

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ТЕХНОЛОГІЙ ТА
ДИЗАЙНУ

Кафедра інформаційних та комп'ютерних технологій

ЗАТВЕРДЖУЮ



РОБОЧА ПРОГРАМА

Навчальної дисципліни «Об'єктно-орієнтоване програмування»

Рівень вищої освіти перший (бакалаврський)

Спеціальність 121 Інженерія програмного забезпечення

Освітня програма Інженерія програмного забезпечення

Факультет Мехатроніки та комп'ютерних технологій

Київ 2024 рік

РОЗРОБЛЕНО: Київський національний університет технологій та дизайну

РОЗРОБНИК ПРОГРАМИ: Пилипенко В.І., старший викладач кафедри інформаційних та комп'ютерних технологій.

Схвалено Вченою Радою факультету мехатроніки та комп'ютерних технологій

Протокол від « 12 » серпня 2024 року № 10

Декан факультету МКТ БЗ Борис ЗЛОТЕНКО

Обговорено та рекомендовано на засіданні кафедри інформаційних та комп'ютерних технологій

Протокол від « 06 » серпня 2024 року № 14

Завідувач кафедри ІКТ В Владислава СКІДАН

1. ОПИС ДИСЦИПЛІНИ

Найменування показників	Галузь знань, спеціальність, освітня програма, рівень вищої освіти	Характеристика навчальної дисципліни	
		денна форма здобуття вищої освіти	заочна форма здобуття вищої освіти
Кількість кредитів – 8	Галузь знань – 12 Інформаційні технології.	обов'язкова	
Змістовні модулі – 3		Рік підготовки	
		2-й	2-й
Індивідуальне науково-дослідне завдання – розрахунково-графічна робота 2 семестр	Галузь знань 12 Спеціальність – 121 Інженерія програмного забезпечення	Семестр	
		3,4-й	3,4-й
		Лекції	
		36 год.	8 год.
Загальна кількість годин - 240	Освітня програма Інженерія програмного забезпечення	Практичні, семінарські	
		-	-
		Лабораторні	
		60 год.	12 год.
		Самостійна робота	
		114 год.	190 год.
		Індивідуальні завдання	
		30 год.	30 год.
		Вид підсумкового контролю	
Тижневих годин для денної форми здобуття вищої освіти: Аудиторних: (1 семестр – 4 год.) (2 семестр – 4 год.)	Рівень вищої освіти перший (бакалаврський)	Екзамен (семестр 3), Екзамен (семестр 4).	

2. АНОТАЦІЯ ДИСЦИПЛІНИ

Мета курсу – оволодіння компетентностями: здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування; здатність розробки архітектури, модулів та компонентів програмних систем; здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення; здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення; здатність до алгоритмічного та логічного мислення.

Результати навчання дисципліни:

знати: основні процеси, фази та ітерації життєвого циклу програмного забезпечення; знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-

орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення; знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення;

вміти: вибирати та використовувати відповідну задачі методологію створення програмного забезпечення; застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення; вміти документувати та презентувати результати розробки програмного забезпечення;

здатен продемонструвати: здатність до застосування знань у практичних ситуаціях;

самостійно вирішувати: питання щодо застосування набутих знань у практичних ситуаціях та самостійно оволодівати сучасними знаннями.

Програмні результати навчання: ПРН 3, ПРН 5, ПРН 6, ПРН 7, ПРН 12, ПРН 23.

Необхідні навчальні компоненти (пререквізити, кореквізити і постреквізити): іноземна мова, комп'ютерні технології програмування, основи системного аналізу, математичні методи дослідження операцій та прийняття рішень, аналіз вимог до програмного забезпечення, WEB-технології.

Види навчальних занять: лекція, лабораторне, консультація.

Методи навчання: словесний, пояснювально-демонстраційний, метод проблемного викладання, дослідницький, метод закріплення на практиці.

Методи контролю: усний, письмовий, практичний, тестовий.

Форми підсумкового контролю: екзамен (семестр 3), екзамен (семестр 4), захист курсової роботи (семестр 4).

Засоби діагностики успішності навчання: індивідуальні завдання, питання для модульного контролю, питання для підсумкового контролю, тести.

Мова навчання: українська.

3. ПРОГРАМА ДИСЦИПЛІНИ

СЕМЕСТР 3

Змістовий модуль 1. Введення до ООП.

