

СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ «ПАТЕРНИ ПРОЕКТУВАННЯ ТА ПРОГРАМУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ»



Ступінь освіти	бакалавр
Спеціальність	126 Інформаційні системи та технології
Тривалість викладання	7 семестр
Заняття:	1,2 чверті
Лекції	2 години на тиждень
Практичні	2 години на тиждень
Мова викладання	українська

Кафедра, яка викладає: Інформаційних технологій та комп'ютерної інженерії

Сторінка курсу в СДО НТУ «ДП»: <https://do.nmu.org.ua/course/view.php?id=4598>

Консультації: за окремим розкладом, що попередньо погоджений зі здобувачами освіти.

Онлайн-консультації: MS Teams, електронна пошта

Інформація про викладача:



Викладач:	Соколова Наталя Олегівна, к.т.н., доц.
Персональна сторінка:	https://it.nmu.org.ua/ua/HR_staff/prepods/Sokolova.php
E-mail:	Sokolova.N.O@nmu.one

1. Анотація до курсу

Програма навчальної дисципліни «Патерни проектування та програмування інформаційних систем», складена відповідно до освітньо професійної програми підготовки бакалавра 126 «Інформаційні системи та технології» галузі знань 12 «Інформаційні технології».

Програмне забезпечення стає основою сучасних досягнень у всіх областях науки, промисловості і оборони. У розробці ПЗ архітектура — процес створення застосунку в правильний, надійний і доступний спосіб. Патерни проектування - це один з інструментів розробника, який допомагає заощадити час і зробити якісніші рішення. Патерни визначають типові шляхи, шаблони розв'язань поширених проблем. Вони можуть бути як абстрактними й концептуальними, так і дуже точними й технічними, це дозволяє розробникам порозумітися. Шаблони проектування це методика, які допомагають розробнику розв'язувати проблеми при програмуванні інформаційних систем, вони значно полегшують щоденну рутину, та роблять проектування більш оптимальним процесом. За допомогою шаблонів проектування описуються та формалізуються типові завдання та їх вирішення.

У даному курсі розглядаються основні принципи проектування, класифікація та принципи реалізації 23 основних патернів, а також прийому рефакторингу програмного коду.

2. Мета та завдання навчальної дисципліни

Мета – формування знань щодо ролі патернів у створенні архітектури складних систем, та формуванні умінь застосовувати патерни при проектуванні власних інформаційних систем управління.

Реалізація мети вимагає трансформації програмних результатів навчання в дисциплінарні та адекватний відбір змісту навчальної дисципліни за цим критерієм..

Основними **завданнями** вивчення дисципліни є

- вивчити основні теоретичні положення з використання патернів проектування;
- формування навичок аналізу програмного коду поширених фреймворків, визначати використані патерни проектування;
- формування навичок аналізу проектних рішень при розробці програмного забезпечення
- навчити здобувачів вищої освіти визначати доцільність та необхідність застосування відповідних патернів проектування;
- оволодіти методологією проектування та реалізації патернів проектування, необхідних для створюваного програмного забезпечення.

3. Результати навчання

1. Використовувати об'єктно-орієнтовані мови програмування для реалізації шаблонів проектування.
2. Вміти проводити об'єктно-орієнтований аналіз предметної області та описувати архітектуру інформаційних систем, що розробляються.
3. Використовувати засоби моделювання шаблонів проектування.
4. Вміти обґрунтовано надати пояснення використання архітектурних шаблонів.
5. Здійснювати реалізацію архітектурних шаблонів.
6. Використовувати певні шаблони проектування та програмування при побудові різноманітних програмних рішень.

4. Структура курсу

ЛЕКЦІЇ

- 1 Об'єктно-орієнтований аналіз та проектування
 - Поняття проектування програмного забезпечення на основі аналізу предметної області
 - Зв'язки між об'єктами
- 2 Введення в шаблони проектування
 - Передумови виникнення шаблонів проектування.
 - Основи патернів
- 3 Принципи проектування
 - Базові принципи проектування
 - Принципи SOLID
- 3 Породжувальні шаблони проектування
 - Фабричний метод
 - Одинак
 - Шаблон проектування «Абстрактна фабрика»
 - Шаблони проектування «Будівельник», «Прототип»
 - Пул об'єктів
- 4 Структурні шаблони проектування
 - Загальний опис структурних патернів
 - Структурні патерни рівня класів та об'єктів

