

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



Затверджено

На засіданні кафедри програмування
факультету прикладної математики та
інформатики
Львівського національного університету
імені Івана Франка
(протокол № 1 від 29 серпня 2024 р.)
Зав. кафедри Сергій ЯРОШКО

Силабус з навчальної дисципліни
«Об'єктно-орієнтований аналіз і проектування програмного
забезпечення»,
що викладається в межах ОПІ другого (магістерського) рівня вищої
освіти для здобувачів зі спеціальності
122 Комп'ютерні науки

Львів 2024 р.

| | |
|--|---|
| Назва дисципліни | Об'єктно-орієнтований аналіз і проектування програмного забезпечення |
| Адреса викладання дисципліни | Львівський національний університет імені Івана Франка, вул. Університетська 1, м. Львів, Україна, 79000 |
| Факультет та кафедра, за якою закріплена дисципліна | Факультет прикладної математики та інформатики, кафедра програмування |
| Галузь знань, шифр та назва спеціальності | Галузь знань: 12 Інформаційні технології Спеціальність: 122 Комп'ютерні науки |
| Викладачі дисципліни | Музичук Анатолій Омелянович, кандидат фізико-математичних наук, доцент кафедри програмування Кушчак Петро Богданович, старший викладач кафедри програмування |
| Контактна інформація викладачів | Електронна пошта: petro.kushchak@lnu.edu.ua, веб-сторінка: https://ami.lnu.edu.ua/employee/kuschak-petro-bohdanovych |
| Консультації з питань навчання по дисципліні відбуваються | Консультації в день проведення лекцій/лабораторних занять (за попередньою домовленістю та за умови проведення аудиторних занять). В іншому випадку можливі он-лайн консультації через Zoom чи MSTEams. Для погодження часу он-лайн консультацій слід писати на електронну пошту викладача або дзвонити. |
| Сторінка курсу | https://ami.lnu.edu.ua/course/object-oriented-analysis-informatics |
| Інформація про дисципліну | Дисципліна «ООА і проектування ПЗ» є нормативною дисципліною з спеціальності комп'ютерні науки, яка викладається в першому семестрі в обсязі 6 кредитів (за Європейською Кредитно-Трансферною Системою ECTS). |
| Коротка анотація дисципліни | Курс розроблено таким чином, щоб надати учасникам необхідні знання, обов'язкові для того, щоб ефективно здійснювати моделювання та проектування ПЗ у професійній діяльності. Тому у курсі представлено як огляд концепцій теоретичних основ та принципів аналізу та проектування ПЗ, побудови та реалізації відповідних моделей. |
| Мета та цілі дисципліни | Мета: вивчення нормативної вибіркової дисципліни "ООА і проектування ПЗ" є ознайомлення студентів із завданнями з основними концепціями та принципами аналізу та проектування ПЗ – починаючи з підходів DRY/YAGNI і завершуючи RMA-моделюванням та побудовою моделі загроз, необхідними для оволодіння сучасними підходами та інструментами для їх вирішення. Цілі: формування знань та навиків про концепції, принципи аналізу та проектування ПЗ, враховування ризиків та забезпечення якісного процесу розробки програмного забезпечення з допомогою сучасних технологій, підходів та методів; набуття компетенцій, знань, умінь та навиків відповідно до кваліфікації магістра за спеціальністю "Комп'ютерні науки"; |
| Література для вивчення дисципліни | <i>Основна література:</i> 1. 2021 - Neal Ford, Mark Richards "Software Architecture: The Hard Parts" - ISBN 978-1492086895 2. 2020 - Mark Richards and Neal Ford "Fundamentals of Software Architecture" - ISBN 978-1492043454 3. 2018 - Joseph Ingeno Software Architect's Handbook - ISBN: 9781788624060 4. https://docs.microsoft.com/en-us/azure/architecture/patterns/ <i>Додаткова література</i> |

