



Основи програмування. Частина 2. Методології програмування

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>12 Інформаційні технології</i>
Спеціальність	<i>121 Інженерія програмного забезпечення</i>
Освітня програма	<i>Інженерія програмного забезпечення інформаційних систем</i>
Статус дисципліни	<i>Нормативна (цикл професійної підготовки)</i>
Форма навчання	<i>очна(заочна)</i>
Рік підготовки, семестр	<i>1 курс, весняний семестр</i>
Обсяг дисципліни	<i>6 кредитів 180 годин</i>
Семестровий контроль/ контрольні заходи	<i>Лабораторні, МКР, Екзамен</i>
Розклад занять	<i>http://rozklad.kpi.ua/Schedules/ScheduleGroupSelection.aspx</i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	<i>Лектор: к.т.н., Крамар Юлія Михайлівна, yulia.kramar@gmail.com, моб. +38(050)3420-99-10 Лабораторні: старший викладач Вітківська Ірина Іванівна, iryna.vitkovska-fiot@lll.kpi.ua , моб. +38(067)598-31-89</i>
Розміщення курсу	

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Навчальна дисципліна "Основи програмування" загалом і кредитний модуль "Основи програмування - 2. Методології програмування" зокрема, спрямовані на вивчення теоретичних та методологічних основ побудови програм на мовах програмування високого рівня з урахуванням сучасних концепцій і тенденцій розвитку, оволодіння інструментальними засобами створення таких програм, отримання практичних навичок розробки програмного забезпечення при вирішенні прикладних задач різного ступеня складності.

Відповідна теоретична та практична підготовка формує базові професійні навички з програмування і є основою для успішного опанування інших фахових дисциплін.

Метою дисципліни (кредитного модуля) є формування у студентів здатності розробляти програмне забезпечення для вирішення прикладних задач різного ступеня складності мовою програмування C# з використанням методології об'єктно-орієнтованого програмування.

Предмет навчальної дисципліни (кредитного модуля) – парадигма об'єктно-орієнтованого програмування, мова програмування високого рівня (C#), інструментальні засоби створення програмного забезпечення мовою C#.

Вивчення кредитного модуля «Основи програмування - 2. Методології програмування» сприяє формуванню у студентів наступних компетентностей. Загальні компетентності:

ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК2. Здатність застосовувати знання у практичних ситуаціях.

ЗК6. Здатність до пошуку, оброблення та узагальнення інформації з різних джерел. Фахові компетентності, формуванню яких сприяє даний кредитний модуль:

ФК1. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення.

ФК2. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.

ФК3. Здатність розробляти архітектури, модулі та компоненти програмних систем

ФК7. Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних.

ФК8. Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення.

ФК10. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя.

ФК11. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.

ФК13. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.

ФК14. Здатність до алгоритмічного та логічного мислення.

Формування зазначених компетентностей забезпечує досягнення наступних програмних результатів навчання:

ПРН1. Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.

ПРН3. Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення.

ПРН6. Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення.

ПРН7. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.

ПРН13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.

ПРН15. Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення.

ПРН18. Знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних.

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Для успішного засвоєння дисципліни необхідно мати базові знання з кредитного модуля «Основи програмування - 1. Базові конструкції», кредитного модуля «Алгоритми і структури даних - 1. Основи алгоритмізації». Успішне оволодіння знаннями з дисципліни готує студентів до вивчення наступних дисциплін, таких як:

- «Компоненти програмної інженерії» (ПО 7),
- «Системне програмне забезпечення» (ПО 13),
- «Програмування веб-застосунків» (ПО 14), –
- «Мультипарадигмне програмування» (ПО18),
- «Технології паралельних обчислень» (ПО 21).

3. Зміст навчальної дисципліни

Розділ 1. Процедурне та об'єктно-орієнтоване програмування.

Тема 1.1 Еволюція програмування. Процедурне та об'єктно-орієнтоване програмування. Мови програмування.

