



ПРОГРАМУВАННЯ. ЧАСТИНА 1. ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ ТА ШАБЛОНИ ПРОЄКТУВАННЯ

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 мультимедійних та інформаційно-пошукових систем</i>
Статус дисципліни	<i>Нормативна</i>
Форма навчання	<i>Очна (денна)</i>
Рік підготовки, семестр	<i>2 рік навчання, 3 семестр</i>
Обсяг дисципліни	<i>Лекції: 54 год., комп'ютерний практикум: 18 год., самостійна робота: 78 год.</i>
Семестровий контроль/ контрольні заходи	<i>Іспит, календарний контроль</i>
Розклад занять	<i>Згідно розкладу на осінній семестр поточного навчального року (roz.kpi.ua)</i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	<i>Лектор: к.т.н., доцент, Заболотня Тетяна Миколаївна, tetiana.zabolotnia@gmail.com Комп'ютерний практикум: к.т.н., доцент, Заболотня Тетяна Миколаївна, асистент Юсин Яків Олексійович yusin.yakiv@gmail.com</i>
Розміщення курсу	<i>Google disk</i>

Програма навчальної дисципліни

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

Вивчення дисципліни «Програмування. Частина 1. Об'єктно-орієнтоване програмування та шаблони проєктування» дозволяє сформувати у здобувачів освіти компетенції, необхідні для розв'язання практичних задач професійної діяльності, пов'язаної із проєктуванням та розробленням об'єктно-орієнтованого програмного забезпечення.

***Метою** вивчення дисципліни «Програмування. Частина 1. Об'єктно-орієнтоване програмування та шаблони проєктування» є формування у здобувачів освіти здатності аналізувати вимоги до програмного продукту та формувати технічну постановку завдання на розроблення програми, обґрунтовано обирати та реалізовувати в коді необхідні принципи ООП, забезпечувати мінімізацію міжмодульних зв'язків у розробленому ПЗ та повторне використання коду.*

***Предметом** дисципліни «Програмування. Частина 1. Об'єктно-орієнтоване програмування та шаблони проєктування» є процес розроблення програмного забезпечення.*

*В контексті освітньої програми вивчення дисципліни «Програмування. Частина 1. Об'єктно-орієнтоване програмування та шаблони проєктування» формує у здобувачів освіти **фахові***

компетентностей (ФК), необхідних для розв'язання практичних задач професійної діяльності, пов'язаної з розробленням об'єктно-орієнтованого програмного забезпечення:

- **ФК02** Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.
- **ФК03** Здатність розробляти архітектури, модулі та компоненти програмних систем;
- **ФК08** Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення;
- **ФК13** Здатність обґрунтовано обирати та освоювати інструментарій з розроблення та супроводження програмного забезпечення;
- **ФК14** Здатність до алгоритмічного та логічного мислення;
- **ФК17** Здатність розробляти програмне забезпечення інформаційно-пошукових систем;
- **ФК19** Здатність розробляти програмне забезпечення мультимедійних та мультимедійних систем;

Програмні результати навчання (ПРН):

- **ПРН01** Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки;
- **ПРН03** Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення;
- **ПРН04** Знати і застосовувати професійні стандарти і інші нормативно-правові документи в галузі інженерії програмного забезпечення;
- **ПРН06** Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення;
- **ПРН07** Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення;
- **ПРН08** Знати та вміти розробляти людино-машинний інтерфейс.
- **ПРН09** Вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення.
- **ПРН12** Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення.
- **ПРН13** Знати і застосовувати методи розроблення 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;
- **ПРН23** Вміти документувати та презентувати результати розроблення програмного забезпечення;
- **ПРН38** Вміти застосовувати технології програмування для розроблення програмного забезпечення мультимедійних та інформаційно-пошукових систем.

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

Успішному вивченню дисципліни «Програмування. Частина 1. Об'єктно-орієнтоване програмування та шаблони проектування» передують вивчення дисциплін «Алгоритми та структури даних» та «Основи програмування» навчального плану підготовки бакалаврів за спеціальністю 121 Інженерія програмного забезпечення.