| | |
|--------------------------------------|--|
| | <ol style="list-style-type: none"> 1. 2022 - Mark Richards "Software Architecture Patterns 2nd Edition" - ISBN 978-1-098-13427-3 2. 2017 - Robert C. Martin. "Principles Of OOD" 3. https://simpleprogrammer.com/kiss-one-best-practice-to-rule-them-all/ 4. https://dzone.com/articles/software-design-principles-dry-and-kiss 5. https://www.itexico.com/blog/software-development-kiss-yagni-dry-3-principles-to-simplify-your-life 6. https://itnext.io/solid-principles-explanation-and-examples-715b975dcad4 7. http://download.microsoft.com/download/D/8/C/D8C599A4-4E8A-49BF-80EE-FE35F49B914D/Resilience_by_Design_for_Cloud_Services_White_Paper.pdf 8. https://www.amazon.com/Threat-Modeling-Designing-Adam-Shostack/dp/1118809998 9. https://enterprisecraftsmanship.com/posts/types-of-cqrs/ 10. https://www.openshift.com/microservices-ebook/ 11. https://www.amazon.com/NoSQL-SQL-Data-Modeling-Semantics/dp/1634621093 12. https://martinfowler.com/eaCatalog/ 13. https://www.manning.com/books/serverless-architectures-on-aws-second-edition 14. https://medium.com/@NeotericEU/single-page-application-vs-multiple-page-application-2591588efe58 15. https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-reliable-actors-introduction 16. https://www.amazon.com/Web-Analytics-2-0-Accountability-Centricity/dp/0470529393 17. https://www.amazon.com/Effective-Monitoring-Alerting-Web-Operations/dp/1449333524 |
| Обсяг курсу | 64 години аудиторних занять. З них 32 годин лекцій, 32 годин лабораторних занять та 116 годин самостійної роботи |
| Очікувані результати навчання | <p>Після завершення цього курсу студент буде :</p> <ul style="list-style-type: none"> ● знати: <ol style="list-style-type: none"> 1) основні концепції та принципами аналізу та проектування ПЗ 2) категорії та види моделювання нефункціональних характеристик ПЗ, особливості їх застосування; ● вміти: <ol style="list-style-type: none"> 1) здійснювати розробку типових технологічних процесів автоматизованої обробки інформації; 2) розробляти моделі потоків даних та загроз, RMA-моделі 3) налаштовувати інструменти збору аналітики та моніторингу ПЗ |
| Компетентності | <p>Інтегральна компетентність: ІК, Загальні компетентності: ЗК 1, ЗК 2, ЗК4, ЗК 5 Спеціальні (фахові) компетентності: СК2, СК 5, СК 6, СК 7, СК 8, СК 9, СК 10, СК 12, СК 13</p> |
| Програмні результати навчання | ПРН 1, ПРН 2, ПРН 3, ПРН 10, ПРН 11, ПРН 12, ПРН 13, ПРН 14, ПРН 15, ПРН 17, ПРН 18, ПРН 20 |
| Ключові слова | Моделі загроз, моделі RMA, аналітика, моніторинг, SOLID, GRASP, EAAP, CQRS |
| Формат курсу | Очний |

