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 31 серпня 2022 р.)

Зав. кафедри: к. ф.-м. н., доц. Ярошко С. А.

СИЛЛАБУС З НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
“Сучасні технології виконання програмних проектів”,
що викладається в межах ОПП другого (магістерського) рівня вищої освіти для здобувачів зі спеціальності
122 Комп’ютерні науки

Львів – 2022

Назва дисципліни	Сучасні технології виконання програмних проектів
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	Клакович Леся Миронівна, к. ф.-м. н., доцент, доцент кафедри програмування
Контактна інформація викладачів	*E-mail:* lesya.klakovych@lnu.edu.ua веб-сторінка: https://ami.lnu.edu.ua/employee/klakovych
Консультації з питань навчання по дисципліні відбуваються	В день проведення лекцій/лабораторних занять (за попередньою домовленістю та за умови проведення аудиторних занять); можливі онлайнові консультації в середовищі MS Teams. Для погодження часу онлайнових консультацій слід писати на електронну пошту викладача.
Сторінка курсу	https://ami.lnu.edu.ua/course/suchasni-tehnolohiji-vykonannya-prohramnyh-proektiv
Інформація про дисципліну	Курс розроблено таким чином, щоб надати учасникам необхідні знання, обов'язкові для керування та побудови програмних проектів. Розглядають складові життєвого циклу розробки програмного забезпечення (SDLC), методології та інструменти для виконання проектів з розробки програмного забезпечення, різні аспекти підготовки для здобуття кваліфікації: основні компоненти життєвого циклу; множина необхідних при розробці артефактів; ролі учасників розробки програмного забезпечення; методології для організації роботи над проектом; документи, інструменти, ролі, обов'язки та активності необхідні для розробки і реалізації програмних проектів.
Коротка анотація дисципліни	Дисципліна "Сучасні технології виконання програмних проектів" є нормативною дисципліною з спеціальності комп'ютерні науки, яка викладається в другому семестрі в обсязі 6 кредитів (за Європейською Кредитно-Трансферною Системою ECTS).
Мета та цілі дисципліни	Мета – формування базової системи знань та навиків для побудови програмних продуктів та управління проектами з розробки програмного забезпечення, потрібних для різних прикладних цілей.

	Цілі: формування знань та навиків про життєвий цикл розробки програмного забезпечення, його етапи, особливості, інструменти для візуалізації та документації роботи на кожному етапі, підходи для забезпечення якості програмного продукту; набуття компетенцій, знань, умінь та навиків відповідно до кваліфікації магістра за спеціальністю “Комп’ютерні науки”;
Література для вивчення дисципліни	*Основна:* 1. Applying UML and Patterns - ISBN 0131489062, 2004 2. Joseph Ingeno Software Architect's Handbook - ISBN: 9781788624060, 2018 3. Robert C. Martin. "Principles Of OOD". http://principles-wiki.net/collections:robert_c._martin_s_principle_collection *Додаткова:* 4. Systems and software engineering – Software Life Cycle Processes. ISO 12207:2008. – [Чинний від 2008-02-01] – II, 122 с.– (Міжнародний стандарт). 5. IEEE Standard Glossary of Software Engineering Terminology, Глосарій. IEEE Std 610.12-2010. – (Галузевий стандарт).
Обсяг курсу	64 години аудиторних занять. З них 32 годин лекцій, 32 годин лабораторних занять та 116 годин самостійної роботи
Очікувані результати навчання	У результаті вивчення даної навчальної дисципліни здобувач освіти буде *знати:* – предмет, методи та завдання дисципліни; – основні компоненти життєвого циклу розробки програмного забезпечення; – основні методології та інструменти розробки програмних проектів, їх сильні і слабкі сторони; – основні ролі, інструменти та артефакти процесу розробки програмного забезпечення; *вміти:* – обирати правильну методологію розробки програмного проекту і використовувати її переваги при розробці проектів – в команді розробляти якісні, документовані програмні проекти з використанням сучасних підходів та інструментів роботи;
Компетентності	Інтегральна компетентність: ІК, Загальні компетентності: ЗК 1, ЗК 2, ЗК4, ЗК 5 Спеціальні (фахові) компетентності: СК2, СК 5, СК 8, СК 13, СК 14
Програмні результати навчання	ПРН 1, ПРН2, ПРН3, ПРН4, ПРН 5, ПРН 10, ПРН13, ПРН14, ПРН 15, ПРН17 ПРН18
Ключові слова	розробка програмних проектів, життєвий цикл розробки

