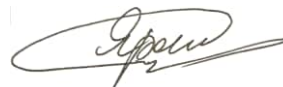


МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Львівський національний університет імені Івана Франка
Факультет прикладної математики та інформатики
Кафедра програмування

Затверджено

На засіданні кафедри програмування
факультету прикладної математики та
інформатики Львівського
національного університету імені Івана
Франка
(протокол № 1 від 31 серпня 2020 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 (ОПН) другого (магістерського) рівня
вищої освіти для здобувачів зі спеціалізації
014.09 Середня освіта (Інформатика)

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: 01 Освіта / Педагогіка Спеціальність: 014 Середня освіта Спеціалізація: 014.09 Середня освіта (Інформатика)
Викладачі дисципліни	Костів Василь Ярославович, старший викладач кафедри програмування;
Контактна інформація викладачів	<i>E-mail:</i> vasyl.kostiv@lnu.edu.ua <i>Веб-сторінки:</i> https://ami.lnu.edu.ua/employee/kostiv
Консультації з питань навчання по дисципліні відбуваються	В день проведення лекцій/лабораторних занять (за попередньою домовленістю та за умови проведення аудиторних занять); можливі онлайнві консультації в середовищі MS Teams. Для погодження часу онлайнвих консультацій слід писати на електронну пошту викладача.
Сторінка курсу	https://ami.lnu.edu.ua/academics/master
Інформація про дисципліну	Курс розроблено таким чином, щоб надати учасникам необхідні знання, обов'язкові для викладання розділів інформатики за програмою середньої загальноосвітньої школи в частині керування побудовою 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. Крім того, здобувачі отримують розширене коло знань про управління процесом розробки програмного забезпечення на рівні вищої освіти.
Коротка анотація дисципліни	Дисципліна “Сучасні технології виконання програмних проектів” є вибірковою дисципліною з спеціальності середня освіта (Інформатика) для освітньої програми середня освіта (Інформатика), яка викладається в першому семестрі в обсязі 4,5 кредитів (за Європейською Кредитно-Трансферною Системою

	ECTS).
Мета та цілі дисципліни	<p>Мета – формування базової системи знань та навиків для викладання розділів інформатики за програмою середньої загальноосвітньої школи в частині побудови програмних продуктів та управління проектами з розробки програмного забезпечення, потрібних для різних прикладних цілей і для побудови навчальних планів учнів.</p> <p>Цілі: формування уявлення про життєвий цикл розробки програмного забезпечення; набуття компетенцій, знань, умінь та навиків з викладання інформатики у середній школі, відповідно до кваліфікації магістра за спеціальністю “Середня освіта (інформатика)”; вивчення прийомів викладання розділів інформатики в частині загальної розробки програмного забезпечення за програмою середньої загальноосвітньої школи; вивчення методів організації роботи учнів школи для цілей засвоєння методів управління проектами з розробки програмного забезпечення; вивчення сучасних прийомів формування теоретичних знань і практичних навичок учнів.</p>
Література для вивчення дисципліни	<p><i>Основна:</i></p> <ol style="list-style-type: none"> 1. Підручники інформатики для середніх загальноосвітніх шкіл. 2. Бабенко Л. П., Лавріщева К. М. <i>Основи програмної інженерії</i>. - К: Знання. - 2001. - 269с. <p><i>Додаткова:</i></p> <ol style="list-style-type: none"> 3. Иан Коммервилл. Инженерия программного обеспечения. Изд. дом “Вильямс”. – 2002 г. – 624 с. 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 610.12-1990. – (Галузевий стандарт). 6. Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения / Спб.: Питер, 2002. - 496 с. 7. Електронний ресурс - SCRUM Україна - https://www.scrum.ua/materials?locale=uk
Обсяг курсу	135 годин. З них: 40 (14) годин аудиторних занять – 16 (6) годин лекцій, 24 (8) годин лабораторних занять; 95 (121) годин самостійної роботи.
Очікувані результати навчання	<p>У результаті вивчення даної навчальної дисципліни здобувач освіти буде</p> <p><i>знати:</i></p> <ul style="list-style-type: none"> – предмет, методи та завдання дисципліни;