| Проведення лекцій, лабораторних робіт та консультації для кращого розуміння тем | | | | | | |
|---|-----|--|---|--|---------------|------------------------------|
| Теми | Тиж | Тема, план, короткі тези | Форма діяльності (заняття)* *лекція, самостійна, дискусія, групова робота) | Література.** * Ресурси в інтернеті | Завдання, год | Термін виконання |
| | 1 | Вступ. огляд принципів KISS, DRY, YAGNI | Лекція | 1, 7, 9, 10 | 2 | |
| | 1 | Розробка завдання з використанням KISS, DRY, YAGNI | Лабораторні заняття | 2, 3, 4, 6, 11 | 2 | Наступне лабораторні заняття |
| | 2 | Огляд принципів SOLID | Лекція | 1, 5, 7, 9, 10 | 2 | |
| | 2 | Розробка завдання з використанням SOLID | Лабораторні заняття | 2, 3, 4, 6, 11 | 2 | Наступне лабораторні заняття |
| | 3 | Огляд принципів GRASP | Лекція | 1, 5, 7, 9, 10 | 2 | |
| | 3 | Розробка завдання з використанням GRASP | Лабораторні заняття | 2, 3, 4, 6, 11 | 2 | Наступне лабораторні заняття |
| | 4 | Огляд принципів CQRS | Лекція | 1, 5, 7, 9, 10 | 2 | |
| | 4 | Розробка завдання з використанням CQRS | Лабораторні заняття | 2, 3, 4, 6, 11 | 2 | Наступне лабораторні заняття |
| | 5 | Проектування на основі SOA/Microservices/3tier/Embedded | Лекція | 1, 5, 7, 9, 10 | 2 | |
| | 5 | Застосування SOA/Microservices/3tier/Embedded у власному проекті | Лабораторні заняття | 2, 3, 4, 6, 11 | 2 | Наступне лабораторні заняття |
| | 6 | Проектування на основі SQL vs NoSQL | Лекція | 1, 5, 7, 9, 10 | 2 | |
| | 6 | Застосування SQL/noSQL у власному проекті | Лабораторні заняття | 2, 3, 4, 6, 11 | 2 | Наступне лабораторні заняття |
| | 7 | Огляд EAAP | Лекція | 1, 5, 7, 9, 10 | 2 | |
| | 7 | Розробка завдання з використанням EAAP | Лабораторні заняття | 2, 3, 4, 6, 11 | 2 | Наступне лабораторні заняття |
| | 8 | Огляд та порівняння підходів TDD vs DDD vs BDD | Лекція | 1, 5, 7, 9, 10 | 2 | |
| | 8 | Розробка завдання з використанням TDD/BDD/DDD | Лабораторні заняття | 2, 3, 4, 6, 11 | 2 | Наступне лабораторні заняття |
| | 9 | Розгортання сервісів на основі Cloud/On Premises/Serverless | Лекція | 1, 5, 7, 9, 10 | 2 | |
| | 9 | Розгортання власного проекту на основі Cloud/On Premises/Serverless | Лабораторні заняття | 2, 3, 4, 6, 11 | 2 | Наступне лабораторні заняття |
| | 10 | Проектування інтерфейсів на основі SPA/MPA | Лекція | 1, 5, 7, 9, 10 | 2 | |
| | 10 | Розробка завдання з використанням SPA/MPA | Лабораторні заняття | 2, 3, 4, 6, 11 | 2 | Наступне лабораторні заняття |
| | 11 | Проектування масштабованих систем на основі Stateless/Stateful/Virtual Actor | Лекція | 1, 5, 7, 9, 10 | 2 | |
| | 11 | Розробка завдання з використанням Stateless/Stateful/Virtual Actor | Лабораторні заняття | 2, 3, 4, 6, 11 | 2 | Наступне лабораторні заняття |
| | 12 | Огляд та порівняння підходів та інструментів збору аналітики та моніторингу | Лекція | 1, 5, 7, 9, 10 | 2 | |
| | 12 | Підключення інструментів збору аналітики та моніторингу до власного проекту | Лабораторні заняття | 2, 3, 4, 6, 11 | 2 | Наступне лабораторні заняття |
| | 13 | Проектування для моделювання потенційних загроз ПЗ і засобів захисту | Лекція | 1, 5, 7, 9, 10 | 2 | |
| | 13 | Розробка моделі загроз для власного проекту | Лабораторні заняття | 2, 3, 4, 6, 11 | 2 | Наступне лабораторні заняття |

