

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 2023 р.)



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

Силабус навчальної дисципліни
«Програмне забезпечення II»,
що викладається в межах ОПП «Прикладна математика»
для здобувачів першого (бакалаврського) рівня вищої освіти
з спеціальності 113 Прикладна математика

Львів 2023 р.

Назва дисципліни	Програмне забезпечення II
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 11 Математика та статистика Спеціальність 113 Прикладна математика
Викладачі дисципліни	Пасічник Тимофій Васильович, кандидат фізико-математичних наук, ст. наук. співробітник, доцент кафедри програмування
Контактна інформація викладачів	Електронна пошта: tymofiy.pasichnyk@lnu.edu.ua веб-сторінка: https://ami.lnu.edu.ua/employee/pasichnyk-t-v
Консультації з питань навчання по дисципліні відбуваються	Консультації проводять раз на тиждень згідно з оприлюдненим розкладом консультацій викладача. Можливі он-лайн консультації через Microsoft Teams. Для погодження часу он-лайн консультацій слід писати на електронну пошту викладача.
Сторінка курсу	https://ami.lnu.edu.ua/course/software-applied-mathematics
Інформація про дисципліну	Курс «Програмне забезпечення II» є нормативною дисципліною зі освітньо-професійної програми «Прикладна математика» першого (бакалаврського) рівня вищої освіти з предметної спеціальності 113 Прикладна математика. У п'ятому семестрі її викладають в обсязі 5 кредитів (за Європейською кредитно-трансферною системою ECTS)
Коротка анотація дисципліни	Фокус уваги курсу спрямовано на оволодіння базовими інструментами середовища .Net та особливості мови програмування C#: вивчення програмування Web-додатків в ASP.NET, створення графічних інтерфейсів у WPF, роботу з даними в ADO.NET та WCF.
Мета та цілі дисципліни	Метою нормативної дисципліни «Програмне забезпечення II» є ознайомити студента з базовими можливостями платформи .Net, навчити його використовувати мову C# для створення Web-додатків, графічних інтерфейсів, та роботи з різними базами даних.
Література для вивчення дисципліни	<i>Основна література</i> 1. Learn C#. OFFICIAL COLLECTION: https://learn.microsoft.com/enus/users/dotnet/collections/yz26f8y64n7k07?WT.mc_id=dotnet-35129-website 2. C# programming guide https://learn.microsoft.com/en-us/dotnet/csharp/programmingguide/?WT.mc_id=dotnet-35129-website 3. Object-oriented programming (C#) https://learn.microsoft.com/enus/dotnet/csharp/fundamentals/tutorials/oop?source=recommendations 4. Inheritance in C# and .NET https://learn.microsoft.com/enus/dotnet/csharp/fundamentals/tutorials/inheritance 5. Language Integrated Query (LINQ) https://learn.microsoft.com/en-us/dotnet/csharp/programmingguide/concepts/linq 6. Troelsen Andrew, Japikse Phil. Pro C# 10 with .NET 6: Foundational Principles and Practices in Programming – APress, 2022 – 1640 pp. 7. Christian Nagel Professional C# and .NET, 2021st Edition – Wrox, 2021 – 1800 pp. <i>Додаткова література</i> 8. Mark J. Price C# 10 and .NET 6 – Modern Cross-Platform Development – Packt Publishing, 2021 – 826 pp. 9. Joseph Albahari C# 10 in a Nutshell: The Definitive Reference – O'Reilly Media, 2022 – 1000 pp. 10. Ian Griffiths C# 10 in a Nutshell – O'Reilly Media, 2022 – 833 pp.