Тема 1.2 Основні терміни та парадигми об'єктно-орієнтованого програмування. Складові процесу розробки програмного забезпечення. Принципи ООП. Класи. Об'єкти. Екземпляри та статичні члени класу.

Розділ 2. Система типів C#. Перевантаження підпрограм.

Тема 2.1 Призначення методів в ООП. Способи передачі параметрів в C#. Повертання значень із методів.

Тема 2.2 Система типів в C#. Конструктори. Деструктори. Динамічний розподіл пам'яті в C#. Пам'ять стек та купа. Прибирання сміття в .Net.

Тема 2.3 Члени класу. Індикатори та властивості. Створення екземплярів об'єктів за допомогою властивостей. Правила використання властивостей

Тема 2.4 Перевантаження підпрограм (методів). Перевантаження унарних, бінарних, операцій, операторів приведення типів.

Розділ 3. Успадкування. Поліморфізм.

Тема 3.1 Успадкування, загальне положення. Методології декомпозиції. Одиночне успадкування в C#. Виклик конструктора базового класу. Порядок виконання конструкторів. Визначення членів, які успадковуються. Приведення типів в ієрархії успадкування.

Тема 3.2 Поліморфізм загальні положення. Раннє, пізнє зв'язування. Присвоєння та посилання на класи в ієрархії успадкування мовою C#. Приведення типів в ієрархії успадкування..

Тема 3.3 Віртуальні функції. Пізнє зв'язування. Реалізація поліморфізму мовою C#. Абстрактні класи та методи. Абстрактні методи в C#. Оператори is та as.

Тема 3.4 Інтерфейс. Порівняння абстрактних класів та інтерфейсів. Явна та неявна реалізація інтерфейсів

Тема 3.5 Відношення між класами та об'єктами. Порівняння та випадки застосування агрегації та композиції, агрегації та успадкування, успадкування та реалізації.

Розділ 4. Виключення.

Тема 4.1 Виключення. Оператори try, catch, finally. Виклик виключень. Оператор throw. Аргументи виключень. Конструктори класу Exception.

Тема 4.2 Вкладення блоків try. Повторна генерація виключень. Умови та правила генерації виключень. Стек виключення. Основні види виключень та їх призначення. Оператори checked unchecked. Власні виключення.

Розділ 5. Делегати. Обробка подій. Лямбда-вирази.

Тема 5.1 Об'явлення та використання делегатів. Обробка подій. Анонімні методи. Лямбда-вирази. Використання лямбда-виразів з делегатами.

Тема 5.2 Використання методів розширення та виразів запитів LINQ. Введення у запити LINQ. Мова запитів LINQ. LINQ to Objects. Операції LINQ.

Розділ 6. Структури даних. Лінійні та нелінійні структури даних.

Тема 6.1. Поняття структури даних. Лінійні та нелінійні структури даних. Прості спискові структури даних: стек, черга, дека. Складені спискові структури: двонаправлені списки, кільцеві списки, асоціативні списки.

Тема 6.2. Нелінійні структури даних. Деревя: бінарні, дерева бінарного пошуку. Алгоритми обробки дерев бінарного пошуку

Тема 6.43 Структури даних у бібліотеках .Net Collections та Collections.Generic. Шаблон об'єктно-орієнтованого програмування. Ітератор у колекціях.

4. Навчальні матеріали та ресурси

Базова література

1. Herbert Schildt «C# 4.0 The Complete Reference» 1st Edition — Osborne/McGraw Hill, 2010. — 1240 p.
2. Herbert Schildt «C++: The Complete Reference» 4st Edition — Osborne, 2002. — 1200 p.