| | | | | | | |
|---|---|---|-------------------------|----------------|--------------|----------------------------------|
| | 14 | Моделювання Resilience/Failover/Scalability. | Лекція | 1, 5, 7, 9, 10 | 2 | |
| | 14 | Розробка моделей RMA для власного проекту | Лабораторн е заняття | 2, 3, 4, 6, 11 | 2 | Наступне лабораторн е заняття |
| | 15 | Розгортання ПЗ та способів повернення до попередніх версій. | Лекція | 1, 5, 7, 9, 10 | 2 | |
| | 15 | Налаштування Release/Rollback для власного проекту. | Лабораторн е заняття | 8 | 2 | Наступне лабораторн е заняття |
| | 16 | Підсумкова лекція | Лекція | 1, 5, 7, 9, 10 | 2 | |
| | 16 | Демонстрація власного проекту | Лабораторн е заняття | | 2 | |
| Підсумковий контроль, форма | іспит в кінці семестру іспит - тестовий | | | | | |
| Пререквізити | Для вивчення курсу студенти потребують базових знань з дисциплін «Програмування», «Архітектура обчислювальних систем та комп'ютерна схемотехніка», «Комп'ютерні інформаційні мережі», «Алгоритми і структури даних», «Бази даних та інформаційні системи», «Програмування під UNIX - подібними системами», «Паралельні та розподілені обчислення». | | | | | |
| Навчальні методи та техніки, які будуть використовуватися під час викладання курсу | Лекції, презентація, лабораторні заняття у вигляді семінарів з мультимедійними презентаціями (в тому числі студентів); виконання лабораторних завдань, результатом яких є звіт в електронному або друкованому вигляді; самостійна робота з вивченням оприлюднених електронних матеріалів. | | | | | |
| Необхідне обладнання | Для проведення лекцій: комп'ютер, проектор. Для проведення лабораторних та виконання завдань: комп'ютер, ОС Windows, доступ до інтернету, безкоштовні/академічні варіанти Visual Studio Code, Azure DevOps/GitHub/BitBucket. За домовленістю з викладачем, індивідуальні завдання можуть виконуватися з використанням довільних прикладних програм чи мов програмування, зокрема Python, Java, C#, SAS тощо. Для оформлення звітів пропонується використовувати LibreOffice, чи MS Office 365, чи WinEdt. | | | | | |
| Критерії оцінювання (окремо для кожного виду навчальної діяльності) | Оцінювання проводиться за 100-бальною шкалою. | | | | | |
| | Оцінка за шкалою ECTS | | Оцінка в балах | | Екзам | |
| | A | Відмінно | 90-100 | | Відмінно | |
| | B | Дуже добре | 81-89 | | Добре | |
| | C | Добре | 71-80 | | | |
| | D | Задовільно | 61-70 | | Задовільно | |
| | E | Достатньо | 51-60 | | | |
| | F (FX) | Незадовільно | 0-50 | | Незадовільно | |
| | <p>Бали нараховуються за наступним співвідношенням:</p> <ul style="list-style-type: none"> Поточний контроль: практичні/самостійні тощо - 50% семестрової оцінки; максимальна кількість балів 50. На протязі семестру необхідно виконати усі завдання. Для кожного завдання встановлено терміни здачі. Роботи, які здаються із порушенням термінів без поважних причин, оцінюються на нижчу оцінку. Запізнення до 7 днів -50%, від 8 до 14 днів - 75%, більше 14 днів - 90%. Семестровий контроль: екзамен - 50% семестрової оцінки. Максимальна кількість балів 50 | | | | | |

Поточний контроль знань студентів здійснюється під час проведення лабораторних занять і має на меті перевірку виконання індивідуальних або командних завдань. Всього студенти отримують **5 індивідуальних та командних завдань, кожне з яких оцінюється 10 балами максимально**

Критерії оцінювання індивідуальних завдань

| Кількість балів | Критерій оцінювання |
|-----------------|---|
| 10 | студент повністю виконав умови завдання, алгоритми реалізовано правильно, при захисті роботи відповідає на всі запитання, пов'язані з тематикою завдання, проводить чіткий аналіз, порівняння та інтерпретацію отриманих результатів, пропонує інші підходи до вирішення поставленого завдання; |
| 7-9 | студент повністю виконав умови завдання, алгоритми реалізовано правильно, на деякі запитання, пов'язані з тематикою завдання відповідає з незначними неточностями, проводить аналіз, порівняння та інтерпретацію отриманих результатів з незначними неточностями; |
| 4-6 | студент виконав завдання з незначними помилками, проте самостійно їх виправляє та може пояснити, якщо на них вкаже викладач, на деякі запитання, пов'язані з тематикою завдання, відповідає з неточностями, проводить аналіз, порівняння та інтерпретацію отриманих результатів з неточностями; |
| 2-3 | студент виконав завдання частково, алгоритми реалізовано з помилками, які частково може виправити, якщо на них вкаже викладач, на запитання відповідає з помилками, проводить аналіз, порівняння та інтерпретацію отриманих результатів з помилками; |
| 1-2 | студент виконав завдання частково, алгоритм реалізовано з помилками, які самостійно не може виправити, переважно не відповідає на запитання, не здатний провести аналіз, порівняння та інтерпретацію отриманих результатів; |
| 0 | студент не володіє навчальним матеріалом і не виконав завдання |

Семестровий контроль проводять у формі іспиту для тих здобувачів, які за результатами роботи протягом семестру допущені до складання іспиту.