Отримані при засвоєнні дисципліни «Програмування. Частина 1. Об'єктно-орієнтоване програмування та шаблони проектування» теоретичні знання та практичні уміння є необхідними для вивчення дисципліни «Програмування. Частина 2. Основи веб-програмування та розроблення мобільних застосунків», «Програмування. Курсова робота», «Стандартизація та технології розроблення мультимедійних та інформаційно-пошукових програмних продуктів», «Програмне забезпечення мультимедійних систем», «Програмне забезпечення інформаційно-пошукових систем» навчального плану підготовки бакалаврів та успішному виконанню курсових та дипломних проєктів за спеціальністю 121 Інженерія програмного забезпечення.

Отримані при засвоєнні дисципліни «Програмування. Частина 1. Об'єктно-орієнтоване програмування та шаблони проектування» теоретичні знання та практичні уміння є необхідними для вивчення дисципліни «Методологія інженерії програмного забезпечення», «Інформаційно-пошукові системи та сервіси», «Технології штучного інтелекту для інформаційно-пошукових систем» навчального плану підготовки магістрів за спеціальністю 121 Інженерія програмного забезпечення.

3. Зміст навчальної дисципліни

Дисципліна «Програмування. Частина 1. Об'єктно-орієнтоване програмування та шаблони проектування» передбачає вивчення таких тем:

Тема 1. Вступ до ООП, основи .NET та C#

Тема 2. Способи реалізації принципів ООП у мові C#

Тема 3. Принципи створення «чистого» програмного коду

Тема 4. Структурні шаблони проектування

Тема 5. Твірні шаблони проектування

Тема 6. Поведінкові шаблони проектування

Тема 7. Сучасні техніки розроблення програмного забезпечення

Екзамен

4. Навчальні матеріали та ресурси

Базова література:

1. Бублик В.В. Об'єктно-орієнтоване програмування. – К.: ІТ-книга, 2015. – 624 с.:іл.

Використати для опанування тем щодо основ ООП. Матеріали знаходяться у вільному доступі в Інтернеті.

2. Шаблони проектування: методичні вказівки до виконання лабораторних робіт з дисципліни «Об'єктно-орієнтоване програмування» [Електронне видання] / Т.М.Заболотня. К.: НТУУ «КПІ», 2015. - 154с.

Використати для опанування практичних навиків щодо розділів про шаблони проектування. Матеріали знаходяться за посиланням: <https://ela.kpi.ua/handle/123456789/11657>.

3. Будаї А. Дизайн-патерни – просто, як двері [Електронне видання] / А.Будаї. URL: <http://designpatterns.andriybuday.com/>.

Ознайомитись з усією книгою, оскільки в ній стисло та зрозуміло викладений матеріал щодо шаблонів проектування. Матеріали знаходяться у вільному доступі в Інтернеті.

4. Head First. Патерни проектування / Ерік Фрімен, Елізабет Робсон, Кеті Сьерра і Берт Бейтс; пер. з англ. Г. Якубовська – Харків : ВД «Фабула», 2020. – 672 с.

Ознайомитись з усією книгою, оскільки в ній стисло та зрозуміло викладений матеріал щодо шаблонів проектування. Матеріали знаходяться у вільному доступі в Інтернеті.

5. Чистий код: створення і рефакторинг за допомогою Agile / Мартін Роберт; пер. з англ. І. Бондар-Терещенко. – Харків : ВД «Ранок» : Фабула, 2019. – 448 с.

Ознайомитись з розділами, що стосуються теми зі створення «чистого» програмного коду. Матеріали знаходяться у вільному доступі в Інтернеті.

6. Чому SOLID — важлива складова мислення програміста [Електронний ресурс] / І.Бранець URL: <https://dou.ua/lenta/articles/solid-principles/>

Ознайомитись зі статтею при вивченні теми про принципи SOLID. Матеріал знаходиться у вільному доступі в Інтернеті.

7. Рефакторинг.Гуру [Електронний ресурс] / О.Швець URL: <https://refactoring.guru/>

Ознайомитись з матеріалами сайту при вивченні шаблонів проектування, правил написання «чистого коду». Матеріали знаходяться у вільному доступі в Інтернеті.

8. Unit tests. Початок. Частина 1 [Електронний ресурс] / URL: <https://travelscode.com/unit-tests-rochatok-chastina-1/>, Unit tests. Частина 2 [Електронний ресурс] / URL: <https://travelscode.com/unit-tests-chastina-2/>

Ознайомитись з матеріалами сайту при вивченні unit-тестування. Матеріали знаходяться у вільному доступі в Інтернеті.