3. Курс «.NET Fundamentals» [Електронний ресурс]. – Режим доступу: <https://learn.epam.com/start>

Додаткова література

1. N. Wirth. Algorithms and Data Structures. — Oberon, 2004. — 360 p.
2. Stephen Prata «C++ Primer Plus 6th Edition (Developer's Library) In 2 Volumes» — AddisonWesley Professional, 2011. — 1 440 p.
3. Ian Griffiths «Programming C# 5.0: Building Windows 8, Web, and Desktop Applications for the .NET 4.5 Framework», 2019. — 1236 p.
4. Головна сторінка середовища MS Visual Studio [Електронний ресурс]. – Режим доступу: <http://visualstudio.com>
5. Головна сторінка MSDN [Електронний ресурс]. – Режим доступу: <http://msdn.microsoft.com>

5. Навчальний контент

5.1. Лекційні заняття

Матеріали для вивчення дисципліни розміщені в електронному вигляді на сайті <https://learn.epam.com> та на гугл-диску, посилання надаються студентам на початку семестру. Контент доступний студенту із будь-якого місця в мережі Інтернет.

Лекції по дисципліні проводяться із використанням сучасних мультимедійних презентаційних технологій.

Таблиця 1

Лекційні заняття

Лекції	Назва теми лекції та перелік основних питань
	Розділ 1. Процедурне та об'єктно-орієнтоване програмування
1	Тема 1.1 Еволюція програмування. Процедурне та об'єктно-орієнтоване програмування. Мови програмування.
2	Тема 1.2 Основні терміни та парадигми об'єктно-орієнтованого програмування. Складові процесу розробки програмного забезпечення. Принципи ООП. Класи. Об'єкти. Екземплярні та статичні члени класу
	Розділ 2. Система типів C#. Перевантаження підпрограм(методів).
3	Тема 2.1 Призначення методів в ООП. Способи передачі параметрів в C#. Повертання значень із методів.
4	Тема 2.2 Система типів в C#. Конструктори. Деструктори. Динамічний розподіл пам'яті в C#. Пам'ять стек та купа. Прибирання сміття в .Net.

5	Тема 2.3 Члени класу. Індексатори. Властивості. Автоматичні властивості. Створення екземплярів об'єктів за допомогою властивостей.
6	Тема 2.4 Перевантаження підпрограми (методів). Перевантаження унарних та бінарних операцій.
	Розділ 3. Успадкування. Поліморфізм.
7	Тема 3.1 Успадкування загальне положення. Методології декомпозиції. Одиночне успадкування в C#. Виклик конструктора базового класу. Порядок виконання конструкторів. Визначення членів, які успадковуються. Приведення типів в ієрархії успадкування. Правила успадкування.
8	Тема 3.2 Поліморфізм загальні положення. Раннє, пізнє зв'язування. Присвоєння та посилання на класи в ієрархії успадкування мовою C#. Приведення типів в ієрархії успадкування.
9	Тема 3.3 Віртуальні функції. Пізнє зв'язування. Реалізація поліморфізму в C#. Абстрактні класи та методи. Абстрактні методи в C#. Оператори is та as.
10	Тема 3.4 Інтерфейс. Порівняння абстрактних класів та інтерфейсів. Явна та неявна реалізація інтерфейсів.
11	Тема 3.5 Відношення між класами та об'єктами. Порівняння та випадки застосування агрегації та композиції, агрегації та успадкування, успадкування та реалізації.
	Розділ 4. Виключення.
12	Тема 4.1 Виключення. Оператори try, catch, finally. Виклик виключень. Оператор throw. Аргументи виключень. Конструктори класу Exception.
13	Тема 4.2 Вкладення блоків try. Повторна генерація виключень. Умови та правила генерації виключень. Стек виключення. Основні види виключень та їх призначення. Оператори checked unchecked. Власні виключення.
	Розділ 5. Делегати. Обробка подій. Лямбда-вирази.
14	Тема 5.1 Об'явлення та використання делегатів. Обробка подій. Анонімні методи. Лямбда-вирази. Використання лямбда-виразів з делегатами.
15	Тема 5.2 Використання методів розширення та виразів запитів LINQ. Введення у запити LINQ. Мова запитів LINQ. LINQ to Objects. Операції LINQ.
	Розділ 6. Структури даних. Лінійні та нелінійні структури даних.
16	Тема 6.1. Поняття структури даних. Лінійні структури даних. Прості спискові структури даних: стек, черга, дека. Складені спискові структури: двонаправлені списки, кільцеві списки, асоціативні списки.
17	Тема 6.2. Нелінійні структури даних. Деревя: бінарні, дерева бінарного пошуку. Алгоритми обробки дерев бінарного пошуку