<table>
<tr><td></td><td>програмного забезпечення, якість, модель, команда</td></tr>
<tr><td>Формат курсу</td><td>очний</td></tr>
<tr><td></td><td>Проведення лекцій, лабораторних робіт та консультації для кращого розуміння тем</td></tr>
<tr><td>Теми</td><td>
<table>
<tr><th>Тиж.</th><th>Тема, план, короткі тези</th><th>Форма діяльності (заняття)* *лекція, самостійна, дискусія, групова робота)</th><th>Література.*** Ресурси в інтернеті</th><th>Завдання, год</th><th>Термін виконання</th></tr>
<tr><td>1</td><td>Методології розробки програмного забезпечення:послідовні, ітераційні, спіральні. Waterfall, Agile, Scrum .</td><td>лекція</td><td>1, 3, 5</td><td>2</td><td></td></tr>
<tr><td>1</td><td>Вибір методології, вибір теми проекту, (concept проекту), формування команд</td><td>лабораторне заняття</td><td>1,2, 4,5</td><td>2</td><td>Наступне лаб. заняття</td></tr>
<tr><td>2</td><td>Команда: ролі та відповідальності, комунікації.</td><td>лекція</td><td>1, 3, 5</td><td>2</td><td></td></tr>
<tr><td>2</td><td>Представлення ідеї проекту і розподіл ролей</td><td>лабораторне заняття</td><td>1,2, 4,5</td><td>2</td><td>Наступне лаб. заняття</td></tr>
<tr><td>3</td><td>Інструменти комунікації в проекті: Jira, Kanbanize, Trello, Github</td><td>лекція</td><td>1, 3, 5</td><td>2</td><td></td></tr>
<tr><td>3</td><td>Налагодження та використання інструментів Jira, Kanbanize, Trello, Github</td><td>лабораторне заняття</td><td>1,2, 4,5</td><td>2</td><td>Наступне лаб. заняття</td></tr>
<tr><td>4</td><td>Вимоги до проекту: види вимог,збір вимог, представлення.</td><td>лекція</td><td>1, 3, 5</td><td>2</td><td></td></tr>
<tr><td>4</td><td>Побудова документації для</td><td>лабораторне</td><td>1,2,,5</td><td>2</td><td>Наступне лаб.</td></tr>
</table>
</td></tr>
</table>

	обліку та пріоритизації вимог.	*заняття*			*заняття*
5	*Оформлення вимог у вигляді User story, SRS, use case*	*лекція*	*1, 3, 5*	*2 год*	
5	*Практика в написанні User story, SRS, use case*	*лабораторне заняття*	*1, 2, 3*	*2*	*Наступне лаб. заняття*
6	*Сучасні архітектурні рішення: Client-Server, MVC, MTV, MVVM, багаторівнева архітектура, REST*	*лекція*	*3, 5*	*2*	
6	*Розробка архітектури проекту, підбір технологій*	*лабораторне заняття*	*2, 4*	*2*	*Наступне лаб. заняття*
7	*Дизайн та архітектура проекту (OOD, OOP). Короткий огляд UML.*	*лекція*	*1, 2, 3*	*2*	
7	*Проектування DB, побудова та представлення діаграми компонент. діаграма сутностей і залежностей*	*лабораторне заняття*	*1, 5*	*2*	*До кінця лаб. заняття*
8	*SOLID, KISS, DRY та інші принципи проектування*	*лекція*	*1, 2, 3*	*2*	
8	*Розробка діаграми сутностей і залежностей, та діаграми послідовності*	*лабораторне заняття*	*1, 5*	*2*	*До кінця лаб. заняття*
9	*Розробка дизайну проекту, GUI. Основні принципи розробки дизайну із урахуванням потреб користувача, UX.*	*лекція*	*1, 2, 5*	*2*	
9	*Розробка*	*лаборатор*	*1, 2,4,5*	*2*	*До кінця*

	графічного інтерфейсу користувача	*не заняття*			*лаб. заняття*
10	*Представлення графічого інтерфейсу користувача з допомогою Scatch, WireFrame, Clicable prototype, Mockup*	*лекція*	*1, 3*	*2*	
10	*Розробка Scatch, WireFrame, Clicable prototype, Mockup щодо інтерфейсу проекту*	*лабораторне заняття*	*1, 3, 5*	*2*	*До кінця лаб. заняття*
11	*Побудова "чистого коду". Метрики. Інструменти*	*лекція*	*2, 3*	*2*	
11	*Застосування практик ’чистого коду’ у розробці проекту*	*лабораторне заняття*	*1, ,2, 5*	*2*	*До кінця лаб. заняття*
12	*Тестування проекту. Принципи тестування, типи тестувань*	*лекція*	*2, 3*	*2*	
12	*Застосування різних типів тестувань для розробки якісного продукту*	*лабораторне заняття*	*5*	*2*	*До кінця лаб. заняття*
13	*Рівні тестування. Ручне та автоматизоване тестування*	*лекція*	*1, 2, 3,5*	*2*	
13	*Розробка ручних тестів та їх виконання на проекті*	*лабораторне заняття*	*1, 5*	*2*	*До кінця лаб. заняття*
14	*Розробка тесткейсів, логування дефектів. Інструменти*	*лекція*	*1, 2, 3,5*	*2*	
14	*Логування та виправлення дефектів у проекті*	*лабораторне заняття*	*2, 4, 5*	*2*	*До кінця лаб. заняття*
15	*Побудова та налагодження*	*лекція*	*1, 2, 3,5*	*2*	

