

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
Львівського національного університету
імені Івана Франка
(протокол № 1 від 29 серпня 2023 р.)



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

Силабус навчальної дисципліни «Програмна Інженерія»,
що викладається в межах ОПП 014.09 Середня освіта (Інформатика)
галузі знань: 01 Освіта / Педагогіка
першого (бакалаврського) рівня вищої освіти
для здобувачів зі спеціальності 014.09 Середня освіта (Інформатика)

Львів 2023 р.

Назва дисципліни	Програмна інженерія
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: 014.09 Середня освіта (Інформатика)
Викладачі дисципліни	Глова Андрій Романович, доктор філософії
Контактна інформація викладачів	Електронна пошта: andrii.hlova@lnu.edu.ua веб-сторінка: https://ami.lnu.edu.ua/employee/hlova-a-r
Консультації з питань навчання по дисципліні відбуваються	Консультації проводять раз на тиждень згідно з оприлюдненим розкладом консультацій викладача. Можливі он-лайн консультації через Zoom чи Microsoft Teams. Для погодження часу он-лайн консультацій слід писати на електронну пошту викладача.
Сторінка курсу	https://ami.lnu.edu.ua/course/prohramna-inzheneriia
Інформація про дисципліну	Курс "Програмна Інженерія" є нормативною дисципліною зі спеціальності 014.09 Середня освіта (Інформатика) для освітньої програми «Середня освіта (Інформатика)», яку викладають у п'ятому і шостому семестрах в обсязі 9 кредитів (за Європейською кредитно-трансферною системою ECTS)
Коротка анотація дисципліни	Фокус уваги курсу спрямовано на те, щоб навчити студентів розробляти програмні продукти з урахуванням сучасних підходів до програмної інженерії: планування, проектування, кодування, тестування та підтримки і налагодження програм. Використовуючи мову програмування C# та технології бібліотеки .Net, студенти здобудуть навички розробки десктопних програм під операційну систему Microsoft Windows та веб застосунків. Основні активності, що становлять суть програмної інженерії, у першому семестрі вивчаються та практично застосовуються на прикладі таких технологій, як ADO.Net, Entity FW, WPF, а у другому семестрі -- ASP.NET Core MVC.
Мета та цілі дисципліни	Метою нормативної дисципліни «Програмна інженерія» є навчити студента: <ul style="list-style-type: none"> ● застосовувати сучасні підходи до планування, дизайну, кодування, тестування та підтримки і налагодження програм ● створювати та аналізувати вимоги до програмних продуктів ● використовувати діаграми UML для відображення вимог до програм та опису архітектури та дизайну програм ● організувати розробку програми, з допомогою однієї з методологій SDLC ● писати код згідно з основними принципами об'єктно-орієнтованого проектування та підходами 'чистого коду' ● розробляти десктопні програми для Windows з допомогою технології WPF ● розробляти веб програми з допомогою технології ASP.Net ● розробляти програми з доступом до баз даних через технологію ADO.Net та Entity FW ● розробляти ручні та автоматизовані тести для гарантії якості коду ● структуровано передавати здобуті теоретичні знання та практичний досвід іншим
Література для вивчення дисципліни	<i>Основна література</i> <ol style="list-style-type: none"> 1. Matthew MacDonald. Pro WPF 4.5 in C#. Windows Presentation Foundation in .NET 4.5, "Apress", 2012. -1078 2. Adam Freeman. Pro ASP.NET Core MVC 2 7th Edition, "Apress", 2017. -1451 3. Windows Presentation Foundation documentation https://docs.microsoft.com/en-us/dotnet/desktop/wpf/?view=netdesktop-5.0 4. Steve Smith. Architecting Modern Web Applications with ASP.NET Core and Azure. EDITION v5.0 -Updated to ASP.NET Core 5.0 --Redmond, Washington, 98052-6399 – 2021, Microsoft Corporation. 5. Architectural principles https://docs.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/architectural-principles#separation-of-concerns 6. Robert Martin. Clean architecture. 7. https://herbertograca.com/2017/07/03/the-software-architecture-chronicles/ 8. ASP .NET Core fundamentals. https://learn.microsoft.com/en-us/aspnet/core/fundamentals/?view=aspnetcore-7.0&tabs=windows 9. Test ASP.NET Core MVC apps https://docs.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/test-asp-net-core-mvc-apps