Допоміжна література:

9. B.Perkins, J.V.Hammer, J.D.Reid *Beginning C# 7 Programming with Visual Studio 2017* ISBN: 978-1-119-45868-5 John Wiley & Sons, Inc, 2018 – 912pp.

Ознайомитись з розділами, що стосуються тем щодо основ ООП та основ синтаксису мови програмування C#. Матеріали знаходяться у вільному доступі в Інтернеті.

10. M.Olsson *C# 7 Quick Syntax Reference: A Pocket Guide to the Language, APIs, and Library* ISBN-13 (pbk): 978-1-4842-3816-5, 2018 – 180pp.

Ознайомитись з розділами, що стосуються тем щодо основ синтаксису мови програмування C#. Матеріали знаходяться у вільному доступі в Інтернеті.

11. M.Fowler *Refactoring: Improving the Design of Existing Code (2nd Edition) (Addison-Wesley Signature Series (Fowler)) 2nd Edition* ISBN-13: 978-0134757599 – 2018. – 448 pp.

Ознайомитись з розділами, що стосуються теми зі створення «чистого» програмного коду. Матеріали знаходяться у вільному доступі в Інтернеті.

12. Bishop *C# 3.0 Design Patterns* ISBN 10: 0-596-52773-X, ISBN 13: 978-0-596-52773-0, O'Reilly, 2008. – 290pp.

Ознайомитись з прикладами програмного коду шаблонів проектування. Матеріали знаходяться у вільному доступі в Інтернеті.

13. J.Bender, J. McWherter *Professional Test-Driven Development with C#: Developing* ISBN: 978-0-470-64320-4 – 327pp.

Ознайомитись з розділами, що стосуються тем щодо сучасних технік розроблення програмного забезпечення. Матеріали знаходяться у вільному доступі в Інтернеті.

14. V.Vernon *Domain-Driven Design Distilled* ISBN-13: 978-0-13-443442-1, 2016. – 176pp.

Ознайомитись з усією книгою. Матеріали знаходяться у вільному доступі в Інтернеті

15. *Tour the Visual Studio IDE 15* [Електронний ресурс] / URL: <https://learn.microsoft.com/en-us/visualstudio/ide/quickstart-ide-orientation?view=vs-2022>

Ознайомитись з вмістом сайту для виконання практичних завдань у середовищі MS VS 2022.

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

№ з/п	Тип навчального заняття	Опис навчального заняття
<i>Тема 1. Вступ до ООП, основи .NET та C#</i>		
1	<i>Лекція 1. Сутність об'єктно-орієнтованого підходу до створення ПЗ</i>	<i>Історія виникнення ОО підходу, його переваги та недоліки. Основні принципи ООП. Типи структурних ієрархій "is-part-of", "is-a", "is-like-a". Основні відношення класів в ООП (UML-позначення). Завдання на СРС: п. 6, № 1.</i>
2	<i>Лекція 2. Платформа .Net та .Net Framework</i>	<i>Історія виникнення платформи .Net, її структура та функціональні можливості. Інфраструктура .Net Framework: призначення, структура, функціональні можливості, версії. Віртуальна машина CLR (common language runtime): структура, функціональні можливості. Особливості компіляції коду. CTS (common type system). CLS (common language specification). Завдання на СРС: п.6, №2</i>
3	<i>Лекція 3. Основи програмування мовою C# - Частина 1</i>	<i>Історія виникнення та розвитку мови C#. Основні синтаксичні конструкції мови C# та структура програми. Базові типи даних в C# (value/reference types) та перетворення типів. Завдання на СРС: п.6, №3</i>
4	<i>Лекція 4. Основи програмування мовою C# - Частина 2</i>	<i>Типи-перерахування, структури, масиви, nullable типи, методи та модифікатори параметрів, робота з файлами, серіалізація, виключні ситуації. Завдання на СРС: п.6, №4</i>
<i>Тема 2. Способи реалізації принципів ООП у мові C#</i>		
5	<i>Лекція 5. Клас як основний механізм абстрагування</i>	<i>Визначення класу, загальні методи. Порівняння класу та структури. Конструктори. Ключове слово this. Статичні дані. Завдання на СРС: п. 6, № 5</i>
6	<i>Лекція 6. Інкапсуляція</i>	<i>Модифікатори доступу. Властивості. Порівняння полів, властивостей та методів. Завдання на СРС: п. 6, № 6</i>
7	<i>Лекція 7. Наслідування та поліморфізм</i>	<i>Опис механізмів наслідування та включення-делегування. Приховування членів класу. Реалізація поліморфізму в кодї. Перевизначення методів (virtual/override). Завдання на СРС: п. 6, № 7</i>
8	<i>Лекція 8. Управління ресурсами пам'яті в .Net</i>	<i>Життєвий цикл об'єктів, покоління об'єктів, garbage collector та його методи, інтерфейс IDisposable, слабкі посилання. Завдання на СРС: п. 6, № 8</i>