ТЕМА 1. Введення до об'єктно-орієнтованого програмування.

Основи об'єктно-орієнтованого підходу. Ключові концепції: об'єкти, класи, інкапсуляція, успадкування, поліморфізм. Порівняння ООП з процедурним програмуванням.

ТЕМА 2. Класи та об'єкти.

Класи в C#. Створення об'єктів та робота з ними. Використання this. Оператори is та as. Простори імен.

ТЕМА 3. Поля в класах.

Поля класу. Константи та readonly поля. Модифікатори рівня доступу. Області видимості. Інкапсуляція.

ТЕМА 4. Конструктори класів.

Конструктори. Перевантаження конструкторів. Статичний конструктор. Деструктори.

ТЕМА 5. Методи та властивості в класах.

Властивості. Методи. Перевантаження методів. Поліморфізм. Успадкування.

ТЕМА 6. Побудова діаграм класів в ООП.

Методи для візуалізації класів, їх властивостей та зв'язків. Розбір принципів створення UML діаграм класів та їх значення в розробці програм.

ТЕМА 7. Інтерфейси.

Поняття інтерфейсів. Інтерфейс IComparable. Статичні інтерфейси. Інтерфейси та серіалізація. Інтерфейси та їх використання в тестуванні.

ТЕМА 8. Колекції. Класи колекцій.

Вбудовані колекції в .NET Framework. Операції з колекціями. Оптимізація роботи з колекціями. Серіалізація та десеріалізація колекцій.

ТЕМА 9. Файлове введення та виведення.

Основи роботи з файлами. Використання потоків у файловому введенні та виведенні. Робота з файловими винятками. Серіалізація та десеріалізація об'єктів у файли. Робота з текстовими потоками.

ТЕМА 10. Створення ієрархії класів.

Концепція ієрархії класів в ООП. Як використовувати успадкування для побудови ієрархій. Практичне застосування для оптимізації та спрощення коду.

СЕМЕСТР 4

Змістовий модуль 2. Застосування принципів ООП.

ТЕМА 11. Поліморфізм.

Типи поліморфізму: статичний і динамічний. Перевантаження методів та перевизначення. Використання поліморфізму для створення гнучких і розширюваних програм

ТЕМА 12. Абстракції в ООП.

Роль абстракції в ООП. Як створювати абстрактні класи та методи. Використання абстракцій для спрощення проектування та підвищення модульності програм.

ТЕМА 13. Патерни проектування: Введення.

Огляд патернів проектування. Їх значення та роль у створенні масштабованих і підтримуваних додатків. Класифікація патернів: створення, структурні та поведінкові.

ТЕМА 14. Патерни проектування: Створення об'єктів.

Основні патерни проектування для створення об'єктів: Singleton, Factory, Abstract Factory. Як правильно вибрати патерни для ефективної організації коду.

ТЕМА 15. Патерни проектування: Структурні патерни.

Структурні патерни проектування: Adapter, Composite, Decorator, Facade, Proxy. Використання патернів для зручного створення гнучких та підтримуваних архітектур.

ТЕМА 16. Патерни проектування: Поведінкові патерни.

Поведінкові патерни проектування: Strategy, Observer, Command, Chain of Responsibility. Вплив на зміну поведінки об'єктів і взаємодії між ними.

ТЕМА 17. Створення та налагодження додатку Windows Forms.

Створення додатка в Windows Forms. Відлагодження додатка. Розгортання додатка. Оптимізація продуктивності.

ТЕМА 18. Види елементів управління в Windows Forms та робота з ними.

Розміщення елементів управління на формі. Елементи для введення даних. Елементи навігації. Елементи відображення даних. Налаштування властивостей елементів управління. Робота з подіями елементів управління.

ТЕМА 19. Створення візуального інтерфейсу в Windows Forms.

Розробка інтерфейсу. Налаштування вигляду та стилізація. Адаптивний дизайн та масштабування.

ТЕМА 20. Створення додатку в Windows Forms на основі ООП.

Інтеграція об'єктно-орієнтованого підходу в розробку додатків Windows Forms. Практичне застосування принципів ООП для побудови зручних та ефективних додатків.