	11. Rob Miles Exam Ref 70-483 Programming in C#, Second Edition – Microsoft Press, 2018 – 433 pp.				
Обсяг курсу	5 кредитів ЄКТС – 150 годин. З них 32 години лекцій, 32 години лабораторних занять та 86 годин самостійної роботи				
Очікувані результати навчання	<p>Після завершення цього курсу студенти оволодіють знаннями про:</p> <ul style="list-style-type: none"> • Об'єктно-орієнтоване програмування; • Середовище .NET; • мову програмування C#; • середовище даних ADO.NET; • графічні інтерфейси WPF; • створення Web-додатків в ASP.NET. <p>Студенти мають оволодіти вміннями:</p> <ul style="list-style-type: none"> • застосувати практичні навички для програмного проектування; • створювати програмні продукти для вирішення завдань за фахом; • застосувати сучасні інформаційні технології; • здійснювати саморозвиток, самоосвіту, самоорганізацію <p>Курс забезпечує набуття таких компетентностей та програмних результатів навчання:</p> <p>Фахові компетентності спеціальності:</p> <p>ФК04. Здатність розробляти алгоритми та структури даних, програмні засоби та програмну документацію.</p> <p>ФК05. Здатність проектувати бази даних, інформаційні системи та ресурси.</p> <p>ФК08. Здатність використовувати сучасні технології програмування та тестування програмного забезпечення.</p> <p>ФК21. Здатність аналізувати можливості наявних технологій розробки програмного забезпечення, обирати та застосовувати найбільш ефективний інструментарій відповідно до специфіки завдання.</p> <p>Програмні результати навчання:</p> <p>РН11. Вміти застосовувати сучасні технології програмування та розроблення програмного забезпечення, програмної реалізації чисельних і символічних алгоритмів.</p> <p>РН23. Порівнювати, обирати та застосовувати інструментарій для розробки програмного забезпечення, виходячи з міркувань ефективності відповідно до специфіки завдання.</p>				
Ключові слова	Платформа .Net, мова програмування C#, клас, інтерфейс, властивість, подія, делегат, ASP.NET, багатопотоковість, ADO/NET, WPF.				
Формат курсу	Очний.				
Теми	Тижд	Тема, план, короткі тези	Форма заняття	Тривалість, год	Термін виконання
	1	Програмування з використанням елементів управління WPF. Синтаксис XAML для WPF.	Лекція	2	
		Програмування з використанням елементів управління WPF	Лабораторна робота	2	Наступне лабораторне заняття
		Ознайомлення з середовищем WPF	Самостійна робота	5	
	2	Побудова додатків WPF з використанням Visual Studio. Опрацювання подій. Об'єкти внутрішніх команд..	Лекція	2	
		Опрацювання подій. Об'єкти внутрішніх команд	Контрольна робота	2	
		Побудова додатків WPF.	Самостійна робота	5	
	3	Проектування інтерфейсу користувача аплікації. Застосування панелей, сіток. Стили, шаблони, ресурси..	Лекція	2	
		Проектування інтерфейсу користувача аплікації.	Лабораторна робота	2	Наступне лабораторне заняття
		Стили, шаблони, ресурси..	Самостійна робота	5	
	4	Служби візуалізації графіки WPF. Конфігурація панелей з використанням візуальних конструкторів Visual Studio.	Лекція	2	
		Конфігурація панелей.	Лабораторна робота	2	Наступне лабораторне заняття