	<p>10. Gamma Erich, Helm Richard, Johnson Ralph, Vlissides John. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 1995.</p> <p>11. C# Design Patterns https://www.dofactory.com/net/design-patterns</p> <p><i>Додаткова література</i></p> <ol style="list-style-type: none"> 1. Russ Miles, Kim Hamilton. Learning UML 2.0: A Pragmatic Introduction to UML - 2006, 290 р 2. Get started with ASP.NET Core MVC https://docs.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/start-mvc?view=aspnetcore-5.0&tabs=visual-studio 3. Robert Martin. Clean Code. 4. Hands-on cloud administration in Azure https://www.amazon.com/Hands-Cloud-Administration-Azure-components/dp/178913496X
Обсяг курсу	9 кредитів ЄКТС – 270 годин. З них 32 + 32 годин лекцій, 32 + 32 години лабораторних занять (5-й + 6-й семестр відповідно) та 142 годин самостійної роботи.
Очікувані результати навчання	<p>Після завершення цього курсу студент буде:</p> <p><i>знати</i></p> <ul style="list-style-type: none"> ● сучасні підходи до планування, дизайну, кодування, тестування та підтримки і налагодження програм ● способи представлення вимог до програмних продуктів ● різні типи UML діаграм ● життєвий цикл розробки програм ● основні принципи ООД та підходи 'чистого коду' <p><i>вміти</i></p> <ul style="list-style-type: none"> ● розробляти десктопні програми для Windows з допомогою технології WPF ● розробляти веб програми з допомогою технології ASP.Net ● розробляти програми з доступом до баз даних через технологію ADO.Net та Entity FW ● розробляти ручні та автоматизовані тести для гарантії якості коду ● структуровано передавати здобуті теоретичні знання та практичний досвід іншим
Компетентності	<p>ІК Здатність розв'язувати складні специфічні проблеми та практичні завдання в галузі середньої освіти, що передбачають застосування теорій та методів педагогічних наук та інформаційних технологій і характеризується комплексністю та мінливістю педагогічних умов організації освітнього процесу в основній (базовій) загальноосвітній школі ЗК4. Здатність спілкуватися державною мовою як усно, так і письмово.</p> <p>СК9. Здатність здійснювати формалізований опис задач дослідження операцій в організаційно-технічних і соціально-економічних системах різного призначення, визначати їх оптимальні розв'язки, будувати моделі оптимального управління з урахуванням змін економічної ситуації, оптимізувати процеси управління в системах різного призначення та рівня ієрархіїСК14. Здатність забезпечити організацію обчислювальних процесів в інформаційних системах різного призначення з урахуванням архітектури, конфігурування, показників результативності функціонування операційних систем і системного програмного забезпечення.</p>
Програмні результати навчання	<p>ПР2. Знати основні психолого-педагогічні теорії навчання, інноваційні технології навчання інформатики, актуальні проблеми розвитку педагогіки та методики навчання інформатики. Аналізувати світоглядні проблеми, приймати рішення на основі сформованих ціннісних орієнтирів.</p> <p>ПР8. Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук</p>
Ключові слова	Програмування мовою C#, .Net технології, WPF, ADO.Net, Entity FW, ASP.Net, MVC, десктопні програми, веб застосунки.
Формат курсу	Очний: проведення лекцій, лабораторних робіт та консультацій в приміщеннях університету, а в умовах карантину – онлайнний на платформі Microsoft Teams