4. СТРУКТУРА ДИСЦИПЛІНИ

Назви змістових модулів і тем	Кількість годин											
	Денна форма здобуття вищої освіти						Заочна форма здобуття вищої освіти					
	усього	у тому числі:					усього	у тому числі:				
		л	пр	лаб	інд	СРС		л	пр	лаб	інд	СРС
1	2	3	4	5	6	7	8	9	10	11	12	13
Модуль 1												
Семестр 3-й												
Змістовий модуль 1. Введення до ООП												
Тема 1. Введення до об'єктно-орієнтованого програмування.	6	1		0		5	9,5	0,5		0		9
Тема 2. Класи та об'єкти.	9	1		3		5	9,5	0,5		0		9
Тема 3 Поля в класах.	10	2		3		5	10	0,5		0,5		9
Тема 4. Конструктори класів.	10	2		3		5	10	0,5		0,5		9
Тема 5. Методи та властивості в класах.	10	2		3		5	10	0,5		0,5		9
Тема 6. Побудова діаграм класів в ООП.	10	2		3		5	10	0,5		0,5		9
Тема 7.Інтерфейси.	10	2		3		5	10	0		1		9
Тема 8. Колекції. Класи колекцій.	10	2		3		5	10	0		1		9
Тема 9. Файлове введення та виведення.	10	2		3		5	10,5	0,5		1		9
Тема 10. Створення ієрархії класів.	13	2		6		5	10,5	0,5		1		9
Модуль 2												
Семестр 4-й												
Змістовий модуль 2. Застосування принципів ООП												
Тема 11. Поліморфізм.	9	1		3		5	11	0,5		0,5		10
Тема 12. Абстракції в ООП.	9	1		3		5	11	0,5		0,5		10
Тема 13. Патерни проектування: Введення.	10	2		3		5	11,5	0,5		1		10

Тема 14. Патерни проектування: Створення об'єктів.	10	2		3		5	11,5	0,5		1		10
Тема 15. Патерни проектування: Структурні патерни.	10	2		3		5	11,5	0,5		1		10
Тема 16. Патерни проектування: Поведінкові патерни.	10	2		3		5	11,5	0,5		1		10
Тема 17. Створення та налагодження додатку Windows Forms.	10	2		0		5	10,5	0		0,5		10
Тема 18. Види елементів управління в Windows Forms та робота з ними.	10	2		3		5	10	0		0		10
Тема 19. Створення візуального інтерфейсу в Windows Forms.	15	2		3		10	10,5	0,5		0		10
Тема 20. Створення додатку в Windows Forms на основі ООП.	19	2		6		14	11	0,5		0,5		10
Разом за модуль 1 та 2	210	36		60		114	210	8		12		190
Модуль 3												
ІНДЗ (курсова робота)	30				30		30				30	
Всього	240	36		60	30	114	240	8		12	30	190

5. ЛАБОРАТОРНІ ЗАНЯТТЯ

№ з/п	Назва теми лабораторного заняття	Кількість годин
1	Робота з класами та об'єктами	3
2	Робота з полями класу. Специфікатори доступу.	3
3	Конструктори класів.	3
4	Властивості та методи.	3

5	Побудова діаграм класів.	3
6	Використання інтерфейсів.	3
7	Робота з колекціями: List, Dictionary	3
8	Файлове введення та виведення.	3
9	Створення ієрархії класів.	6
10	Застосування поліморфізму як парадигми ООП.	3
11	Робота з структурними патернами проектування.	3
12	Робота з поведінковими патернами проектування.	3
13	Застосування паттерну по створенню об'єктів.	3
14	Використання абстракцій для спрощення коду.	3
15	Інкапсуляція та її застосування в реальному кодї.	3
16	Підбір паттерну проектування.	3
17	Створення візуального інтерфейсу в Windows Forms.	3
18	Створення додатку в Windows Forms на основі ООП.	6
Всього		60

6. САМОСТІЙНА РОБОТА

№ з/п	Назва теми лабораторного заняття	Кількість годин
1	Підготовка до виступу на лекціях, лабораторних заняттях	34
2	Самостійне опрацювання питань, які не виносились на лекцію	46
3	Підготовка до всіх видів контролю	20
4	Індивідуальні завдання	14
Всього		114

7. ІНДИВІДУАЛЬНЕ НАУКОВО-ДОСЛІДНЕ ЗАВДАННЯ

Індивідуальне науково-дослідне завдання – виконання курсової роботи в 4 семестрі – 30 годин.