	Вивчення вимог стандартних інтерфейсів.	Самостійна робота	5	
5	Фігури, геометрія, пензлі, трансформація компонування	Лекція	2	
	трансформація компонування.	Лабораторна робота	2	Наступне лабораторне заняття
	Фігури, геометрія, пензлі,	Самостійна робота	5	
6	Ресурси, анімація та стилі WPF	Лекція	2	
	Ресурси, анімація та стилі WPF	Контрольна робота	2	
	Ресурси, анімація та стилі WPF	Самостійна робота	5	
7	Технології ADO.NET доступу до даних. Архітектура ADO.NET. Огляд основних класів.	Лекція	2	
	Огляд основних класів.	Лабораторна робота	2	Наступне лабораторне заняття
	Архітектура ADO.NET.	Самостійна робота	6	
8	Команди. Тип SqlDataReader. Набори даних DataSet і таблиці DataTable.	Лекція	2	
	Набори даних DataSet і таблиці DataTable	Лабораторна робота	2	Наступне лабораторне заняття
	Команди. Тип SqlDataReader.	Самостійна робота	6	
9	Тип DataAdapter для роботи у від'єднаному режимі.	Лекція	2	
	роботи у від'єднаному режимі.	Лабораторна робота	2	Наступне лабораторне заняття
	Тип DataAdapter	Самостійна робота	6	
10	Сортування, пошук, фільтрація даних. Клас DataView	Лекція	2	
	Клас DataView	Лабораторна робота	2	Наступне лабораторне заняття
	Сортування, пошук, фільтрація даних	Самостійна робота	6	
11	ADO.NET Entity Framework – об'єктно-орієнтована технологія доступу до даних	Лекція	2	
	ADO.NET Entity Framework	Лабораторна робота	2	Наступне лабораторне заняття
	ADO.NET Entity Framework	Самостійна робота	6	
12	Робота з LINQ to SQL з використанням Visual Studio. Побудова операцій.	Лекція	2	
	Робота з LINQ to SQL.	Лабораторна робота	2	Наступне лабораторне заняття
	Побудова операцій.	Самостійна робота	6	
13	Звернення до бази даних SQL Server з використанням LINQ. Збережені процедури та LINQ to SQL	Лекція	2	
	Звернення до бази даних SQL Server з використанням LINQ	Контрольна робота	2	
	Збережені процедури та LINQ to SQL	Самостійна робота	5	
14	Вступ у ASP.NET Web Forms. Побудова однофайлової web-сторінки ASP.NET	Лекція	2	
	Побудова однофайлової web-сторінки ASP.NET	Лабораторна робота	2	Наступне лабораторне заняття
	ASP.NET	Самостійна робота	5	
15	Порівняння web-сайтів та web-додатків ASP.NET. Структура каталогів web-сайту ASP.NET	Лекція	2	
	Структура каталогів web-сайту ASP.NET	Лабораторна робота	2	Наступне лабораторне заняття
	Структура каталогів web-сайту ASP.NET	Самостійна робота	5	
16	Web-елементи керування ASP.NET. Базові класи Control та WebControl..	Лекція	2	
	Захист індивідуальних робіт	Лабораторна робота	2	

	Web-елементи керування ASP.NET.	Самостійна робота	5	
Підсумковий контроль, форма	Екзамен			
Пререквізити	Для вивчення курсу студенти потребують знань з попередніх семестрів навчання: основи ООП, використання винятків, взаємодія з потоками, використання контейнерів, модульне тестування.			
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 проектування алгоритмів і програм, виконання практичних завдань, у тому числі у команді з 3-4 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 програмування мовою C# (Microsoft Visual Studio). 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. 50 балів нараховують за виконання лабораторних завдань і контрольних робіт, ще 50 балів – за виконання екзаменаційного завдання. Лабораторні завдання можуть бути індивідуальні та командні. Упродовж семестру студент виконує не менше 10 лабораторних робіт, кожен з яких оцінюють 3-5 балів залежно від складності. Оцінка за екзаменаційне завдання може бути поділена на дві частини: 20 балів за засвоєння теоретичного матеріалу, виставлені після опитувань упродовж семестру (у формі тестувань, колоквиумів тощо) та 30 за написання комп'ютерної програми (декількох програм). Завдання до курсу можуть передбачати самостійне опрацювання актуальних публікацій щодо програмування для .Net.</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. Альтернативою відвідування лабораторних занять в університеті може бути дистанційна онлайн робота за розкладом проведення занять. Активність на лекціях і лабораторних урахують при оцінюванні відповідного лабораторного завдання.</p> <p>Академічна доброчесність: очікується, що роботи студентів будуть їхнім оригінальними дослідженнями 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 письмовій роботі студента є підставою для її незарахування викладачем, незалежно від масштабів плагіату чи обману. Жодні форми порушення академічної доброчесності не толеруються.</p>			
Запитання до екзамену	<ol style="list-style-type: none"> 1. Охарактеризуйте склад середовища .NET. 2. Засоби середовища .NET компіляції та виконання програм. 3. Система типів мови програмування C#. 4. Консольне введення мовою C#. Як вводити числові дані? 5. Консольне виведення мовою C#. Які засоби форматування ви знаєте? 6. Тип перелік мови C#. Приклад оголошення, використання, що 			