Теми				
Тиж д.	Тема, план, короткі тези	Форма заняття	Тривалість, год	Термін виконання
1.1	Життєвий цикл розробки програм (SDLC). Ролі в команді	Лекція	2	
	Формування команд, вибір теми проекту, вибір методології розробки проекту	Лабораторна робота	2	Наступне лабораторне заняття
1.2	Вимоги до програмних продуктів. Збір, типи, аналіз вимог	Лекція	2	
	Збір вимог щодо проекту, їх документування	Тест	2	Наступне лабораторне заняття
1.3	UML, основні види діаграм, які використовуються при проектуванні та розробці програм	Лекція	2	
	Застосування UML діаграм для опису проекту та його розробки	Лабораторна робота	2	Наступне лабораторне заняття
1.4	Принципи ООД - SOLID, KISS	Лекція	2	
	Використання StyleCop та Code Analysis для підтримки конвенцій коду С#	Лабораторна робота	2	Наступне лабораторне заняття
1.5	Аналіз коду інструментами StyleCop та Code Analysis	Лекція	2	
	Опис вимог інтерфейсу користувача - мокапи, вайфрейми	Лабораторна робота	2	Наступне лабораторне заняття
1.6	Технології ADO.NET доступу до даних. Провайдери даних	Лекція	2	
	Проектування бази даних, архітектурні діаграми	Тест	2	
1.7	Під'єднаний режим роботи з базою даних. Рядок з'єднання. команди, SqlDataReader	Лекція	2	
	Від'єднаний режим роботи з базою даних. DataAdapter. Генерація бази даних, доступ через ADO.Net	Лабораторна робота	2	Наступне лабораторне заняття
1.8	Entity Framework, підходи DBFirst, ModelFirst, Code First Патерн Репозиторій та UnitOfWork	Лекція	2	
	Створення трьох рівневої архітектури проекту. Розробка рівня доступу до бази даних з допомогою Entity FW	Лабораторна робота	2	Наступне лабораторне заняття
1.9	Принципи та інструменти розробки інтерфейсу користувача, UX	Лекція	2	
	Розробка презентаційного рівня проекту, Графічний інтерфейс за допомогою бібліотеки WPF	Лабораторна робота	2	Наступне лабораторне заняття
1.10	Архітектура WPF. Ієрархія основних класів	Лекція	2	

	Додавання в програму необхідної графіки, анімації	Лабораторна робота	2	Наступне лабораторне заняття
1.11	ХАМЛ для розробки GUI. Обробка подій Розробка рівня бізнес логіки проекту	Лекція Лабораторна робота	2 2	Наступне лабораторне заняття
1.12	Елементи керування WPF. Data Binding. Стили, шаблони WPF Покриття коду юніт тестами	Лекція Лабораторна робота	2 2	Наступне лабораторне заняття
1.13	Розробка WPF програм з використанням патерну MVVM Застосування патерну MVVM в WPF частині програми	Лекція Тест	2 2	
1.14	Inversion Of Control. Dependency Injection. Unity Логування подій, зауважень та помилок	Лекція Лабораторна робота	2 2	Наступне лабораторне заняття
1.15	Тестування програмного забезпечення Презентація та захист проекту	Лекція Лабораторна робота	2 2	Наступне лабораторне заняття
1.16	Logger. Логування подій, зауважень та помилок Презентація та захист проекту	Лекція Лабораторна робота	2 2	
2.1	Підходи до проектування складних систем. Класифікація вимог до ПЗ та відношення між ними. Створення проектів. Налаштування середовища	Лекція Лабораторна робота	2 2	Наступне лабораторне заняття
2.2	Процес розробки вимог: виявлення, аналіз, специфікація, валідація Формування переліку вимог до Web-аплікацій	Лекція Лабораторна робота	2 2	Наступне лабораторне заняття
2.3	Архітектура ПЗ: моделювання, архітектурні стилі Специфікація та валідація вимог. Use-case моделі	Лекція Лабораторна робота	2 2	Наступне лабораторне заняття
2.4	Архітектура ASP Core MVC Web-аплікацій. Контролери. Виокремлення підсистем, затвердження архітектури	Лекція Лабораторна робота	2 2	Наступне лабораторне заняття

	2.5	Особливості розробки методів контролера	Лекція	2	
		Розробка моделі предметної області	Лабораторна робота	2	Наступне лабораторне заняття
	2.6	Основи проектування Views	Лекція	2	
		Unit-тестування. Імплементация контролерів для окремої підсистеми	Лабораторна робота	2	Наступне лабораторне заняття
	2.7	Проектування моделей предметної області. Використання БД	Лекція	2	
		Імплементация контролерів для окремої підсистеми	Лабораторна робота	2	Наступне лабораторне заняття
	2.8	Проектування типізованих Views	Лекція	2	
		Імплементация методів та їхніх тестів для основної функціональності. Вдосконалення Views.	Лабораторна робота	2	Наступне лабораторне заняття
	2.9	Оптимізація методів контролера	Лекція	2	
		Завершення імплементации методів та їхніх тестів для основної функціональності	Лабораторна робота	2	Наступне лабораторне заняття
	2.10	ASP.NET Core Identity	Лекція	2	
		Рефакторинг з врахуванням Identity	Лабораторна робота	2	Наступне лабораторне заняття
	2.11	Класифікація атрибутів якості ПЗ. Техніки забезпечення якості (QA).	Лекція	2	
		Завершення імплементации повного набору юскейсів	Лабораторна робота	2	Наступне лабораторне заняття
	2.12	Якість процесу розробки ПЗ	Лекція	2	
		Вдосконалення UI/UX	Лабораторна робота	2	Наступне лабораторне заняття
2.13	Патерни проектування GoF (DP). Структурні DP.	Лекція	2		
	Вдосконалення UI/UX	Лабораторна робота	2	Наступне лабораторне заняття	
2.14	Твірні DP	Лекція	2		
	Демонстрація командами Web-аплікацій та ревію стосовно параметрів якості.	Лабораторна робота	2	Наступне лабораторне заняття	
2.15	Поведінкові DP.	Лекція	2		
	Демонстрація командами Web-аплікацій та ревію стосовно параметрів якості.	Лабораторна робота	2	Наступне лабораторне заняття	
2.16	Патерни рефакторингу	Лекція	2		