8. РОЗПОДІЛ БАЛІВ, ЯКІ ОТРИМУЮТЬ СТУДЕНТИ

Розподіл балів, які отримують здобувачі вищої освіти

Екзамен (3 семестр)

Поточне оцінювання та самостійна робота												ПК (тестовий)	Екзамен	Сума
T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12			
6	8	8	8	8	8	6	6	6	6	5	5	10	10	100

Розподіл балів за видами робіт

Види робіт, що оцінюються в балах	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	Усього
Опрацювання теоретичного матеріалу	2	2	2	2	3	3	2	2	2	2	2	2	26
Виконання та захист лабораторних робіт	4	6	6	6	5	5	4	4	4	4	3	3	54
Підсумковий контроль	10												10
Екзамен	10												10
Всього з дисципліни													100

Екзамен (4 семестр)

Поточне оцінювання та самостійна робота								ПК (тестовий)	Екзамен	Сума
T13	T14	T15	T16	T17	T18	T19	T20			
5	5	10	10	10	10	10	10	20	10	100

Розподіл балів за видами робіт

Види робіт, що оцінюються в балах	T13	T14	T15	T16	T17	T18	T19	T20	Усього
Опрацювання теоретичного матеріалу	2	2	3	3	3	3	2	2	20
Виконання та захист лабораторних робіт	3	3	7	7	7	7	8	8	50
Підсумковий контроль	20								20
Екзамен	10								10
Всього з дисципліни									100

Критерії оцінювання видів робіт

Завдання оцінюються у відсотках від кількості балів відведених на відповідний вид роботи за темою із заокругленням до цілого числа залежно від вагомості помилок:

0% – завдання не виконано;

30% – завдання виконано частково і містить суттєві помилки під час виконання та захисту роботи;

60% – завдання виконано повністю, але містить суттєві помилки під час розрахунків та захисту роботи;

80% – завдання виконано повністю і вчасно, але містить несуттєві помилки у виконаних завданнях та під час захисту роботи;

100 % – завдання виконано у повному обсязі та захищено без помилок та зауважень.

Виконання курсової роботи

Пояснювальна записка	Ілюстративна частина	Захист роботи	Сума
40	40	20	100

Критерії оцінювання екзамену

Екзамен проводиться у вигляді тестування та оцінюється від 0 до 10 балів. Загальна кількість тестових питань – 10. Кожна правильна відповідь оцінюється в 1 бал.

Відповідність шкал оцінок якості засвоєння навчального матеріалу

Оцінка за національною шкалою	Оцінка в балах	Оцінка за шкалою ECTS	Пояснення
відмінно	90-100	A	Відмінно (відмінне виконання лише з незначною кількістю помилок)
добре	82-89	B	Дуже добре (вище середнього рівня з кількома помилками)
	74-81	C	Добре (в загальному вірне виконання з певною кількістю суттєвих помилок)
задовільно	64-73	D	Задовільно (непогано, але зі значною кількістю недоліків)
	60-63	E	Достатньо (виконання відповідає мінімальним критеріям)
незадовільно	35-59	FX	Незадовільно (з можливістю повторного складання)
	0-34	F	Незадовільно (з обов'язковим повторним вивченням дисципліни)

9. ПОЛІТИКА КУРСУ

3.1. Обов'язкове дотримання академічної доброчесності здобувачами вищої освіти, а саме:

- самостійне виконання всіх видів робіт, завдань, форм контролю, передбачених робочою програмою даної навчальної дисципліни;
- посилання на джерела інформації у разі використання ідей, розробок, тверджень, відомостей;
- дотримання норм законодавства про авторське право і суміжні права;
- надання достовірної інформації про результати власної навчальної (наукової, творчої) діяльності, використанні методики досліджень і джерела інформації.

3.2. Оцінювання проводиться відповідно до таблиці розподілу балів з дисципліни.

3.3. Для отримання мінімальної позитивної оцінки необхідно набрати мінімальну кількість балів за кожен вид робіт, що оцінюється (в сумі – 60 балів).

3.4. Для отримання максимального балу з дисципліни необхідно виконати й захистити усі завдання в повному обсязі (90 балів під час семестру) і 10 балів під час екзамену.

3.5. У випадку несвоєчасного виконання запланованих робіт здобувач вищої освіти погоджує з викладачем нові терміни їх здачі але не пізніше дати початку екзаменаційної сесії.

