

ДЕРЖАВНИЙ ТОРГОВЕЛЬНО-ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
СИСТЕМА УПРАВЛІННЯ ЯКІСТЮ
Система забезпечення якості освітньої діяльності та якості вищої освіти
сертифікована на відповідність ДСТУ ISO9001:2015 / ISO 9001:2015

Кафедра цифрової економіки та системного аналізу

ЗАТВЕРДЖЕНО

вченою радою ДТЕУ

(пост. п. 5 від «15 травня» 2022 р.)

Ректор

Анатолій МАЗАРАКІ



КРОСПЛАТФОРМЕНЕ ПРОГРАМУВАННЯ /
CROSS-PLATFORM PROGRAMMING

РОБОЧА ПРОГРАМА /
COURSE OUTLINE

освітній ступінь	бакалавр / bachelor
галузь знань	12 Інформаційні технології / Information Technologies
спеціальність	124 Системний аналіз / System Analysis
освітня програма	Інформаційні технології та бізнес-аналітика (Data Science) / Information Technologies and Business Analytics (Data Science)

Київ 2022

Розповсюдження і тиражування без офіційного дозволу ДТЕУ заборонено

Автор: С. В. Міценко, доцент кафедри цифрової економіки та системного аналізу, к.т.н., доцент

А.М.Тарасюк, асистент кафедри цифрової економіки та системного аналізу

Розглянуто і схвалено на засіданні кафедри цифрової економіки та системного аналізу та 24 жовтня 2022 р., протокол №3.

Рецензенти: В.В.Кулаженко, доцент кафедри цифрової економіки та системного аналізу, к.е.н., доцент

М. Г. Шарафутдінов, директор з розвитку компанії «CenterResearch&Development», бізнес-аналітик

**КРОС-ПЛАТФОРМНЕ ПРОГРАМУВАННЯ /
CROSS-PLATFORM PROGRAMMING**

**РОБОЧА ПРОГРАМА /
COURSE OUTLINE**

1. СТРУКТУРА ДИСЦИПЛІНИ ТА РОЗПОДІЛ ГОДИН ЗА ТЕМАМИ (ТЕМАТИЧНИЙ ПЛАН)

Для вивчення дисципліни відводиться 180 год. / 6 кредитів ЄКТС (лекцій – 14 год., практичних занять – 42 год., самостійної роботи – 124 год.), підсумковий контроль – письмовий екзамен.

Назва теми	Кількість годин				Форми контролю*
	Усього годин/ кредитів	із них:			
		лекції	лабора- торні заняття	самостій- на робота студентів	
<i>III семестр</i>					
Тема 1. Принципи крос-платформного підходу	12	1	0	10	ВПЗ, ВІЗ
Тема 2. Підходи до розробки крос-платформного програмного забезпечення	14	1	2	14	ВПЗ, ВІЗ
Тема 3. Типи даних у мові С# Застосування алгоритмів у крос-платформному програмному забезпеченні. Поняття Big O	22	1	4	12	ВПЗ, ВІЗ
Тема 4. Типи даних DateTime. Робота з датою та часом у С#. Типи даних String. Робота з текстовими даними	16	1	4	12	ВПЗ, ВІЗ
Тема 5. Класи та інтерфейси у крос-платформному програмному забезпеченні. Призначення та специфіка	16	1	4	12	ВПЗ, ВІЗ
Тема 6. Структура крос-платформного проекту	18	1	2	12	ВПЗ, ВІЗ
Тема 7. Робота з даними мовою С#. Структури даних мови С#. Робота з базами даних.	20	1	4	12	ВПЗ, ВІЗ
Тема 8. Робота з базами даних у С#. Специфіка роботи з NoSQL базами даних	20	1	4	12	ВПЗ, ВІЗ
Тема 9. Додаткові можливості об'єктно орієнтованого	22	1	2	14	ВПЗ, ВІЗ

Назва теми	Кількість годин				Форми контролю*
	Усього годин/кредитів	із них:			
		лекції	лабораторні заняття	самостійна робота студентів	
<i>III семестр</i>					
програмування					
Тема 10. LINQ та його застосування. F# та його можливості у C#	20	1	4	14	ВПЗ, ВІЗ
Тема 11. Коректність і безпека крос-платформного програмного забезпечення	20	1	2	14	ВПЗ, ВІЗ
Тема 12. Патерни у крос-платформному програмному забезпеченні	20	1	4	14	ВПЗ, ВІЗ
Тема 13. Логіювання та документування коду. Code convention.	20	1	2	14	ВПЗ, ВІЗ
Тема 14. Тестування крос-платформного програмного забезпечення	20	1	4	14	ВПЗ, ВІЗ
<i>Підсумковий контроль – письмовий екзамен</i>					
РАЗОМ	180/6	14	42	124	x

Примітка*:

ВПЗ – виконання практичних завдань; ВІЗ – виконання індивідуальних завдань.