	<ul style="list-style-type: none"> – прийоми викладання розділів інформатики в СЗШ в частині життєвого циклу розробки програмного забезпечення; – основні компоненти життєвого циклу; – основні методології та інструменти розробки програмних проектів, їх сильні і слабкі сторони; – основні ролі, інструменти та артефакти процесу розробки програмного забезпечення; <p><i>вміти:</i></p> <ul style="list-style-type: none"> – обирати правильну методологію розробки програмного проекту; – будувати уроки інформатики різного типу для різних підходів розробки програмного забезпечення; – застосовувати до конкретних прикладних задач відповідну методологію; – застосовувати різні методи контролю знань учнів. 																				
Ключові слова	інформатика, життєвий цикл розробки програмного забезпечення, модель, програмування																				
Формат курсу	очний (заочний)																				
	Проведення лекцій, лабораторних робіт та консультації для кращого розуміння тем																				
Теми	<table border="1"> <thead> <tr> <th>№</th> <th>Тема, план, короткі тези</th> <th>Форма діяльності</th> <th>Завдання, год</th> <th>Термін виконання</th> </tr> </thead> <tbody> <tr> <td>1</td> <td><i>Розробка програмного забезпечення як процес. Етапи розробки програмного забезпечення: Предмет та об'єкт дисципліни. Розробка програмного забезпечення як базовий процес інформаційних технологій. Історичні передумови формалізації процесу управління програмними проектами..</i></td> <td><i>лекція</i></td> <td>2 (1)</td> <td></td> </tr> <tr> <td>1</td> <td><i>Аналіз початкових вимог до програмного забезпечення.</i></td> <td><i>лабораторне заняття</i></td> <td>2 (1)</td> <td><i>Наступне лаб. заняття</i></td> </tr> <tr> <td>2</td> <td><i>Поняття про життєвий цикл розробки програмного забезпечення: етапи роботи над проектом, їх основні властивості,</i></td> <td><i>лекція</i></td> <td>2 (1)</td> <td></td> </tr> </tbody> </table>	№	Тема, план, короткі тези	Форма діяльності	Завдання, год	Термін виконання	1	<i>Розробка програмного забезпечення як процес. Етапи розробки програмного забезпечення: Предмет та об'єкт дисципліни. Розробка програмного забезпечення як базовий процес інформаційних технологій. Історичні передумови формалізації процесу управління програмними проектами..</i>	<i>лекція</i>	2 (1)		1	<i>Аналіз початкових вимог до програмного забезпечення.</i>	<i>лабораторне заняття</i>	2 (1)	<i>Наступне лаб. заняття</i>	2	<i>Поняття про життєвий цикл розробки програмного забезпечення: етапи роботи над проектом, їх основні властивості,</i>	<i>лекція</i>	2 (1)	
№	Тема, план, короткі тези	Форма діяльності	Завдання, год	Термін виконання																	
1	<i>Розробка програмного забезпечення як процес. Етапи розробки програмного забезпечення: Предмет та об'єкт дисципліни. Розробка програмного забезпечення як базовий процес інформаційних технологій. Історичні передумови формалізації процесу управління програмними проектами..</i>	<i>лекція</i>	2 (1)																		
1	<i>Аналіз початкових вимог до програмного забезпечення.</i>	<i>лабораторне заняття</i>	2 (1)	<i>Наступне лаб. заняття</i>																	
2	<i>Поняття про життєвий цикл розробки програмного забезпечення: етапи роботи над проектом, їх основні властивості,</i>	<i>лекція</i>	2 (1)																		