9	Комп'ютерний практикум 1. Реалізація основних принципів ООП мовою С#	Завдання: ознайомитися з основами об'єктного підходу у мові С#, створенням класів, об'єктів, механізмами інкапсуляції, наслідування та поліморфізму; вивчити механізм управління ресурсами, реалізований у .Net. Завдання на СРС: п. 6, № 9
10	Лекція 9. Розширені можливості з наслідування	Абстрактні класи та інтерфейси, їх порівняння. Завдання на СРС: п. 6, № 10
11	Лекція 10. Механізм зворотного виклику	Делегати, події, лямбда-вирази, анонімні методи. Методи-розширення. Завдання на СРС: п. 6, № 11
12	Комп'ютерний практикум 2. Розширені можливості реалізації ООП у мові С#. Події	Завдання: ознайомитися з такими можливостями мови програмування С# як абстрактні класи, інтерфейси, делегати. Вивчити механізми оброблення подій у С#, а також можливості, які мають методи-розширення. Завдання на СРС: п. 6, № 12
Тема 3. Принципи створення «чистого» програмного коду		
13	Лекція 11. Правила створення «чистого» коду	«Чистий» код: правила написання, основні ознаки проблемних місць в коді, способи недопущення появи «нечистого» коду. Завдання на СРС: п. 6, № 13
14	Лекція 12. Основні принципи рефакторингу програмного коду	Основні підходи до рефакторингу програмного коду, перелік «запахів» неякісного коду, способи приведення коду до більш коректного стану. Завдання на СРС: п. 6, № 14
15	Комп'ютерний практикум 3. Управління ресурсами. «Чистий» код та рефакторинг	Завдання: Ознайомитися з правилами написання чистого коду та проведення рефакторингу, проаналізувати код комп'ютерного практикуму №№1,2 щодо дотримання правил «чистого коду», виконати рефакторинг. Завдання на СРС: п. 6, № 15
Тема 4. Структурні шаблони проектування		
16	Лекція 13. Вступ до шаблонів ООП, структурні шаблони, шаблон «Декоратор»	Передісторія виникнення шаблонів проектування, їх різновиди, переваги та недоліки, призначення, особливості реалізації. Узагальнений опис структурних шаблонів, їх функціональне призначення. Структура, функціональні характеристики, приклади застосування шаблону «Декоратор». Завдання на СРС: п. 6, № 16
17	Лекція 14. Шаблони «Компонувальник», «Міст», «Адаптер»	Структура, функціональні характеристики, приклади застосування шаблонів «Компонувальник», «Міст», «Адаптер»