18	Тема 6.3. Структури даних у бібліотеках .Net Collections та Collections.Generic. Шаблон об'єктно-орієнтованого програмування. Ітератор у колекціях.
----	---

5.2. Лабораторні заняття

Лабораторні заняття виконуються з використанням мов програмування C# та версій середовищ розробки програм (IDE), які надаються розробниками для навчальних цілей безкоштовно. Вказівки до виконання лабораторних робіт включені в онлайн курс на платформі <https://learn.epam.com>. Для виконання кожної лабораторної роботи студенту надається не більше 2 тижнів.

За умови дистанційного навчання заняття, у тому числі контрольні заходи, проводяться з використанням сервісу Google, MS Teams та цифрової платформи Learn.

Вихідний код лабораторних робіт має бути розміщеним кожним студентом особисто у глобальному репозиторії <https://autocode.git.epam.com>.

Таблиця 2

Лабораторні заняття

№ з/п	Назва лабораторної роботи (комп'ютерного практикуму)	Ресурс	Кількість ауд. годин
1	Функції та процедури	Autocode	4
2	Опис та використання класів	Autocode	4
3	Успадкування	Autocode	4
4	Поліморфізм	Autocode	4
5	Інтерфейси. Реалізація інтерфейсів	Autocode	4
6	Обробка виключень	Autocode	4
7	Побудова та використання структур даних	Метод.вказ.	8
8	Делегати та події	Метод.вказ.	4

5.3. Практичні заняття

Практичні заняття слугують для закріплення теоретичних знань, набутих на лекціях, та для відпрацювань практичних навичок, які дозволять студентам виконати завдання лабораторних робіт та курсову роботу. Матеріалами для проведення практичних занять є лекційні матеріали та практичні завдання та приклади, подібні до завдань відповідних лабораторних робіт та курсової роботи.

Таблиця 3

№ з/п	Тема практичного заняття
1	Відпрацювання навичок створення функцій та процедур, передачі параметрів різними способами
2	Відпрацювання навичок опису та використання класів
3	Дослідження механізмів успадкування, порівняння одиночного та множинного успадкувань
4	Дослідження видів поліморфізму, відпрацювання навичок написання віртуальних функцій

5	Дослідження принципів побудови інтерфейсів, огляд бібліотеки інтерфейсів
6	Відпрацювання навичок написання коду з обробкою виключень
7	Відпрацювання навичок використання структур даних
8	Відпрацювання навичок роботи з делегатами, написання LINQ- Фільтрація, впорядкування та групування даних та об'єднання декількох наборів даних у LINQ-запиті.
9	Відпрацювання навичок роботи з текстовими даними. Дослідження стандартних бібліотечних засобів обробки тексту

5.4. Самостійна робота студента

До самостійної роботи студента відноситься виконання індивідуальних завдань з тематики, яка виноситься на лабораторні роботи, а також опрацювання теоретичного матеріалу за наданими текстами лекцій та додатковою літературою, у тому числі за темами, які винесені на самостійне вивчення (згідно таблиці 1). Усі навчальні матеріали (тексти лекцій, методичні вказівки до виконання лабораторних робіт) розміщені в електронному вигляді на сайті (<https://learn.epam.com/study/>), а також в середовищі «Електронний кампус». Навчальний контент доступний із будь-якого місця в мережі Інтернет.

На самостійну роботу студент має витратити кількість годин, що співмірна із кількістю годин, проведених ним на аудиторних заняттях.