2. ТЕМАТИКА ТА ЗМІСТ ЛЕКЦІЙНИХ, ЛАБОРАТОРНИХ ЗАНЯТЬ І САМОСТІЙНОЇ РОБОТИ СТУДЕНТІВ

Результати навчання	Навчальна діяльність	Робочий час студента, год.
<p>Знати: основні терміни програмування</p> <p>Вміти: виявляти особливості синтаксису мов програмування</p>	<p>Тема 1. Принципи крос-платформного підходу План лекції 1</p> <ol style="list-style-type: none"> 1. Підходи до розробки програмних продуктів. Крос-платформне програмування. 2. Порівняння нативної та крос-платформної розробки. Переваги та недоліки обох підходів. 3. Розробка програмного забезпечення на мові програмування C# на базі платформ .Net Core та платформи мобільної розробки Xamarin. Переваги застосування мови C# для крос-платформної розробки. 4. Розробка програмного забезпечення на основі мови програмування Python. Фреймворк Kivy для ефективної розробки крос-платформного програмного забезпечення. <p>Список рекомендованих джерел: Основний: 1[с.75-82],2[с.85-97],4[с.72-90]. Додатковий: 6[с.58-90],8[с.48-60],10[с.75-89]. Інтернет-ресурси: 4,5</p> <p>Самостійна робота студентів. Вивчення основних та додаткових джерел, Інтернет-ресурсів, підготовка до тестування, підготовка презентації або наукового повідомлення 1[с.12-32],2[с.8-20],4[с.21-32],6[с.20-44],8[с.28-40],9[с.14-32].</p>	<p>1</p>

Результати навчання	Навчальна діяльність	Робочий час студента, год.
<p>Знати: Основни програмування</p> <p>Вміти: писати код на с подібних мовах</p>	<p>Тема 2. Підходи до розробки крос-платформного програмного забезпечення План лекції 2</p> <ol style="list-style-type: none"> Паридигми розробки програмного забезпечення. Об'єктно-орієнтована парадигма розробки Основні принципи об'єктно-орієнтованої парадигми. Поняття інкапсуляції. Підходи до визначення інкапсуляції. Інкапсуляція та приховування. Поняття спадкування. Поняття Upcastingта Downcasting. Поняття поліморфізму. <p>Список рекомендованих джерел: Основний:1[с.75-82],2[с.85-97],4[с.72-90]. Додатковий:6[с.58-90],8[с.48-60],10[с.75-89]. Інтернет-ресурси:4,5</p> <p>Самостійна робота студентів. Підготовка до лабораторного заняття за матеріалами теми, вивчення основних та додаткових джерел, Інтернет-ресурсів, підготовка до тестування, виконання індивідуальних завдань, підготовка презентації або наукового повідомлення 1[с.12-32],2[с.8-20],4[с.21-32],6[с.20-44],8[с.28-40],9[с.14-32].</p> <p style="text-align: center;">Лабораторні заняття 1</p> <p>Онови розробки крос-платформних додатків на мові C# Мета: ознайомлення із можливостями мови програмування C#, основними типами даних математичними та логічними операціями</p>	<p>1</p> <p>2</p>