		Демонстрація командами Web-аплікацій та ревізії стосовно параметрів якості.	Лабораторна робота	2	
Підсумковий контроль, форма	іспит в кінці 6 семестру				
Пререквізити	Для вивчення курсу студенти потребують базових знань з дисциплін “Програмування”, “Бази даних”, “ Організація та обробка електронної інформації ”. Одночасно з вивченням цього курсу, студенти в команді розробляють проекти з використанням сучасних підходів до розробки програм				
Навчальні методи та техніки, які використовують під час викладання курсу	Лекції з мультимедійними презентаціями та з демонстрацією прийомів практичного використання середовища програмування; лабораторні заняття у вигляді розробки проектів у команді з 3-5 осіб; самостійне опрацювання навчальних матеріалів: підручників, конспектів лекцій, готових програм мовою C#, додаткових навчальних посібників, розміщених у хмарному сховищі (Moodle, Microsoft Teams, Google Classroom). Обговорення теоретичного та практичного матеріалу в онлайн сервісах, формулювання творчих завдань для студентів, виконання яких готує до вивчення нового теоретичного матеріалу.				
Необхідне обладнання	Для проведення лекцій: комп’ютер, проектор, доступ до мережі інтернет. Для проведення лабораторних та виконання завдань: комп’ютер, ОС Windows, доступ до інтернету, середовище розробки програм технологіями .Net C# (Microsoft Visual Studio, Code Blocks тощо). 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.				
Критерії оцінювання (окремо для кожного виду навчальної діяльності)	<p>Оцінювання проводиться за 100-бальною шкалою. У п’ятому семестрі до 70 балів нараховується за розробку проекту, ще до 30 балів за знання теоретичної частини курсу. У шостому семестрі до 50 балів нараховують за розробку проекту, ще до 50 балів – за виконання екзаменаційного завдання. Розробка проекту поділена на 6-8 частин, кожна з яких оцінюється 5-10 балів залежно від складності. Оцінка за екзаменаційне завдання може бути поділена на дві частини: до 30 балів за розробку нової функціональності програми з оновленням відповідних моделей та специфікацій і 20 балів за засвоєння матеріалу усіх розділів курсу, з яких частина може бути виставлена в процесі опитувань упродовж семестру (у формі тестувань, колоквіумів тощо) .</p> <p>Відвідання занять є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції і лабораторні заняття курсу. Активність під час проведення лекцій і лабораторних заохочується балами. У будь-якому випадку студенти зобов’язані дотримуватися усіх строків визначених для виконання усіх видів письмових робіт, передбачених курсом. Виконані роботи завантажують у відповідне хмарне сховище. За відповідних обставин альтернативою відвідування лабораторних занять в університеті може бути дистанційна онлайн робота за розкладом проведення занять. Активність на лекціях і лабораторних ураховують при оцінюванні відповідного лабораторного завдання.</p> <p>Академічна доброчесність: роботи студентів мають бути їхніми оригінальними дослідженнями, розробками чи міркуваннями. Відсутність посилань на використані джерела, фабрикування джерел, списування, втручання в роботу інших студентів, здавання чужих комп’ютерних програм як своїх становлять, але не обмежують, приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в роботі студента є підставою для її незарахування викладачем, незалежно від масштабів плагіату чи обману.</p>				
Опитування	Анкету-оцінку з метою оцінювання якості курсу буде надано після завершення курсу.				