		<i>Завдання на СРС: п. 6, № 17</i>
18	<i>Лекція 15. Шаблони «Легковаговик», «Фасад», «Заступник»</i>	<i>Структура, функціональні характеристики, приклади застосування шаблонів «Легковаговик», «Фасад», «Заступник» Завдання на СРС: п. 6, № 18</i>
19	<i>Комп'ютерний практикум 4 (частина 1). Реалізація структурних шаблонів проектування</i>	<i>Завдання: Створити програмне забезпечення (будь-якою мовою програмування) для вирішення поставлених задач за допомогою використання структурних шаблонів проектування Завдання на СРС: : п. 6, № 19</i>
<i>Тема 5. Твірні шаблони проектування</i>		
20	<i>Лекція 16. Шаблони «Прототип», «Фабричний метод», «Абстрактна фабрика»</i>	<i>Структура, функціональні характеристики, приклади застосування шаблонів «Прототип», «Фабричний метод», «Абстрактна фабрика» Завдання на СРС: п. 6, № 20</i>
21	<i>Лекція 17. Шаблони «Одинак», «Будівельник»</i>	<i>Структура, функціональні характеристики, приклади застосування шаблонів «Одинак», «Будівельник» Завдання на СРС: п. 6, № 21</i>
22	<i>Комп'ютерний практикум 4 (частина 2). Реалізація твірних шаблонів проектування</i>	<i>Завдання: Створити програмне забезпечення (будь-якою мовою програмування) для вирішення поставлених задач за допомогою використання твірних шаблонів проектування Завдання на СРС: п. 6, № 22</i>
<i>Тема 6. Поведінкові шаблони проектування</i>		
23	<i>Лекція 18. Шаблони «Стратегія», «Стан», «Шаблонний метод»</i>	<i>Структура, функціональні характеристики, приклади застосування шаблонів «Стратегія», «Стан», «Шаблонний метод». Завдання на СРС: п. 6, № 23</i>
24	<i>Лекція 19. Шаблони «Ланцюжок обов'язків», «Спостерігач», «Команда»</i>	<i>Структура, функціональні характеристики, приклади застосування шаблонів «Ланцюжок обов'язків», «Спостерігач», «Команда». Завдання на СРС: п. 6, № 24</i>
25	<i>Лекція 20. Шаблони «Посередник», «Відвідувач», «Ітератор»</i>	<i>Структура, функціональні характеристики, приклади застосування шаблонів «Посередник», «Відвідувач», «Ітератор». Завдання на СРС: п. 6, № 25</i>
26	<i>Комп'ютерний практикум 4 (частина 3). Реалізація поведінкових шаблонів проектування</i>	<i>Завдання: Створити програмне забезпечення (будь-якою мовою програмування) для вирішення поставлених задач за допомогою використання поведінкових шаблонів проектування Завдання на СРС: п. 6, № 26</i>
27	<i>Лекція 21. Шаблони рівня програми</i>	<i>Структура, функціональні характеристики, приклади застосування шаблонів MVC, MVVM Завдання на СРС: п. 6, № 27</i>

<i>Тема 7. Сучасні техніки розроблення програмного забезпечення</i>		
28	<i>Лекція 22. Реалізація принципів SOLID при проектуванні програмного забезпечення</i>	<i>Принципи SOLID розроблення об'єктно-орієнтованого програмного забезпечення Завдання на СРС: п. 6, № 28.</i>
29	<i>Лекція 23. Застосування DDD при проектуванні програмного забезпечення</i>	<i>Методи стратегічного та тактичного проектування моделей ПЗ Завдання на СРС: п. 6, № 29.</i>
30	<i>Комп'ютерний практикум 5 (частина 1). Застосування DDD для розроблення ПЗ</i>	<i>Завдання: За допомогою методів DDD розробити модель програмного забезпечення для вирішення задач, сформульованих в комп. практикумі №4. Завдання на СРС: п. 6, № 30</i>
31	<i>Лекція 24. Основи unit-тестування</i>	<i>Визначення unit-тестів. Mock-об'єкти. Фази unit-тестування. Завдання на СРС: п. 6, № 31</i>
32	<i>Лекція 25. Практичні аспекти впровадження TDD</i>	<i>Шаблон DI у unit-тестуванні, створення навчального застосунку з використанням TDD. Завдання на СРС: п. 6, № 32</i>
33	<i>Лекція 26. Сценарії тестування</i>	<i>Інтеграційне тестування. Тестування веб-застосунку. Тестування WPF-застосунку. Завдання на СРС: п. 6, № 33</i>
34	<i>Комп'ютерний практикум 5. Застосування TDD та DDD для розроблення ПЗ</i>	<i>Завдання: За допомогою прийомів TDD розробити програмне забезпечення для вирішення задач, які сформульовані в комп'ютерному практикумі №4. Завдання на СРС: п. 6, № 34</i>
35	<i>Лекція 27. Вивчення додаткових функціональних можливостей середовища розроблення MS Visual Studio 2022.</i>	<i>Завдання: Ознайомитись з набором функцій, які надає MS VS 2022 розробнику для ефективної роботи над програмним кодом. Завдання на СРС: п. 6, № 35</i>
<i>Іспит</i>		

6. Самостійна робота студентів

Дисципліна «Програмування. Частина 1. Об'єктно-орієнтоване програмування та шаблони проектування» ґрунтується на самостійних підготовках до аудиторних занять на теоретичні та практичні теми.