Результати навчання	Навчальна діяльність	Робочий час студента, год.
	<p>1. Практичне завдання:</p> <ul style="list-style-type: none"> - За наданими викладачем матеріалом для практичного заняття ознайомитись: - Із вводом та виводом інформації - Основними математичними операціями - Логічним операціям <p>Циклам та роботі з ними умовними конструкціями</p>	4
<p>Знати: Структури даних та алгоритми сортування</p> <p>Вміти: Проектувати структури, аналізувати роботу алгоритмів</p>	<p>Тема 3. Типи даних у мові С# Застосування алгоритмів у крос-платформному програмному забезпеченні. Поняття Big O</p> <p>План лекції 3</p> <ol style="list-style-type: none"> 1. Типи даних у мові С#. Базові та посилальні типи даних, їх основні відмінності. 2. Поняття умовних конструкцій. Види умовних конструкцій. 3. Поняття циклу. Види циклів. 4. Поняття алгоритму. Класифікація алгоритмів. 5. Складність алгоритмів та концепція Big O. <p>Список рекомендованих джерел: Основний: 1[с.75-82],2[с.85-97],4[с.72-90]. Додатковий:6[с.58-90],8[с.48-60],10[с.75-89]. Інтернет-ресурси:4,5</p> <p>Самостійна робота студентів. Підготовка до лабораторного заняття за матеріалами теми, вивчення основних та додаткових джерел, Інтернет-ресурсів, підготовка до тестування, виконання індивідуальних завдань 1[с.12-32],2[с.8-20],4[с.21-32],6[с.20-44],8[с.28-40],9[с.14-32].</p>	1

Результати навчання	Навчальна діяльність	Робочий час студента, год.
	<p align="center">Лабораторні заняття 2 Типи даних у мові C#</p> <p>Мета: ознайомлення із можливостями мови програмування C#, основними типами даних математичними та логічними операціями</p> <p>1. Практичне завдання:</p> <ul style="list-style-type: none"> - За наданим викладачем завданнями реалізувати - Усі змінні базових типів - Реалізувати перетворення типу змінних - Запропонувати тип змінних для роботи: - З віком - Сеансом роботи - Номерами телефонів - Користувацькими запитами 	4
<p>Знати: UML діаграми</p> <p>Вміти: Проектувати UML діаграм</p>	<p>Тема 4. Типи даних Date Time. Робота з датою та часом у C#. Типи даних String. Робота з текстовими даними</p> <p align="center">План лекції 4</p> <ol style="list-style-type: none"> 1. Тип даних DateTime. Форми часу. 2. Форматування дати та часу. Сортування дати і часу. Особливі випадки сортування дати та часу 3. Тип даних String. Операція з рядками. Форматування та інтерполяція строк. 4. Клас StringBuilder 5. Регулярні вирази. Правила застосування регулярних виразів та вимоги до них. <p>Список рекомендованих джерел: Основний: 1[с.75-82], 2[с.85-97], 4[с.72-90].</p>	1

Результати навчання	Навчальна діяльність	Робочий час студента, год.
	<p>Додатковий:6[с.58-90],8[с.48-60],10[с.75-89]. Інтернет-ресурси:4,5 Самостійна робота студентів. Вивчення основних та додаткових джерел, Інтернет-ресурсів, підготовка до тестування, підготовка презентації або наукового повідомлення 1[с.12-32],2[с.8-20],4[с.21-32],6[с.20-44],8[с.28-40],9[с.14-32].</p> <p style="text-align: center;">Лабораторні заняття 3 Дата та час у мові С#</p> <p>Мета: ознайомлення із можливостями роботи з часом у С# 1. Практичне завдання:</p> <ul style="list-style-type: none"> - За наданими викладачем матеріалом для практичного заняття ознайомитись: - Із форматами дати - Основними операціями з датами - Логічним операціям з датами - Фільтрація та сортування 	<p style="text-align: center;">12</p> <p style="text-align: center;">4</p>
<p style="text-align: center;">Знати: базові принципи розробки додатків на статично типізованих мовах</p> <p style="text-align: center;">Вміти: створювати</p>	<p style="text-align: center;">Тема 5. Класи та інтерфейси у крос-платформному програмному забезпеченні. Призначення та специфіка План лекції 5</p> <ol style="list-style-type: none"> 1. Поняття класу у програмування. Види класів. 2. Абстрактні класи та їх можливості. 3. Приватний та статичний конструктор. 4. Поняття інтерфейсу у програмування. Можливості застосування інтерфейсів. 5. Інтерфейси і абстрактні класи. 6. Поняття Узагальнень (Generics), та їх застосування. 	<p style="text-align: center;">1</p>