- Адаптер
- Міст
- Компонувальник
- Декоратор
- Фасад
- Легковаговик
- Замісник
- 5 Патерни поведінки
 - Загальний опис патернів поведінки
 - Програмні механізми, що використовуються при застосуванні патернів поведінки
 - Ланцюг відповідальностей
 - Ітератор
 - Команда
 - Посередник
 - Знімок
 - Спостерігач
 - Стан
 - Стратегія
 - Шаблонний метод
 - Відвідувач
- 6 Архітектурні шаблони
 - Модель-вид-контролер
 - Клієнт-серверна архітектура
 - Триярусна архітектура
 - Сервісно-орієнтована архітектура
 - Мікросервіси
- 7 Шаблони інтеграції
 - Сервіс-орієнтована архітектура
 - Подійно-орієнтована архітектура

ПРАКТИЧНІ ЗАНЯТТЯ

- 1 Розв'язання задач із використанням основних принципів об'єктно-орієнтованого програмування
- 2 Принципи рефакторингу. Чистий код
- 3 Прийоми рефакторингу. Складання методів
- 4 Прийоми рефакторингу. Переміщення функцій між об'єктами
- 5 Прийоми рефакторингу. Організація даних
- 6 Прийоми рефакторингу. Спрощення умовних виразів
- 7 Прийоми рефакторингу. Спрощення викликів методів
- 8 Прийоми рефакторингу. Задачі узагальнення об'єктів

Технічне обладнання та/або програмне забезпечення

1. Персональний комп'ютер або ноутбук зі сталим доступом до мережі Інтернет
2. Активованій акаунт університетської пошти (student.i.p.@nmu.one) на Офіс365.
3. Активний обліковий запис у системі дистанційної освіти Moodle.
4. Програмне забезпечення:
 - Microsoft Office;
 - Середовище програмування.

5. Система оцінювання та вимоги

5.1. Навчальні досягнення здобувачів вищої освіти за результатами вивчення курсу оцінюватимуться за шкалою, що наведена нижче:

Рейтингова шкала	Інституційна шкала
90 – 100	відмінно
75 – 89	добре
60 – 74	задовільно
0 – 59	незадовільно

5.2. Здобувач вищої освіти може отримати **підсумкову оцінку** з навчальної дисципліни на підставі поточного оцінювання знань за умови, якщо набрана кількість балів з поточного тестування та самостійної роботи складатиме не менше 60 балів.

Поточна успішність складається з оцінок за лекційну частину курсу та практичні заняття. Отримані бали осереднюються і є підсумковою оцінкою за вивчення навчальної дисципліни. Максимально за поточною успішністю здобувач вищої освіти може набрати 100 балів.

Максимальне оцінювання:

Теоретична частина	Практична частина		Разом (середнє)
	При своєчасному складанні	При несвоєчасному складанні	
50	50	40 і нижче	100

Практичні завдання приймаються за контрольними запитаннями до кожної роботи.

Теоретична частина оцінюється за результатами задачі тестових робіт, яка містить 20 запитань.

5.3. Критерії оцінювання теоретичної частини курсу.

Під час проведення контрольних заходів наприкінці першої, другої/третьої, четвертої чверті здобувачі вищої освіти складають тести, що містять 20 питань. На кожне питання надається 4 варіанти відповіді, серед яких лише 1 – вірний. Максимальна оцінка за тест складає 25 балів. Опитування за тестом проводиться з використанням системи дистанційної освіти Moodle.

5.4. Критерії оцінювання практичних робіт.

З кожної практичної роботи здобувач вищої освіти оформлює звіт, що містить завдання, результати його виконання та висновки. При своєчасному (протягом тижня від дати проведення заняття) та вірному виконанні завдання здобувач отримує оцінку 100. При несвоєчасному виконанні завдання та/або наявності помилок оцінка складає 90 балів і нижче.

6. Політика курсу

6.1. Політика щодо академічної доброчесності

Академічна доброчесність здобувачів вищої освіти є важливою умовою для опанування результатами навчання за дисципліною і отримання задовільної оцінки з поточного та підсумкового контролів. Академічна доброчесність базується на засудженні практик списування (виконання письмових робіт із залученням зовнішніх джерел інформації,

крім дозволених для використання), плагіату (відтворення опублікованих текстів інших авторів без зазначення авторства), фабрикації (вигадування даних чи фактів, що використовуються в освітньому процесі). Політика щодо академічної доброчесності регламентується положенням "Положення про систему запобігання та виявлення плагіату у Національному технічному університеті "Дніпровська політехніка". http://www.nmu.org.ua/ua/content/activity/us_documents/System_of_prevention_and_detection_of_plagiarism.pdf.

У разі порушення здобувачем вищої освіти академічної доброчесності (списування, плагіат, фабрикація), робота оцінюється незадовільно та має бути виконана повторно. При цьому викладач залишає за собою право змінити тему завдання.