<table>
<tr><td></td><td>
<table>
<tr><td></td><td>автоматизованих тестів. REST Api тестування, Unit та інтеграційне тестування.</td><td></td><td></td><td></td><td></td></tr>
<tr><td>15</td><td>Розобка Unit та інтеграційних тестів. Автоматизація ручних тестів</td><td>лабораторне заняття</td><td>2, 4</td><td>2</td><td>До кінця лаб. заняття</td></tr>
<tr><td>16</td><td>Системи контролю версій та CI/CD</td><td>лекція</td><td>1, 2, 4</td><td>2</td><td></td></tr>
<tr><td>16</td><td>Налагодження CI/CD на проекті</td><td>лабораторне заняття</td><td>2, 5</td><td>2</td><td>До кінця лаб. заняття</td></tr>
</table>
</td></tr>
<tr><td>Підсумковий контроль, форма</td><td>екзамен - в кінці семестру
екзамен - тестовий</td></tr>
<tr><td>Пререквізити</td><td>Для вивчення курсу студенти потребують базових знань з дисциплін “Програмування”, “Бази даних”, “Програмна інженерія” .</td></tr>
<tr><td>Навчальні методи та техніки, які будуть використовуватися під час викладання курсу</td><td>Лекції, презентації, 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. Вивчення матеріалів веб-сайтів за темами занять. Аналітичний аналіз матеріалів. Семінари та консультації засобами системи Microsoft Teams (дистанційне навчання).</td></tr>
<tr><td>Необхідне обладнання</td><td>Для проведення лекцій: комп’ютер, проектор.
Для проведення лабораторних та виконання завдань: комп’ютер, ОС Windows/Linux, доступ до інтернету.</td></tr>
<tr><td>Критерії оцінювання (окремо для кожного виду навчальної діяльності)</td><td>Оцінювання проводиться за 100-бальною шкалою. Бали нараховуються за наступним співвідношенням:
• Поточний контроль: практичні/самостійні тощо - 50% семестрової оцінки; максимальна кількість балів 50
• 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 рівня підготовленості здобувача до виконання завдань за розділом, запланованим на окреме кожне заняття. Крім того, поточний контроль виконують для перевірки виконання студентом індивідуальних та командних завдань.
Всього студенти отримують 5 індивідуальних та командних</td></tr>
</table>

	завдань, кожне з яких оцінюється 10 балами максимально 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**
Питання до екзамену	1. Перерахувати основні етапи життєвого циклу розробки програмного забезпечення. 2. Перерахувати особливості етапу аналізу вимог до програмного забезпечення. 3. Функціональні та нефункціональні вимоги. Особливості формулювання нефункціональних вимог. 4. Проектування програмного забезпечення. Способи задання архітектури програмного продукту. 5. Використання UML для проектування програмного продукту. 6. Побудова плану виконання програмного проекту. Основні елементи плану виконання. 7. Вплив етапу проектування на процес уточнення вимог до програмного забезпечення. 8. Методи розробки програмного забезпечення через тестування. 9. Основні елементи тест-плану і його застосування. 10. Які типи автоматизованих тестів потрібно застосувати до певного програмного продукту. 11. Наведіть приклади модульних тестів для програмного продукту. 12. Побудуйте план впровадження програмного забезпечення з певними властивостями.. 13. Перерахуйте відмінності між впровадженням розподіленого web-продукту та мобільної аплікації. 14. Особливості супроводу і підтримки хмарних програмних продуктів. 15. Основні характеристики каскадного підходу при розробці програмного забезпечення. 16. Клас задач розробки, які потребують використання каскадного підходу. 17. Особливості ітераційного процесу розробки програмного забезпечення.. 18. Переваги і недоліки ітераційних процесів розробки програмного забезпечення. 19. Гнучкі методології. Основні класи задач для застосування гнучких підходів. 20. Основні цінності екстремального програмування як методології. 21. Ролі, артефакти та церемонії в Scrum. 22. Переваги та недоліки Kanban.
Опитування	Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенню курсу.