Результати навчання	Навчальна діяльність	Робочий час студента, год.
<p>додатки на с подібних мовах</p>	<p>Список рекомендованих джерел: Основний: 1[с.75-82],2[с.85-97],4[с.72-90]. Додатковий: 6[с.58-90],8[с.48-60],10[с.75-89]. Інтернет-ресурси: 4,5</p> <p>Самостійна робота студентів. Підготовка до лабораторного заняття за матеріалами теми, вивчення основних та додаткових джерел, Інтернет-ресурсів, підготовка до тестування, виконання індивідуальних завдань 1[с.12-32],2[с.8-20],4[с.21-32],6[с.20-44],8[с.28-40],9[с.14-32].</p> <p>Лабораторні заняття 4 Інтерфейси у мові С#</p> <p>Мета: розробити інтерфейси та абстрактні класи для додатку із лабораторної роботи №4</p> <p>1. Практичне завдання:</p> <ul style="list-style-type: none"> - На основі додатку із лабораторної №4 спроектувати та розробити - Інтерфейси у системі – визначити їх основні напрями застосування та потребу у їх підключенні - Абстрактні класи та потребу їх реалізації - Пояснити різницю між інтерфейсами та абстрактними класами 	<p>14</p> <p>4</p>
<p>Знати: Основні підходи до роботи з відео, аудіо та фото</p>	<p>Тема 6. Структура крос-платформного проекту План лекції 6</p> <ol style="list-style-type: none"> 1. Найпоширеніші патерни розробки додатків. Підходи до поділу додатку на рівні. Принципи SOLID, KISS, DRY. 2. Патерн MVC та його особливості. Роль моделі у додатку. Робота з контролерами. Проектування візуального дизайну 	<p>1</p>

Результати навчання	Навчальна діяльність	Робочий час студента, год.
<p>матеріалами</p> <p>Вміти: Застосовувати додаткові матеріали та зовнішні ресурси у розробці</p>	<p>3. Патерн MVVM та його особливості. Розробка моделей для системи. Проектування візуального представлення. Розробка рівня View-Model.</p> <p>4. Мова UML. UML діаграми. Види UML діаграм. Діаграма класів. Діаграма активностей.</p> <p>Список рекомендованих джерел: Основний: 1[с.75-82],2[с.85-97],4[с.72-90]. Додатковий: 6[с.58-90],8[с.48-60],10[с.75-89]. Інтернет-ресурси: 4,5</p> <p>Самостійна робота студентів. Підготовка до лабораторного заняття за матеріалами теми, вивчення основних та додаткових джерел, Інтернет-ресурсів, підготовка до тестування, виконання індивідуальних завдань 1[с.12-32],2[с.8-20],4[с.21-32],6[с.20-44],8[с.28-40],9[с.14-32].</p> <p>Лабораторні заняття 5 Візуальне представлення додатку у мові C# Мета: розробити візуальне представлення для додатку із лабораторної роботи №4</p> <p>1. Практичне завдання:</p> <ul style="list-style-type: none"> - Ознайомитись із можливостями візуальних редакторів інтерфейсу WinForms, WPF UWP - На основі завдання із лабораторної №4 розробити - Візуальне представлення сторінок входу та реєстрації у додатку - Персонального кабінету користувача - Головної сторінки - Сторінки вибору 	<p>14</p> <p>2</p>

Результати навчання	Навчальна діяльність	Робочий час студента, год.
<p>Знати: базові принципи взаємодії компонентів додатку</p> <p>Вміти: реалізувати взаємодію компонентів за різних функціональних етапів взаємодії</p> <p>Знати: Об'єктно-орієнтоване програмування, його основні парадигми</p>	<p><i>Додавання та зміни параметрів сутностей</i></p> <p>Тема 7. Робота з даними мовою С#. План лекції 7</p> <ol style="list-style-type: none"> 1. Поняття даних у програмному забезпеченні. Види обробки даних 2. .Вимоги збереження даних 3. Хешування даних 4. Поняття бази даних. Види баз даних. 5. NoSQL бази даних та їх класифікація. <p>Основний:1[с.75-82],2[с.85-97],4[с.72-90]. Додатковий:6[с.58-90],8[с.48-60],10[с.75-89]. Інтернет-ресурси:4,5</p> <p>Самостійна робота студентів. Підготовка до лабораторного заняття за матеріалами теми, вивчення основних та додаткових джерел, Інтернет-ресурсів, підготовка до тестування, виконання індивідуальних завдань1[с.12-32],2[с.8-20],4[с.21-32],6[с.20-44],8[с.28-40],9[с.14-32].</p> <p>Лабораторні заняття 6 Робота з базами даних у додатку</p> <p>Мета: навчитись взаємодіяти з базами даних</p> <p>2. Практичне завдання:</p> <ul style="list-style-type: none"> - <i>Ознайомитись із доступними базами даних</i> - <i>Підключить до бази даних MY SQL/MS SQL</i> - <i>Підключить до бази даних NOSQL (за вибором студента)</i> - <i>Створити хеш даних вбудованими засобами</i> 	<p>1</p> <p>14</p> <p>4</p>