Таблиця 4

Самостійна робота

№ з/п	Назва тем самостійної роботи
1	Тема 2.1 Опціональні та іменовані параметри, параметри-масиви та довільна кількість параметрів
2	Тема 2.2. Оптимізація роботи прибиральника сміття
3	Тема 2.3 Рекомендації використання властивостей та індексаторів
4	Тема 2.4 Перевантаження операторів приведення типів
5	Тема 3.1 Відкрите, захищене та закрите наслідування
6	Тема 3.2 Побічні ефекти множинного наслідування
7	Тема 3.3 Використання класу Object у бібліотечних класах
8	Тема 3.4 Перевантаження функцій
9	Тема 3.5 Перевизначення віртуальної функції в C#
10	Тема 3.6 Перевизначення інтерфейсів
11	Тема 3.7 Відношення між класами та об'єктами
12	Тема 3.8 Використання відношень у шаблонах проектування – базові відомості та приклади
13	Тема 4.1 Аналіз стеку виклику виключення. Вкладене виключення

14	Тема 4.2 Основні стандартні виключення та практика їх застосування
15	Тема 5.1 Використання подій в застосуваннях с графічним інтерфейсом.
16	Тема 5.2 Дерево виразів
17	Тема 5.3 Різновиди LINQ
18	Тема 5.4 Динамічний LINQ
20	Тема 6.1. Реалізація колізій хеш-таблиць
21	Тема 6.4. Сильно розгалужені дерева. Прошиті дерева
22	Тема 6.3. Додаткові типи колекцій: множина, відсортований словник
23	Тема 6.5. Побудова індексів в реляційних базах даних на основі дерев
24	Тема 6.6. Мова регулярних виразів

6. Політика навчальної дисципліни (освітнього компонента) Студент має вивчати дисципліну (кредитний модуль) протягом семестру, дотримуючись календарного плану виконання завдань комп'ютерного практикуму, вивчення тем лекційного матеріалу та виконання контрольних і модульних контрольних робіт. Усі завдання студент має виконувати **самостійно і вчасно**.

Для перевірки як теоретичних знань, так і практичних навичок в семестрі передбачена модульна контрольна робота (МКР), що складається з двох частин, які проводяться під час семестрових атестацій.

Оцінювання студентів здійснюється згідно рейтингової оцінки рівня підготовки студентів з дисципліни. Поточний стан успішності студенти можуть бачити в системі «Електронний кампус».

7. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

7.1. Поточний контроль

Поточний контроль успішності засвоєння знань виконується шляхом виконання ними: лабораторних робіт, МКР, завдань, що видані для самостійної проробки.

Семестровий рейтинг студента з дисципліни складається з балів, що він отримує за: – лабораторні роботи;

– одну модульну контрольну роботу (МКР поділяється на дві контрольні роботи тривалістю по одній академічній годині); – самостійну роботу; – іспиту.

7.2. Система рейтингових (вагових) балів та критерії оцінювання

Лабораторні роботи

Вагові бали кожної лабораторної роботи наведені у Таблиці 4. Сумарний ваговий бал за даний контрольний захід (*r_з*) складає **52 бали**.

Критерії оцінювання лабораторних робіт включають якість її виконання та захисту (Таблиця 5)

Таблиця 5

Вагові бали та критерії оцінювання лабораторних робіт.

	Назва роботи	Бали

№		Виконання	Захист	Сума
1	Функції та процедури	5		5
2	Опис та використання класів	5		5
3	Успадкування	5		5
4	Поліморфізм	5		5
5	Інтерфейси. Реалізація інтерфейсів	5		5
6	Обробка виключень	5		5
	Проміжний тестовий контроль по лабораторних роботах 1- 6	6		6
7	Побудова та використання структур даних	4	4	8
8	Делегати	4	4	8
	Разом за лабораторні роботи	44	8	52

Критерії оцінювання лабораторних робіт 1- 6 (оцінюються Autocode):

- Виконане завдання з автоматичною перевіркою на Autocode – 5 балів
- Після виконання лабораторних робіт 1-6 виконується тестове завдання для перевірки теоретичних знань з тем робіт 1-6 - 6 балів.