- пояснює можливості типу.
7. Тип перелік мови C#. Введення-виведення значень типу перелік.
 8. Типи одновимірний масив, багатовимірний масив мови C#. Наведіть приклади оголошення, ініціалізації, використання.
 9. Зубчастий масив мови C#. Поясніть структуру, можливості, наведіть приклад оголошення, ініціалізації, використання.
 10. Оператор перебору послідовності мови C#. Приклад. Умови, за яких його можна використовувати.
 11. Укажіть всі способи перебирання елементів одновимірного та двовимірного масивів у мові програмування C#.
 12. Копіювання, впорядкування масивів у мові програмування C#.
 13. Методи класу System.Object у мові C#.
 14. Що таке кортеж у мові C#? Як його оголошують, використовують?
 15. Охарактеризуйте тип структуру мови C#
 16. З якою метою застосовують блок ітератора? Наведіть приклад використання, поясніть алгоритм його функціонування.
 17. Охарактеризуйте члени-дані та члени-функції класу у мові C# (вичерпний перелік).
 18. Охарактеризуйте модифікатори доступу класу у мові C#.
 19. Охарактеризуйте модифікатори оголошення (не доступу) у мові C#.
 20. Охарактеризуйте конструктори класу у мові C#. Як з одного конструктора викликати інший?
 21. Що ви знаєте про параметри методів у мові C# (оголошення та використання)?
 22. Охарактеризуйте властивості (property) класу у мові C#.
 23. Що ви знаєте про автоматичні властивості у мові C#?
 24. Статичні класи у мові C#: призначення оголошення, використання.
 25. Розширення функціональності класу у мові C# (без доступу до самого класу).
 26. Синтаксис наслідування класів у мові C#. Конструктори підкласу.
 27. Особливості оголошення та використання абстрактного класу мовою C#.
 28. Як оголошують та перевизначають віртуальні методи мовою C#?
 29. Сумісність і приведення споріднених типів у мові C#. Перевірка типу.
 30. Охарактеризуйте поняття інтерфейсу у мові C#. Наведіть приклад наслідування інтерфейсу класом мови C#.
 31. Охарактеризуйте інтерфейси IEnumerable, IEnumerator мови C#.
 32. Охарактеризуйте інтерфейси ICloneable, IComparable мови C#.
 33. Типи, що допускають null мови C#. Операція поглинання нуля мови C#.
 34. Перевірка рівності об'єктів мовою C#. Опишіть усі варіанти.
 35. Як перевантажи операцію мовою C#? Наведіть приклад для одної операції.
 36. Методи індексатори, методи перетворення типу мовою C#.
 37. Охарактеризуйте тип делегата мови C#. Члени класу System.MulticastDelegate
 38. Використання групових делегатів, анонімних делегатів, лямбда-виразів мовою C#
 39. Охарактеризуйте поняття події мовою C#. Які кроки потрібно виконати, щоб оголосити подію у власному класі? Як її використати?
 40. Наведіть приклад взаємодії об'єктів власних класів через механізм подій (наприклад, подія зміни властивості).
 41. Які можливості є у мови C# для побудови контейнера елементів довільного типу?
 42. До чого можна застосувати узагальнення в мові C#? Перелічіть переваги узагальнень.