Результати навчання	Навчальна діяльність	Робочий час студента, год.
<p>Вміти: Розробляти механізми взаємодії різних компонентів</p>	<p>- Реалізувати підключення до бази даних <i>MY SQL/MS SQL</i> з допомогою <i>ORM</i></p> <p>- Проаналізувати усі рішення, надати свою оцінку кожному із варіантів</p> <p style="text-align: center;">Тема 8. Структури даних мови С# План лекції 8</p> <ol style="list-style-type: none"> 1. Поняття структури даних. Основні структури даних. 2. Масиви. Їх види, застосування. Масив як базова структура даних. 3. Список як удосконалений вид масиву. Зв'язаний список та його напрями застосування. 4. Граф як спосіб класифікації даних. 5. Дерево як елемент для створення алгоритмів та ієрархій. 6. Стек та Черга як спосіб контролю входу та виходу даних. 7. Хеш-таблиця та її застосування. <p style="text-align: center;">8. Список рекомендованих джерел:</p> <p>Основний: 1[с.75-82], 2[с.85-97], 4[с.72-90]. Додатковий: 6[с.58-90], 8[с.48-60], 10[с.75-89]. Інтернет-ресурси: 4, 5</p> <p>Самостійна робота студентів. Підготовка до лабораторного заняття за матеріалами теми, вивчення основних та додаткових джерел, Інтернет-ресурсів, підготовка до тестування, виконання індивідуальних завдань 1[с.12-32], 2[с.8-20], 4[с.21-32], 6[с.20-44], 8[с.28-40], 9[с.14-32].</p>	<p style="text-align: center;">2</p> <p style="text-align: center;">1</p> <p style="text-align: center;">12</p>

Результати навчання	Навчальна діяльність	Робочий час студента, год.
	<p align="center">Лабораторні заняття 7</p> <p align="center">Реалізація структур даних на мові С#</p> <p>Мета: ознайомлення із можливостями мови програмування С#, основними типами даних математичними та логічними операціями</p> <p>1. Практичне завдання:</p> <ul style="list-style-type: none"> - На основі вивченої інформації розробити та реалізувати у середовищі Visual Studio: - Масив 2 та 3 вимірний - Стек - Бінарне дерево - Хеш таблиця - Список та зв'язаний список <p>Проаналізувати усі наведені структури та можливості їх застосування</p>	4
<p>Знати: Концепції написання чистого коду, базові питання підтримки</p> <p>Вміти: Працювати з сервером та структурами даних</p>	<p align="center">Тема 9. Додаткові можливості об'єктно орієнтованого програмування</p> <p align="center">План лекції 9</p> <ol style="list-style-type: none"> 1. Поняття делегатів. Роль делегатів у реалізації подій системи. 2. Виключення. Види та підходи до обробки. Конструкція try...catch...finally. Створення виключень у системі шляхом спадкування. Створення виключень з допомогою конструкції throw new exception. 3. Індексатори. 4. Методи розширення. Підходи до застосування методів розширення. Методи розширення 5. Часткові класи та методи. Підходи до застосування часткових класів. 6. Анонімні методи. 	1

Результати навчання	Навчальна діяльність	Робочий час студента, год.
	<p>7. Робота з кортежами.</p> <p>Список рекомендованих джерел: Основний: 1[с.75-82], 2[с.85-97], 4[с.72-90]. Додатковий: 6[с.58-90], 8[с.48-60], 10[с.75-89]. Інтернет-ресурси: 4, 5</p> <p>Самостійна робота студентів. Підготовка до лабораторного заняття за матеріалами теми, вивчення основних та додаткових джерел, Інтернет-ресурсів, підготовка до тестування, виконання індивідуальних завдань 1[с.12-32], 2[с.8-20], 4[с.21-32], 6[с.20-44], 8[с.28-40], 9[с.14-32].</p> <p>Лабораторні заняття 8 Проектування на мові C#</p> <p>Мета: удосконалити проект користувацького додатку, виділити основні класи у додатку, розробити систему виключень</p> <p>1. Практичне завдання:</p> <ul style="list-style-type: none"> - На основі обраного з викладачем проекту мобільного додатку розробити у UML діаграмі модель поведінки - Визначити точки можливих виключень та підготувати їх обробку - Реалізувати обробку анонімних типів - Створити методи для повернення кортежів 	<p>10</p> <p>4</p>
<p>Знати: Основні питання коректної роботи, діаграми поведінки</p>	<p>Тема 10. LINQ та його застосування. F# та його можливості у C#</p> <p>План лекції 10</p> <p>1. LINQ. Ключові слова. Директиви. Підходи до написання запитів. Проекція даних. Фільтрація даних. Сортування. Об'єднання перерізів. Агрегатні операції та групування. Делегати і запити у LINQ.</p>	<p>1</p>