<i>№ з/п</i>	<i>Назва теми, що виноситься на самостійне опрацювання</i>	<i>Кількість годин</i>	<i>Література</i>
1	<i>Підготовка до лекції 1</i>	1	1, 9, 10
2	<i>Підготовка до лекції 2</i>	1	1, 9, 10
3	<i>Підготовка до лекції 3</i>	1	1, 9, 10
4	<i>Підготовка до лекції 4</i>	1	1, 9, 10
5	<i>Підготовка до лекції 5</i>	1	1, 9, 10
6	<i>Підготовка до лекції 6</i>	1	1, 9, 10

7	Підготовка до лекції 7	1	1, 9, 10
8	Підготовка до лекції 8	1	1, 9, 10
9	Підготовка до комп'ютерного практикуму №1	2	Всі джерела до лекцій №3-8
10	Підготовка до лекції 9	1	1, 9, 10
11	Підготовка до лекції 10	1	1, 9, 10
12	Підготовка до комп'ютерного практикуму №2	2	Всі джерела до лекцій №9 - 10
13	Підготовка до лекції 11	1	5, 7
14	Підготовка до лекції 12	1	5, 7, 11
15	Підготовка до комп'ютерного практикуму №3	3	Всі джерела до лекцій №11 - 12
16	Підготовка до лекції 13	1	2, 3, 4, 7, 12
17	Підготовка до лекції 14	1	2, 3, 4, 7, 12
18	Підготовка до лекції 15	1	2, 3, 4, 7, 12
19	Підготовка до комп'ютерного практикуму №4 (частина 1)	3	Всі джерела до лекцій №13-15
20	Підготовка до лекції 16	1	2, 3, 4, 7, 12
21	Підготовка до лекції 17	1	2, 3, 4, 7, 12
22	Підготовка до комп'ютерного практикуму №4 (частина 2)	3	Всі джерела до лекцій №16-17
23	Підготовка до лекції 18	1	2, 3, 4, 7, 12
24	Підготовка до лекції 19	1	2, 3, 4, 7, 12
25	Підготовка до лекції 20	1	2, 3, 4, 7, 12
26	Підготовка до комп'ютерного практикуму №4 (частина 3)	3	Всі джерела до лекцій №18-20
27	Підготовка до лекції 21	1	2, 3, 4, 7, 12
28	Підготовка до лекції 22	1	6, 7
29	Підготовка до лекції 23	1	14
30	Підготовка до комп'ютерного практикуму №5 (частина 1)	2	14
31	Підготовка до лекції 24	1	8, 13
32	Підготовка до лекції 25	1	8, 13
33	Підготовка до лекції 26	1	13
34	Підготовка до комп'ютерного практикуму №5 (частина 2)	3	Всі джерела до лекцій №24-26
35	Підготовка до лекції 27	1	15
36	Підготовка до складання іспиту	30	Весь матеріал семестру

7. Політика навчальної дисципліни (освітнього компонента)

- Відвідування лекційних занять є обов'язковим.
- Відвідування занять комп'ютерного практикуму може бути епізодичним та за потреби захисту робіт комп'ютерного практикуму.
- Правила поведінки на заняттях: активність, повага до присутніх, відключення телефонів.
- Дотримання політики академічної доброчесності.
- Правила захисту робіт комп'ютерного практикуму: роботи повинні бути зроблені згідно варіанту здобувача освіти, що визначається його номером у списку групи.
- Правила призначення заохочувальних та штрафних балів є такими:
Заохочувальні бали нараховуються за:
- виконання додаткових творчих завдань (максимальна кількість балів за всі завдання – 5 балів).

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

Протягом семестру студенти виконують 5 комп'ютерних практикумів. Максимальна кількість балів за комп'ютерні практикуми – 10 балів за кожен практикум.