	інструменти, артефакти, ролі. Виконавці та зацікавлені особи проекту.			
2	Побудова документу Бачення програмного продукту на основі аналізу базових вимог	лабораторне заняття	2 (1)	Наступне лаб. заняття
3	Аналіз вимог: Джерела вимог до програмного продукту, функціональні та нефункціональні вимоги до програмного забезпечення. Уточнення і зміна вимог. Інструменти для аналізу вимог. Основні типи документів.	лекція	2 (1)	
3	Побудова документації для обліку та пріоритизації вимог.	лабораторне заняття	4 (1)	Наступне лаб. заняття
4	Проектування та розробка. Забезпечення якості: Підходи для проектування програмного забезпечення, ітеративне уточнення вимог, аналіз вимог через проектування. Планування розробки. Тестування програмного забезпечення: тест-плани, тестові модулі, тести-затвердження.	лекція	2 (1)	
4	Приклади планів розробки програмного продукту. Вплив особливостей проектування на зміни до вимог.	лабораторне заняття	2 (1)	Наступне лаб. заняття
5	Впровадження та підтримка: особливості розгортання різних типів програмного забезпечення, необхідні ролі для кожного типу розгортання. Особливості впровадження	лекція	4 (1)	

	<p><i>продуктів у різних середовищах. Поняття підтримки і супроводу програмного продукту.</i></p> <p>5 <i>Побудова плану розгортання програмного продукту.</i> лабораторне заняття 4 (1) Наступне лаб. заняття</p> <p>6 <i>Методології та інструменти розробки: каскадна, ітераційна та гнучка методології, як базові підходи в інженерії програмного забезпечення. Особливості каскадного підходу і клас задач для його використання. Переваги ітераційного підходу. Особливості гнучких методологій і їх застосування. Agile. Scrum. Lean. Kanban.</i> лекція 4 (1)</p> <p>6 <i>Вибір методології процесу розробки продукту та уточнення необхідних артефактів.</i> лабораторне заняття 2 (1) До кінця лаб. заняття</p>
Підсумковий контроль, форма	екзамен
Пререквізити	Для вивчення курсу студенти потребують базових знань з дисциплін “Алгоритми та структури”, “Програмування”.
Навчальні методи та техніки, які будуть використовуватися під час викладання курсу	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).
Необхідне обладнання	Для проведення лекцій: комп’ютер, проектор. Для проведення лабораторних та виконання завдань: комп’ютер, ОС Windows/Linux, доступ до інтернету.
Критерії оцінювання (окремо для кожного виду навчальної	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

<p>діяльності)</p>	<p>контроль виконують для перевірки виконання студентом індивідуальних завдань. Форми поточного контролю:</p> <ul style="list-style-type: none"> – усне індивідуальне опитування; – письмове загальне опитування у формі задач; – опитування у формі тестування; – письмова контрольна робота; – перевірка плану виконання з теми лабораторного заняття; – експертна оцінка виконання завдань і отримання достовірних результатів; – тестування результатів програмної реалізації завдань; – перевірка правильності виконання індивідуальних завдань, виданих на самостійне опрацювання; – захист студентом індивідуальних завдань. <p>Максимальна кількість балів за поточний контроль – 50.</p> <p>Семестровий контроль проводять у формі іспиту для тих здобувачів, які за результатами роботи протягом семестру допущені до складання іспиту. Він включає в себе результати поточного контролю знань студентів, що проводили протягом семестру, та екзаменаційний контроль. Екзаменаційний контроль проводиться за розкладом, визначеним навчальним графіком для складання екзаменів. На іспиті студент отримує декілька питань – по одному з різних тем програми дисципліни. Кількість питань визначають перед екзаменаційною сесією. До кожного питання потрібно скласти письмову відповідь у розгорнутій формі з демонстрацією основних положень відповіді на конкретних прикладах. Відповідь перевіряється шляхом експертного оцінювання і співбесіди за змістом відповіді.</p> <p>Максимальна кількість балів за екзамен – 50. Максимальна кількість балів семестрового контролю – 100.</p>
<p>Питання до заліку чи екзамену</p>	<ol style="list-style-type: none"> 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.

	<ol style="list-style-type: none"> 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.
<p>Опитування</p>	<p>Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенню курсу.</p>