Результати навчання	Навчальна діяльність	Робочий час студента, год.
<p>Вміти: Проводити написання функції для аналізу компонентів та навантаження</p>	<p>2. Рефлексія. Асинхронне програмування. 3. Pattern matching. Патерн типів. Паттерн властивостей. Паттерни кортежів. Позиційний патерн. Реляційний та логічний патерни. Паттерни списків. 4. F# як доповнення до мови C#. Підходи до застосування функціональної парадигми у об'єктно-орієнтованому програмуванні. 5. Типи даних у F#. Основні функції у F#. F# і ООП.</p> <p>Список рекомендованих джерел: Основний: 1[с.75-82],2[с.85-97],4[с.72-90]. Додатковий: 6[с.58-90],8[с.48-60],10[с.75-89]. Інтернет-ресурси: 4,5</p> <p>Самостійна робота студентів. Підготовка до лабораторного заняття за матеріалами теми, вивчення основних та додаткових джерел, Інтернет-ресурсів, підготовка до тестування, виконання індивідуальних завдань 1[с.12-32],2[с.8-20],4[с.21-32],6[с.20-44],8[с.28-40],9[с.14-32].</p> <p>Лабораторні заняття 9 Застосування LINQ</p> <p>Мета: навчитись аналізувати поведінку додатку Практичне завдання:</p> <ul style="list-style-type: none"> - Провести аналіз додатку та розроблених структур даних - Визначити мінімум задачі для групування фільтрації та агрегування даних з умовами - Запропонувати та реалізувати у своєму додатку ці задачі класичними методами 	<p>10</p> <p>2</p>

Результати навчання	Навчальна діяльність	Робочий час студента, год.
	<p>- Запропонувати та реалізувати у своєму додатку ці задачі з допомогою LINQ</p> <p>- Пояснити різницю та свій вибір</p> <p style="text-align: center;">Тема 11. Коректність і безпека крос-платформного програмного забезпечення</p> <p style="text-align: center;">План лекції 11</p> <ol style="list-style-type: none"> 1. Поняття коректності у крос-платформному програмному забезпеченні. Мінімізація впливу користувача. 2. Некоректна поведінка програмного забезпечення. Поняття виключення у програмних засобах. Обробка виключень. Фільтр виключень. 3. Користувацькі виключення. 4. Поняття безпеки програмного забезпечення. Ролі у системі. 5. Аутентифікація та авторизація у системі. Вимоги до паролів та шифрування. <p style="text-align: center;">Список рекомендованих джерел:</p> <p>Основний: 1[с.75-82],2[с.85-97],4[с.72-90]. Додатковий: 6[с.58-90],8[с.48-60],10[с.75-89]. Інтернет-ресурси: 4,5</p> <p style="text-align: center;">Самостійна робота студентів. Підготовка до лабораторного заняття за матеріалами теми, вивчення основних та додаткових джерел, Інтернет-ресурсів, підготовка до тестування, виконання індивідуальних завдань 1[с.12-32],2[с.8-20],4[с.21-32],6[с.20-44],8[с.28-40],9[с.14-32].</p>	1