Бали нараховуються за:

- відповідь під час захисту роботи: 3 бали;
- якість виконання програмного коду: 4 бали;
- самостійність виконання завдання, нестандартне рішення задачі : 2 бали
- якість оформлення протоколу: 1 бал.

Критерії оцінювання відповіді:

- 3 бали – відповідь повна, добре аргументована;
- 2 бали – в цілому відповідь вірна, але має недоліки або незначні помилки;
- 1 бал – у відповіді є суттєві помилки;
- 0 балів – немає відповіді або відповідь невірна.

Критерії оцінювання якості виконання програмного коду:

- 4 бали – робота виконана якісно, в повному обсязі;
- 3 бали – робота виконана якісно, в повному обсязі, але має недоліки;
- 2 бали – робота виконана в повному обсязі, але містить незначні помилки;
- 1 бал – робота виконана не в повному обсязі та містить несуттєві помилки.
- 0 балів – робота виконана не в повному обсязі та містить суттєві помилки.

Критерії оцінювання самостійного виконання завдання, нестандартного рішення задачі:

- 2 бали – подана робота відрізняється оригінальним підходом до вирішення поставленої задачі або в роботі використані самостійно опановані синтаксичні конструкції мови C#;
- 1 бал – у роботі самостійно реалізовано стандартний підхід до вирішення поставленої задачі;
- 0 балів – в роботі запозичено стандартне рішення задачі.

Критерії оцінювання якості оформлення протоколу:

- 1 бал – протокол виконано якісно, в повному обсязі;
- 0 балів – оформлення протоколу має недоліки.

Максимальна кількість балів за виконання та захист комп'ютерних практикумів:

10 балів × 5 = 50 балів.

Таким чином, семестрова складова рейтингової шкали $R_C = 50$ балів, визначається як сума балів, отриманих за виконання та захист робіт комп'ютерного практикуму.

$R_C = 50$ балів.

Склад та критерії оцінювання відповіді на іспиті:

Екзаменаційний білет складається з 3 питань – 2 теоретичних та 1 практичного.

Екзаменаційна складова рейтингової шкали $R_E = 50$ балів.

Ці бали можна отримати таким чином: за перше теоретичне завдання – від 0 до 20 балів, за друге – від 0 до 10 балів, за практичне завдання – від 0 до 20 балів.

Критерії оцінювання першого теоретичного запитання екзаменаційної роботи:

20 балів – вірна та змістовна відповідь;

15-19 балів – відповідь змістовна, але має незначні недоліки;

12-15 – відповідь неповна;

0-11 балів – немає відповіді, відповідь невірна, відповідь має суттєві недоліки.

Критерії оцінювання практичного запитання екзаменаційної роботи:

18-20 балів – вірна та змістовна відповідь;

10-17 балів – відповідь змістовна, але має незначні недоліки;

5-9 балів – відповідь неповна або містить незначні помилки;

0-4 бали – немає відповіді або відповідь невірна.

Максимальна кількість балів за відповідь на екзамені:

$R_E = 20 + 10 + 20 = 50$ балів.

Рейтингова шкала з дисципліни дорівнює: $R = R_C + R_E = 50$ балів + 50 балів = 100 балів.

Календарний контроль: провадиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу.

На першій атестації (8-й тиждень) студент отримує «зараховано», якщо його поточний рейтинг не менше 5 балів (50 % від максимальної кількості балів, яку може отримати студент до першої атестації).

На другій атестації (14-й тиждень) студент отримує «зараховано», якщо його поточний рейтинг не менше 20 балів (50 % від максимальної кількості балів, яку може отримати студент до другої атестації).

Семестровий контроль: екзамен

Умови допуску до семестрового контролю: виконання і захист всіх робіт комп'ютерного практикуму та семестровий рейтинг студента (r_C) не менше 60 % від R_C , тобто не менше 30 балів. В іншому разі студент повинен виконати додаткову роботу та підвищити свій рейтинг.

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

Кількість балів	Оцінка
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

Робочу програму навчальної дисципліни (силабус):

Складено к.т.н., доц., Заболотня Т.М.

Ухвалено кафедрою ПЗКС (протокол № 12 від 26.04.2023 р.)

Погоджено Методичною комісією факультету прикладної математики (протокол № 10 від 26.05.2023 р.)