3.6. При виявленні ознак плагіату робота повертається на доопрацювання. У випадку повторного виявлення плагіату робота анулюється і видається новий варіант завдання.

3.7. Теоретичний матеріал пропущених лекційних занять здобувач опрацьовує в МСОП, у разі потреби може отримати консультацію викладача.

3.8. Пропущені лабораторні заняття повинні бути самостійно відпрацьовані та захищені у терміни погоджені з викладачем.

3.9. У разі робочої потреби в ході лекційних та лабораторних занять студенти використовують мобільні пристрої, як інформаційно-комунікаційний інструмент.

3.10. Оскарження оцінювання за конкретним видом роботи можливе з перескладанням, але не пізніше екзаменаційної сесії

3.11. Допускається визнання результатів навчання здобувачів вищої освіти, отриманих у неформальній освіті.

10.РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна література

1. Troelsen, A., & Japikse, P. (2020). Pro C# 8 with. NET Core 3: Foundational Principles and Practices in Programming. Apress.
2. Albahari, J. (2022). C# 10 in a nutshell: the definitive reference. " O'Reilly Media, Inc."

3. Bancila, M., Rialdi, R., & Sharma, A. (2020). Learn C# Programming: A guide to building a solid foundation in C# language for writing efficient programs. Packt Publishing Ltd.
4. Ferrone, H. (2020). Learning C# by Developing Games with Unity 2020: An enjoyable and intuitive approach to getting started with C# programming and Unity. Packt Publishing Ltd.
5. Ćerim, H. (2020). Extending C# with a Library of Functional Programming Concepts.
6. McGrath, M. (2020). C# Programming in easy steps: Updated for Visual Studio 2019. In Easy Steps.
7. Sarcar, V. (2021). Simple and Efficient Programming with C#. Apress.
8. Price, M. J. (2024). C# 13 and .NET 9—Modern Cross-Platform Development Fundamentals. Packt Publishing Ltd.
9. Sharp, J. (2022). Microsoft Visual C# step by step. Microsoft Press.
10. Nagel, C. (2021). Professional C# and .NET. John Wiley & Sons.
11. Albahari, J. (2021). C# 9.0 in a Nutshell. " O'Reilly Media, Inc."
12. Ferrone, H. (2021). Learning C# by Developing Games with Unity 2021: Kickstart your C# programming and Unity journey by building 3D games from scratch. Packt Publishing Ltd.
13. Akella, R., Tamirisa, A. K., Kunani, S. K., & Muthiyalu, B. G. (2021). Enterprise Application Development with C# 9 and .NET 5: Enhance your C# and .NET skills by mastering the process of developing professional-grade web applications. Packt Publishing Ltd.
14. Krpan, D., Mladenović, S., & Zaharija, G. (2022). Teaching object oriented programming: programming language challenge. In ICERI2022 Proceedings (pp. 5880-5887). IATED.
15. Ogala, J. O., & Ojie, D. V. (2020). Comparative analysis of c, c++, c# and java programming languages. GSJ, 8(5), 1899-1913.
16. Pinheiro, P., Viana, J. C., Ribeiro, M., Fernandes, L., Ferrari, F., Gheyi, R., & Fonseca, B. (2020). Mutating code annotations: An empirical evaluation on Java and C# programs. Science of Computer Programming, 191, 102418.
17. Abbas, S. H., & Ranjan, E. R. (2022). LEARN .NET WITH PROGRAMMING (3-in-1): Covers .NET using C#, Visual Basic ASP .NET. Shrihind Publications Pvt Ltd.
18. Painter, S. J. (2023). Functional Programming with C#: Create More Supportable, Robust, and Testable Code. " O'Reilly Media, Inc."
19. Clark, D., & Sanders, J. (2011). Beginning C# object-oriented programming (Vol. 1). np: Apress.

ПОГОДЖЕНО

Завідувач кафедри
інформаційних та комп'ютерних
технологій



Владислава СКІДАН

«06» червня 2024 р.

ПЕРЕЗАТВЕРДЖЕНО

Протокол засідання кафедри від « ____ » _____ 20__ року № ____
Завідувач кафедри _____

підпис

Власне ім'я ПРІЗВИЩЕ

ПЕРЕЗАТВЕРДЖЕНО

Протокол засідання кафедри від « ____ » _____ 20__ року № ____
Завідувач кафедри _____

підпис

Власне ім'я ПРІЗВИЩЕ