Результати навчання	Навчальна діяльність	Робочий час студента, год.
	<p style="text-align: center;">Лабораторні заняття 10 Коректність роботи додатку</p> <p>Мета: навчитись аналізувати поведінку додатку</p> <p>Практичне завдання:</p> <ul style="list-style-type: none"> - <i>Визначити доцільність та підходи до перевірки вхідних параметрів</i> - <i>Запропонувати та реалізувати у своєму додатку рівні безпеки (мінімум 3)</i> - <i>Запропонувати та реалізувати у своєму додатку систему центру повідомлень</i> <p style="text-align: center;">Тема 12. Патерни у крос-платформному програмному забезпеченні План лекції 12</p> <ol style="list-style-type: none"> 1. Поняття патерну програмування. Їх переваги та недоліки. Класифікації патернів програмування. 2. Породжувальні патерни програмування. 3. Поведінкові патерни програмування. 4. Структурні патерни програмування. <p style="text-align: center;">Список рекомендованих джерел:</p> <p>Основний: 1[с.75-82],2[с.85-97],4[с.72-90]. Додатковий: 6[с.58-90],8[с.48-60],10[с.75-89]. Інтернет-ресурси:4,5</p> <p style="text-align: center;">Самостійна робота студентів. Підготовка до лабораторного заняття за матеріалами теми, вивчення основних та додаткових джерел, Інтернет-ресурсів, підготовка до тестування, виконання</p>	<p style="text-align: center;">4</p> <p style="text-align: center;">1</p> <p style="text-align: center;">10</p>

Результати навчання	Навчальна діяльність	Робочий час студента, год.
	<p>індивідуальних завдань 1[с.12-32],2[с.8-20],4[с.21-32],6[с.20-44],8[с.28-40],9[с.14-32].</p> <p style="text-align: center;">Лабораторні заняття 11 Патерни у додатках</p> <p>Мета: навчитись застосовувати поняття патернів при розробці додатку</p> <p>Практичне завдання:</p> <ul style="list-style-type: none"> - <i>Ознайомитись із поняттям патернів та їх основними видами</i> - <i>Визначити доцільність застосування патернів у власному додатку</i> - <i>Запропонувати та реалізувати у своєму додатку 2 породжуючих патерни</i> - <i>Запропонувати та реалізувати у своєму додатку 3 структурних патерни</i> <p style="text-align: center;">Запропонувати та реалізувати у своєму додатку 3 поведінкові патерни</p> <p style="text-align: center;">Тема 13. Логіювання та документування коду. Code convention.</p> <p style="text-align: center;">План лекції 13</p> <ol style="list-style-type: none"> 1. Поняття логіювання коду. Створення логів у системі. Правила створення логів. 2. Поняття Code convention. Підходи до іменування у програмному забезпеченні. Вимоги до іменування класів та інтерфейсів. 3. Різниця у іменуванні приватних та публічних полів. Іменування структур. 4. Види іменувань. Угорське іменування та його недоліки. 	<p style="text-align: center;">4</p> <p style="text-align: center;">1</p>

Результати навчання	Навчальна діяльність	Робочий час студента, год.
	<p>5. Поняття документування коду. Підходи до документування коду.</p> <p>6. Документування бізнес-логіки. Правила документування методів та виключень.</p> <p>Список рекомендованих джерел: Основний: 1[с.75-82], 2[с.85-97], 4[с.72-90]. Додатковий: 6[с.58-90], 8[с.48-60], 10[с.75-89]. Інтернет-ресурси: 4, 5</p> <p>Самостійна робота студентів. Підготовка до лабораторного заняття за матеріалами теми, вивчення основних та додаткових джерел, Інтернет-ресурсів, підготовка до тестування, виконання індивідуальних завдань 1[с.12-32], 2[с.8-20], 4[с.21-32], 6[с.20-44], 8[с.28-40], 9[с.14-32].</p> <p style="text-align: center;">Лабораторні заняття 12 Якість коду</p> <p>Мета: навчитись проводити аналіз коду</p> <p>Практичне завдання:</p> <ul style="list-style-type: none"> - <i>Аналізуючи свій додаток оцінити з викладачем такі показники:</i> - <i>Якість найменування змінних</i> - <i>Написання та обробку функцій та вхідних параметрів</i> - <i>Обробки виключень</i> - <i>Запропонувати шляхи для рефакторинку наявного коду з метою покращення його підтримки</i> 	<p style="text-align: center;">10</p> <p style="text-align: center;">4</p>