На іспиті студент повинен пройти тест, в який включено 10 питань з різних тем програми дисципліни. Відповіді до тестів студент вносить письмові, тестування проводиться в навчальній системі moodle. **Кожне питання оцінюється максимально 5 балами, максимальна кількість балів за тест – 50**

Жодні форми порушення академічної доброчесності не толеруються.

| | |
|---|---|
| <p>Індивідуальні та командні завдання для поточного контролю</p> | |
| <p>Питання до екзамену</p> | <ol style="list-style-type: none"> 1. Які умови необхідні для успішного застосування DDD 2. Для чого можна використовувати Domain Driven Design 3. При використанні TDD рефакторинг виконується 4. Чи підтримує база даних MongoDB відношення первинного-вторинного ключа 5. Яка стратегічна цінність SOA 6. Які базові принципи SOA 7. Як розшифровується SOA 8. Які характерні риси Embedded систем 9. Які існують об'єктно-реляційні шаблони поведінки(Object-Relational Behavioral Patterns) 10. Які існують шаблони стану сесії?(Session State Patterns) 11. Як визначається принцип єдиної відповідальності (The Single Responsibility Principle) 12. Як визначається принцип Відкрито-Закрито (The Open-Closed Principle) 13. Як визначається принцип поділу інтерфейсів (Interface Segregation Principle) 14. Як визначається принцип інверсії залежностей (The Dependency Inversion Principle) 15. Як визначається принцип єдиної відповідальності (The Single Responsibility Principle) 16. Як визначається принцип заміщення Барбари Лісков (The Liskov Substitution Principle) 17. Які приклади порушення принципу Відкрито-Закрито (The Open-Closed Principle) 18. Які приклади порушення принципу заміщення Барбари Лісков (The Liskov Substitution Principle) 19. Які приклади порушення принципу єдиної відповідальності (The Single Responsibility Principle) 20. Які приклади порушення принципу поділу інтерфейсів (Interface Segregation Principle) 21. Які приклади порушення принципу інверсії залежностей (The Dependency Inversion Principle) 22. Як визначається Конфіденційність (confidentiality) 23. Як визначається Цілісність (integrity) 24. Як визначається Доступність (availability) 25. Як визначається Апелювання (non-repudiation) 26. Як визначається Підзвітність (accountability) 27. Як визначається Достовірність (reliability) 28. Як визначається Автентичність (authenticity) 29. Які є принципи класифікації STRIDE 30. Що містить найвищий рівень сервісно-орієнтованого додатку 31. Який тип накладних витрат понесеться, коли віртуальне програмне забезпечення реалізовано у хмарі? 32. Яке твердження найкраще описує взаємозв'язок між програмою, сервером і клієнтом в хмарі? 33. Як хмарні обчислення пов'язані з віддаленим зберіганням даних 34. Що є одним з унікальних атрибутів Cloud Computing? |

| | |
|--------------------------|---|
| | <p>35. Яка модель розгортання хмари відповідатиме потребам підприємства, яке хоче контролювати розробку та доставку всіх хмарних сервісів?</p> <p>36. Яка роль віртуалізації у хмарних обчислень?</p> <p>37. Який тип бізнесу надає програмне забезпечення як послугу (Software as a Service)?</p> <p>38. Програмне забезпечення як послуга вважається включеним до того, який обчислювальний шар хмари?</p> <p>39. Який тип технології забезпечує використання ресурсів хмари для інвентаризації та управління компонентами інфраструктури?</p> <p>40. Який з наступних рівнів є найвищим рівнем інтеграції в хмарні обчислення?</p> <p>41. Який найважливіший компонент під час розробки та планування хмарного сервісу?</p> <p>42. Яка зі служб хмарних обчислень має розширення «mobile Lite»?</p> <p>43. Які можливості надання ресурсів (capacity provisioning capabilities) необхідні для операційного середовища в хмарі?</p> <p>44. Що таке паравіртуалізація?</p> <p>45. Як відбувається валідація даних в процесі проектування архітектури ПЗ?</p> <p>46. В чому полягає межа порушення принципу єдиності відповідальності God object?</p> <p>47. Які наслідки використання Interface Segregation Principle?</p> <p>48. Які переваги Single Responsibility Principle?</p> <p>49. Які недоліки міросервісної архітектури?</p> <p>50. Що таке мікросервісна архітектура?</p> <p>51. Які є відмінності між мікросервісною та монолітною архітектурами?</p> <p>52. Що таке безперервна інтеграція?</p> <p>53. Опишіть властивість Protected Variations</p> <p>54. Опишіть властивість Pure Fabrication</p> <p>55. Опишіть властивість Інформаційний експерт</p> <p>56. Опишіть 9 основних GRASP принципів</p> <p>57. Що таке GRASP?</p> <p>58. Яким чином відбувається перевірка типу абстракції?</p> <p>59. Опишіть Dependency Inversion Principle (Принцип інверсії залежностей)</p> |
| <p>Опитування</p> | <p>Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенню курсу.</p> |