43. Використання обмежень для оголошення узагальнених класів мовою С#.
44. З якою метою застосовують винятки? Які класи винятків мови програмування С# визнаєте?
45. Опишіть засоби мови С# для перехоплення та опрацювання винятків.
46. Огляд контейнерів простору імен System.Collections мови С#.
47. Огляд контейнерів простору імен System.Collections.Specialized мови С#.
48. Огляд контейнерів простору імен System.Collections.Generic мови С#.
49. Індексована колекція змінного розміру мови С#.
50. Контейнери черга, стек мови С#.
51. Контейнери списки мови С#.
52. Контейнери словники мови С#.
53. Огляд класів простору імен System.IO мови С#.
54. Опишіть, для чого і як використовують Класи Directory, DirectoryInfo мови С#.
55. Опишіть, для чого і як використовують Класи File, FileInfo мови С#.
56. Опишіть, як мовою С# вивести інформацію до текстового файлу.
57. Опишіть, як мовою С# прочитати дані з текстового файлу.
58. Опишіть, як мовою С# використовують двійкові файли, для чого.
59. Опишіть один зі способів серіалізації об'єктів мовою С#.
60. Що ви знаєте про клас GarbageCollector мови С#?
61. Як оголошують та як використовують метод-фіналізатор мовою С#? У яких випадках потрібні фіналізатори?
62. Охарактеризуйте інтерфейс IDisposable мови С#. Які класи наслідують його?
63. Опишіть шаблон використання фіналізатора в класі, що реалізує IDisposable мови С#.
64. Наведіть приклад «лінивого» створення об'єктів мовою С#.
65. Що таке програмний потік ОС Windows? Для чого програмі декілька потоків? Поясніть загальні засади написання таких програм. Які проблеми багатопотокових програм ви знаєте? Як їх вирішують?
66. Які можливості класу Thread мови С# ви знаєте? Наведіть приклад використання.
67. З якою метою використовують клас С# BackgroundWorker? Опишіть його можливості та налаштування.
68. Які можливості класу Task мови С# ви знаєте? Наведіть приклад використання.
69. Як отримати доступ до типу довільного об'єкта С# програми? Поясніть на прикладах. Які властивості типу можна дослідити?
70. Як програмно отримати перелік методів, конструкторів довільного об'єкта С# програми? Наведіть приклад.
71. Як програмно отримати перелік полів, властивостей довільного об'єкта С# програми? Наведіть приклад.
72. Як програмно отримати доступ до С# збірки з метою дослідження? Які властивості збірки можна одразу ж перевірити?
73. Як програмно отримати колекцію типів, визначених у С# збірці?
74. Що таке атрибути програмного коду С#? Для чого їх використовують? Наведіть приклад
75. Як визначити власні атрибути програмного коду С#? Як їх використати?
76. Які види застосунків WPF ви знаєте? Охарактеризуйте стисло особливості кожного.
77. Охарактеризуйте призначення та можливості основних класів, які використовують для побудови WPF-застосунку.
78. Яку роль відіграє XAML для побудови WPF-застосунку? Назвіть

	<p>основні синтаксичні елементи XAML, наведіть приклади.</p> <p>79. Охарактеризуйте елементи керування WPF: кнопки, перемикачі, списки.</p> <p>80. Охарактеризуйте елементи керування WPF: поля введення, таблиці, написи.</p> <p>81. Охарактеризуйте елементи керування WPF: меню, інструментальні панелі.</p> <p>82. Охарактеризуйте діалогові вікна WPF.</p> <p>83. Охарактеризуйте LINQ to XML провайдер.</p> <p>84. Охарактеризуйте типи XElement, XContainer, XDocument.</p> <p>85. Як виконується побудова XML-дерева?</p>
Опитування	Анкету-оцінку з метою оцінювання якості курсу буде надано після завершення курсу.