Результати навчання	Навчальна діяльність	Робочий час студента, год.
	<p style="text-align: center;">Тема 14. Тестування коду План лекції 14</p> <p>6. Поняття тестування коду. Поняття тесту. Правила тестування. Методи тестування. Види тестів.</p> <p>7. Тестування за знанням системи.</p> <p>8. Тестування за ступенем ізольованості компонентів. К</p> <p>9. Тестування за часом проведення.</p> <p>10. Тестування за об'єктом.</p> <p>11. Тестування продуктивності</p> <p style="text-align: center;">Список рекомендованих джерел:</p> <p>Основний: 1[с.75-82], 2[с.85-97], 4[с.72-90]. Додатковий: 6[с.58-90], 8[с.48-60], 10[с.75-89]. Інтернет-ресурси: 4, 5</p> <p>Самостійна робота студентів. Підготовка до лабораторного заняття за матеріалами теми, вивчення основних та додаткових джерел, Інтернет-ресурсів, підготовка до тестування, виконання індивідуальних завдань 1[с.12-32], 2[с.8-20], 4[с.21-32], 6[с.20-44], 8[с.28-40], 9[с.14-32].</p> <p style="text-align: center;">Лабораторні заняття 13 Тестування засобами C#</p> <p>Мета: навчитись проводити тестування додатків</p> <p>Практичне завдання:</p> <ul style="list-style-type: none"> - <i>Аналізуючи свій додаток розробити модульні тести для аналізу основних компонентів</i> - <i>Довести покриття тестами коду до 90%+</i> - <i>Провести інтеграційний тест з застосування зовнішнього фреймворку чи бібліотеки запропонованої викладачем</i> 	<p style="text-align: center;">1</p> <p style="text-align: center;">10</p> <p style="text-align: center;">2</p>

Результати навчання	Навчальна діяльність	Робочий час студента, год.
	<ul style="list-style-type: none"> - <i>Провести тестування безпеки</i> - <i>Провести тест інтерфейсу користувача</i> 	
<i>Підсумковий контроль – письмовий екзамен</i>		
	<i>Разом</i>	180

Примітка:

курсивом позначено інтерактивні методи навчання

3. СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ

Основний

1. PRICE, Mark J. *C# 8.0 and .NET Core 3.0—Modern Cross-Platform Development: Build applications with C#, .NET Core, Entity Framework Core, ASP.NET Core, and ML.NET using Visual Studio Code*. Packt Publishing Ltd, 2019.
2. ALBAHARI, Joseph. *C# 9.0 in a Nutshell*. " O'ReillyMedia, Inc.", 2021.
3. PRICE, Mark J. *C# 9 and .NET 5—Modern Cross-Platform Development: Build intelligent apps, websites, and services with Blazor, ASP.NET Core, and Entity Framework Core using Visual Studio Code*. Packt Publishing Ltd, 2020.
4. CLEARY, Stephen. *Concurrency in C# Cookbook: Asynchronous, Parallel, and Multithreaded Programming*. O'Reilly Media, 2019.
5. DOOLEY, John F.; DOOLEY. *SoftwareDevelopment, DesignandCoding*. Apress, 2017.

Додатковий

6. BILGIN, Can. *MasteringCross-PlatformDevelopmentwithXamarin*. PacktPublishingLtd, 2016
7. TURNER, Ryan. *C#: The Ultimate Beginner's Guide to Learn C# Programming Step by Step*. Publishing Factory, 2020.
8. SOLIS, Daniel; SCHROTENBOER, Cal. *Illustrated C# 7: The C# Language Presented Clearly, Concisely, and Visually*. Apress, 2018.
9. SKEET, Jonathan. *C# inDepth*. SimonandSchuster, 2019.
10. ALBAHARI, Joseph; ALBAHARI, Ben. *C# 10 Pocket Reference*. " O'Reilly Media, Inc.", 2022.
11. JAMRO, Marcin. *C# Data Structures and Algorithms: Explore the possibilities of C# for developing a variety of efficient applications*. Packt Publishing Ltd, 2018.
12. RUMPE, Bernhard. *Agilemodelingwith UML: codegeneration, Testing, Refactoring*. Cham: Springer, 2017.
13. XU, Jack. *Practical C# Charts and Graphics: Advanced Chart and Graphics Programming for Real-World .NET Applications*. UniCAD, 2019.

Інтернет-ресурси

1. Типи даних. URL: https://uk.wikipedia.org/wiki/%D0%A2%D0%B8%D0%BF_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85
2. Структура даних. URL: https://uk.wikipedia.org/wiki/%D0%A1%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85
3. Патерни програмування. URL: <https://refactoring.guru/uk/design-patterns>
4. Документація на C#. URL: <https://docs.microsoft.com/ru-ru/dotnet/csharp/>

* Курсивом виділені джерела, що є у бібліотеці ДТЕУ, або наявні повнотекстові електронні версії джерел.