6.2. Комунікаційна політика

Здобувачі вищої освіти повинні мати активовану університетську пошту.

Усі письмові запитання до викладачів стосовно курсу мають надсилатися на університетську електронну пошту.

6.3. Політика щодо перескладання

Роботи, які здаються із порушенням термінів без поважних причин оцінюються на нижчу оцінку. Перескладання відбувається із дозволу деканату за наявності поважних причин (наприклад, лікарняний).

6.4. Відвідування занять

Для здобувачів вищої освіти денної форми відвідування занять є обов'язковим. Поважними причинами для неявки на заняття є хвороба, участь в університетських заходах, академічна мобільність, які необхідно підтверджувати документами. Про відсутність на занятті та причини відсутності здобувач вищої освіти має повідомити викладача або особисто, або через старосту.

За об'єктивних причин (наприклад, міжнародна мобільність) навчання може відбуватись в он-лайн формі за погодженням з керівником курсу.

6.5. Політика щодо оскарження оцінювання

Якщо здобувач вищої освіти не згоден з оцінюванням його знань він може опротестувати виставлену викладачем оцінку у встановленому порядку.

6.6. Студентоцентризований підхід

Для врахування інтересів та потреб студентів на початку вивчення курсу здобувачам вищої освіти пропонується відповісти у системі Moodle на низку питань щодо інформаційного наповнення курсу. Відповідно до результатів опитування формується траєкторія навчання з урахуванням потреб студентів.

Під час навчання студенти реалізують своє право вибору індивідуальних завдань практичних завдань.

Наприкінці вивчення курсу та перед початком сесії здобувачам вищої освіти пропонується анонімно заповнити у системі Moodle електронні анкети для оцінки рівня задоволеності методами навчання і викладання та врахування пропозицій стосовно покращення змісту навчальної дисципліни. За результатами опитування вносяться відповідні корективи у робочу програму та силабус.

7. Рекомендовані джерела інформації

- 1 Стандарт вищої освіти України: перший (бакалаврський) рівень, галузь знань 12 – Інформаційні технології, спеціальність 126 – Інформаційні системи та технології. Затверджено Наказом Міністерства освіти і науки України 12.12.2018 р. № 1380. – 17 с.
- 2 Mark Richards. Fundamentals of Software Architecture: An Engineering Approach. – O'Reilly Media, 2020. – 432 p. ISBN-10: 1492043451, ISBN-13: 978-1492043454.
- 3 Alan Dennis, Barbara Wixom, David Tegarden. Systems Analysis and Design: An Object-Oriented Approach with UML, 5th Edition. – Wiley, 2015. – 544 p. ISBN-10: 1118804678, ISBN-13: 978-1118804674.
- 4 Grady Booch. Object-Oriented Analysis and Design with Applications, 3rd Edition / Grady Booch, Robert A. Maksimchuk, Michael W. Engle, Bobbi J. Young, Jim Conallen, Kelli A. Houston. – Addison-Wesley Professional, 2007. – 720 p. ISBN-10: 020189551X, ISBN-13: 978-0201895513.
- 5 Eric Freeman, Bert Bates, Kathy Sierra, Elisabeth Robson. Head First. Design Patterns: A Brain-Friendly Guide. – O'Reilly Media, 2004. – 694 p. ISBN-10: 9780596007126, ISBN-13: 978-0596007126.
- 6 Erich Gamma. Design Patterns: Elements of Reusable Object-Oriented Software / Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Grady Booch. – Addison-Wesley Professional, 1994. – 416 p. ISBN-10: 0201633612, ISBN-13: 978-0201633610.
- 7 Stephen D. Burd. Systems Architecture, 7th Edition. – Cengage Learning, 2015. – 656 p. ISBN-10: 130508019X, ISBN-13: 978-1305080195.
- 8 Eberhard Wolff. Microservices: Flexible Software Architecture. – Addison-Wesley Professional, 2016. – 432 p. ISBN-10: 0134602412, ISBN-13: 978-0134602417.
- 9 Mathieu Nayrolles. Angular Design Patterns: Implement the Gang of Four patterns in your apps with Angular, 2018. – 225p.
- 10 Wessel Badenhorst. Practical Python Design Patterns: Pythonic Solutions to Common Problems. Apress, 2017. - 352 Pages
- 11 Edward Sciore . Java Program Design: Principles, Polymorphism, and Patterns. Apress, 2019. - 465 Pages
- 12 Р.С.Мартин. Чистий код. Створення і рефакторинг за допомогою Agile. Фабула, 2019. – 448с.
- 13 Е. Фрімен, Е. Робсон. Head First. Патерни проектування. – Харків.: Фабула, 2020. – 672 с. ISBN: 978-617-09-6159-4.