностями (або відсутні приклади запуску коду на виконання взагалі);  4-8 – студент погано розуміє розглянуту тему. Виявлені суттєві неточності та невідповідності. Наведені приклади коду з суттєвими недоліками, або не відповідають темі;  4 – 0 – студент взагалі не розуміє розглянуту тему. Тему не розкрито, кількість викладеного матеріалу не відповідає загальним нормам обраного виду роботи. Наведений код не робочий, або відсутній як такий.</p> <p><b>Критерії оцінювання результатів неформальної освіти:</b>  Нарахування балів відбувається за написання студентом тез доповідей на конференціях, наукових статей, участь у діяльності наукових гуртків, участь у наукових семінарах та круглих столах, конкурсах, участь у заходах неформальної освіти за отримання сертифікатів про проходження навчання на різних освітніх платформах (Coursera, Prometheus тощо), курсах на провідних ІТ компаніях за тематикою навчальної дисципліни. Кількість балів визначається відсотком покриття результатів відповідної активності до вимог результатів навчання з навчальної дисципліни.</p>
<p><b>Питання до заліку чи екзамену.</b></p>	<p>Об'єктно-орієнтоване програмування  Поняття об'єкту, класу  Парадигми ООП. Інкапсуляція та поліморфізм.  Інтерфейс та стан об'єкту  Базовий та абстрактний класи. Успадкування  Принципи та підходи до проектування ООП систем  Класифікація патернів проектування  Група патернів фабрика. Ідіома  Фабричний метод та абстрактна фабрика  Патерн будівельник  Патерн одинак  Патерн прототип. Змінні та незмінні типи.  Метакласи та їх особливості  Структурні патерни. Адаптер</p>

	Патерн Міст та фасад Декоратори функцій. Патерн декоратор Поведінкові патерни. Патерн спостерігач Патерн стратегія Патерн стан та шаблонний метод Архітектурні патерни. Клієнт-серверна взаємодія Патерн Модель — Вид — Контролер.
<b>Опитування</b>	Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенню курсу.



**Схема курсу “Патерни проектування”  
для студентів спеціальності 121 – Інженерія програмного забезпечення**

Тиж .	Тема, план, короткі тези	Форма діяльності (заняття)* *лекція, самостійна, дискусія, групова робота)	Література.** * Ресурси в інтернеті	Завдання, год	Термін виконання
1	Огляд об'єктно-орієнтованого програмування та основних засад.	Лекція	[1], [2], [3], [8] Сайт курсу	2	кінець поточного тижня
1	Лаб. 1. Робота з класами. Реалізація взаємозв'язків між об'єктами.	Лабораторна робота	Сайт курсу	2	протягом двох тижнів
2	Патерни та принципи проектування. SOLID та Роберт Мартін.	Лекція	[1], [2], [3] Сайт курсу	2	кінець поточного тижня
2	Опрацювання книжки Роберта Мартіна — Чистий код (перші три глави)	Самостійна робота	[4]	2	кінець поточного тижня
3	Управління атрибутами. Абстрактний клас та абстрактні методи	Лекція	[1], [2], [3], Сайт курсу	2	кінець поточного тижня
3	Лаб. 2. Наслідування та абстрактні класи	Лабораторна робота	Сайт курсу	2	протягом двох тижнів
4	Фабричний метод та абстрактна фабрика	Лекція	[1], [2], [3], [6] Сайт курсу	2	кінець поточного тижня
4	Перегляд та засвоєння реалізації патернів (фабрики) різними мовами	Самостійна робота	[6], [8]	2	кінець поточного тижня
5	Патерн будівельник. Поєднання будівельника та фабрики	Лекція	[1], [2], [3], [6] Сайт курсу	2	кінець поточного тижня
5	Лаб. 3. Реалізація фабричного методу та абстрактної фабрики	Лабораторна робота	Сайт курсу	2	протягом двох тижнів
6	Змінні та незмінні типи. Прототип. Введення у метакласи	Лекція	[1], [2], [3], [6] Сайт курсу	2	кінець поточного тижня
6	Перегляд та засвоєння реалізації патерну прототип різними мовами. Ознайомлення з додатковими матеріалами	Самостійна робота	[6], [8]	2	кінець поточного тижня
7	Метакласи та їх використання у патерні одинак. Приклади	Лекція	[1], [2], [3], [6] Сайт курсу	2	кінець поточного тижня
7	Лаб. 4. Реалізація патернів будівельник та прототип	Лабораторна робота	Сайт курсу	2	протягом двох тижнів
8	Структурні патерни. Фасад. Основи декораторів	Лекція	[1], [2], [3], [6] Сайт курсу	2	кінець поточного тижня
8	Перегляд та засвоєння реалізації патерну фасад різними мовами. Ознайомлення з додатковими матеріалами	Самостійна робота	[1], [2], [3], [6] Сайт курсу	2	кінець поточного тижня
9	Патерн декоратор та патерн фасад	Лекція	[1], [2], [3], [6] Сайт курсу	2	кінець поточного тижня

9	Модульне завдання (породжувані та структурні патерни)	Модульне завдання	Сайт курсу	2	кінець поточного тижня
10	Поглиблення у структурні патерни. Патерн міст.	Лекція	[1], [2], [3], [6] Сайт курсу	2	кінець поточного тижня
10	Лаб. 5. Реалізація декоратора та адаптера	Лабораторн а робота	Сайт курсу	2	протягом двох тижнів
11	Патерн проксі. Компонувальник та дерево.	Лекція	[1], [2], [3], [6] Сайт курсу	2	кінець поточного тижня
11	Перегляд та засвоєння реалізації патерну проксі та компонентувальник різними мовами. Ознайомлення з додатковими матеріалами	Самостійна робота	[1], [2], [3], [6] Сайт курсу	2	кінець поточного тижня
12	Вступ до поведінкових патернів. Патерн стратегія.	Лекція	[1], [2], [3], [6] Сайт курсу	2	кінець поточного тижня
12	Лаб. 6. Реалізація патернів фасад та міст	Лабораторн а робота	Сайт курсу	2	протягом двох тижнів
13	Патерн спостерігач. Приклади	Лекція	[1], [2], [3], [6] Сайт курсу	2	кінець поточного тижня
13	Перегляд та засвоєння реалізації патерну стратегія та спостерігач різними мовами. Ознайомлення з додатковими матеріалами	Самостійна робота	[1], [2], [3], [6] Сайт курсу	2	кінець поточного тижня
14	Стейт-машина. Патерн стан.	Лекція	[1], [2], [3], [6] Сайт курсу	2	кінець поточного тижня
14	Лаб. 7. Реалізація патернів проксі, стратегії та спостерігача	Лабораторн а робота	Сайт курсу	2	протягом двох тижнів
15	Веб-патерни. Вступ до архітектурного патерну модель-вид-контролер ч.1.	Лекція	[1], [2], [3], [6] Сайт курсу	2	кінець поточного тижня
15	Модульне завдання (структурні та поведінкові патерни)	Модульне завдання	Сайт курсу	2	кінець поточного тижня
16	MVC патерн проектування та приклади розробки	Лекція	[1], [2], [3], [8] Сайт курсу	2	кінець поточного тижня
16	Лаб. 8. Реалізація простого MVC застосунку з використанням фреймворку на вибір студента	Лабораторн а робота	Сайт курсу	2	кінець поточного тижня