Критерії оцінювання лабораторних робіт 7- 8:

- “**відмінно**”, повна відповідь (не менше 95% потрібної інформації); - 8 балів;
- “**добре**”, достатньо повна відповідь (не менше 75% потрібної- 6 - 7 балів; інформації, або незначні неточності);
- “**задовільно**”, неповна відповідь (не менше 60% потрібної- 4 - 5 бали; інформації та деякі помилки);
- “**незадовільно**”, незадовільна відповідь. - < 4 балів

Модульна контрольна робота

Ваговий бал МКР –8 балів.

Критерії оцінювання кожної частини МКР:

- “**відмінно**”, повна відповідь (не менше 95% потрібної інформації); - 8 балів
- “**добре**”, достатньо повна відповідь (не менше 75% потрібної- 6 – 7 балів інформації, або незначні неточності);
- “**задовільно**”, неповна відповідь (не менше 60% потрібної- 4 - 5 балів інформації та деякі помилки);
- “**незадовільно**”, незадовільна відповідь. - < 4 балів

Умови позитивної проміжної атестації

Для отримання “зараховано” з першої проміжної атестації (8 тиждень) студент повинен мати не менше ніж 10 балів (за умови, якщо на початок 8 тижня згідно з календарним планом контрольних заходів “ідеальний” студент має отримати 15 балів).

Для отримання “зараховано” з другої проміжної атестації (14 тиждень) студент повинен мати не менше ніж 20 балів (за умови, якщо на початок 14 тижня згідно з календарним планом контрольних заходів “ідеальний” студент має отримати 30 балів).

Необхідною умовою допуску до екзамену є:

- Виконання перших 6 лабораторних робіт з автоматичною перевіркою та тестового завдання проміжного контролю;
- виконання МКР не нижче ніж на оцінку “задовільно”.

Критерії екзаменаційного оцінювання:

Згідно Додатку 1 (пункт 3.15) до наказу 7/86 від 8.05.2020 семестровий контроль з дисципліни “Основи програмування. Частина 2. Модульне програмування” передбачений у формі екзамену здійснюється за РСО-1 (“автоматом”) шляхом пропорційного перерахунку стартових балів у підсумкові бали за 100-бальною шкалою.

За умови, що здобувач вищої освіти виконав умови допуску до заходу семестрового контролю та набрав кількість балів не меншу за допустимий бал за РСО (R), переведення балів здійснюється за формулою (зокругленням результату до найближчого цілого).

У підсумку за екзамен студент отримує:

- “відмінно”	- 38-40 бали
- “дуже добре”	- 34-38 балів
- “добре”	30-33 балів
- “задовільно”	- 26-29 балів
- “достатньо”	24-25 балів
- “незадовільно”	- < 24 балів

Розрахунок шкали рейтингу R :

Максимальна сума вагових балів контрольних заходів протягом семестру складає: R_c
 $= 52+8 = 60$ балів

Екзаменаційна складова шкали дорівнює 40% , а саме:

$$R_{RE} = c = 40 \frac{0,4}{1 - 0,4}$$

балів.

Таким чином, рейтингова шкала з дисципліни складає **60+40 = 100 балів**.

Для отримання студентом відповідних оцінок - його рейтингова оцінка R_D переводиться в оцінку згідно з таблицею:

Бали $R_D=R_C+R_E$	Оцінка
95... 100	відмінно
85 ... 94	дуже добре
75 ... 84	добре
65 ... 74	задовільно
60 ... 64	достатньо
$R_D < 60$	незадовільно
$r^C < 50$ або не виконані інші умови допуску до екзамену	Не допущений

Робочу програму навчальної дисципліни (силабус):

Складено доцентом, к.т.н, доцентом, Крамар Юлією Михайлівною, старшим викладачем Вітківською Іриною Іванівною.

Ухвалено кафедрою ІІІ (протокол № 16 від 29.05.2024р)

Погоджено Методичною комісією факультету (протокол № 10 від 21.06.2024р.)