

Київський університет імені Бориса Грінченка
Факультет інформаційних технологій та математики
Кафедра інформаційної та кібернетичної безпеки
імені професора Володимира Бурячка

«ЗАТВЕРДЖУЮ»

Проректор з науково-методичної
та навчальної роботи


Олексій ЖИЛЬЦОВ
« » 2023

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
«ПРОГРАМУВАННЯ: ОБ'ЄКТНО-ОРІЄНТОВАНЕ
ПРОГРАМУВАННЯ»

для студентів

спеціальності	123 Комп'ютерна інженерія
освітнього рівня	першого (бакалаврського)
освітньої програми	123.00.01 Комп'ютерна інженерія

2023 – 2024 навчальний рік



Розробник:

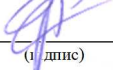
Бєбєшко Богдан Тарасович, старший викладач кафедри інформаційної та кібернетичної безпеки імені професора Володимира Бурячка Факультету інформаційних технологій та математики Київського університету імені Бориса Грінченка.

Викладач:

Бєбєшко Богдан Тарасович, старший викладач кафедри інформаційної та кібернетичної безпеки імені професора Володимира Бурячка Факультету інформаційних технологій та математики Київського університету імені Бориса Грінченка.

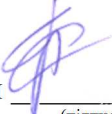
Робочу програму розглянуто і затверджено на засіданні кафедри інформаційної та кібернетичної безпеки імені професора Володимира Бурячка

Протокол від 17.10.2023 р. № 10

Завідувач кафедри  Павло СКЛАДАННИЙ
(підпис)


Робочу програму погоджено з гарантом освітньої програми (керівником освітньої програми 123.00.01 Комп'ютерна інженерія)

____.____. 2023 р.

Керівник освітньої програми  Павло СКЛАДАННИЙ
(підпис)

Робочу програму перевірено

____.____. 2023 р.

Заступник декана  Євген ІВАНІЧЕНКО
(підпис)

Пролонговано:

на 20__/20__ н.р. _____ (_____), «____» ____ 20__ р., протокол № ____
(підпис) (ПІБ)

на 20__/20__ н.р. _____ (_____), «____» ____ 20__ р., протокол № ____
(підпис) (ПІБ)

на 20__/20__ н.р. _____ (_____), «____» ____ 20__ р., протокол № ____
(підпис) (ПІБ)

на 20__/20__ н.р. _____ (_____), «____» ____ 20__ р., протокол № ____
(підпис) (ПІБ)

1. Опис навчальної дисципліни

Найменування показників	Характеристика дисципліни за формами навчання	
	денна	заочна
Вид дисципліни	обов'язкова	
Мова викладання, навчання та оцінювання	українська	
Загальний обсяг кредитів / годин	3 / 90	
Курс	2	
Семестр	3	
Кількість змістових модулів з розподілом:	3	
Обсяг кредитів	3	
Обсяг годин, в тому числі:	90	
Аудиторні	42	
Модульний контроль	6	
Семестровий контроль	25	
Самостійна робота	17	
Форма семестрового контролю	Залік	

2. Мета та завдання навчальної дисципліни

Робоча навчальна програма з курсу «Програмування: Об'єктно-орієнтоване програмування» є нормативним документом Київського університету імені Бориса Грінченка, який розроблено кафедрою інформаційної та кібернетичної безпеки імені професора Володимира Бурячка на основі освітньо-професійної програми підготовки здобувачів першого (бакалаврського) рівня відповідно до навчального плану спеціальності 123 Комп'ютерна інженерія, освітньої програми 123.00.01 Комп'ютерна інженерія.

Робочу навчальну програму укладено згідно з вимогами Європейської кредитної трансферно-накопичувальної системи (ЄКТС) організації навчання.

Програма визначає обсяги знань, які повинен опанувати здобувач першого (бакалаврського) рівня відповідно до вимог освітньо-кваліфікаційної характеристики, алгоритму вивчення навчального матеріалу дисципліни «Програмування: Об'єктно-орієнтоване програмування» та необхідне методичне забезпечення, складові і технологію оцінювання навчальних досягнень студентів.

Навчальна дисципліна «Програмування: Об'єктно-орієнтоване програмування» складається з трьох змістових модулів: «Вступ до ООП», «Аспекти програмування на С#», «Розширені концепції С#». Обсяг дисципліни – 90 год (3 кредити).

Метою викладання навчальної дисципліни «Програмування: Об'єктно-орієнтоване програмування» є формування у майбутніх фахівців необхідного рівня об'єктно-орієнтованого програмування; ознайомлення з сучасними мовами об'єктно-орієнтованого програмування та сучасними засобами розробки програм; набуття первинних знань з об'єктно-орієнтованого аналізу та проектування інформаційних систем.

Завдання:

- вивчення базових понять об'єктно-орієнтованого підходу до розробки програм, синтаксису та семантики сучасних об'єктно-орієнтованих мов програмування (на прикладі мови програмування С#);
- знайомство з об'єктно-орієнтованою технологією розробки програмного забезпечення (об'єктно-орієнтований аналіз предметної області, об'єктно-орієнтоване проектування та програмування ПЗ);
- оволодіння сучасними інструментальними програмними засобами розробки об'єктно-орієнтованого програмного забезпечення.

У результаті вивчення навчальної дисципліни формуються наступні **загальні та фахові компетентності:**

загальні компетентності:

ЗК 3 – Здатність застосовувати знання у практичних ситуаціях;

ЗК 7 – Вміння виявляти, ставити та вирішувати проблеми;

фахові компетентності:

ФК 2 – Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.

ФК 3 – Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж.

ФК 7 – Здатність використовувати та впроваджувати нові технології, включаючи технології розумних, мобільних, зелених і безпечних обчислень, брати участь в модернізації та реконструкції комп'ютерних систем та мереж, різноманітних вбудованих і розподілених додатків, зокрема з метою підвищення їх ефективності.

ФК 11 – Здатність оформляти отримані робочі результати у вигляді презентацій, науково-технічних звітів.

ФК 12 – Здатність ідентифікувати, класифікувати та описувати роботу програмно-технічних засобів, комп'ютерних та кіберфізичних систем, мереж та їхніх компонентів шляхом використання аналітичних методів і методів моделювання.

ФК 13 – Здатність вирішувати проблеми у галузі комп'ютерних та інформаційних технологій, визначати обмеження цих технологій..

3. Результати навчання за дисципліною

У результаті вивчення навчальної дисципліни студент повинен

знати:

- Основні концепції та принципи об'єктно-орієнтованого програмування (ООП).
- Структуру та елементи об'єктно-орієнтованих мов програмування.
- Особливості синтаксису мов об'єктно-орієнтованого програмування, зокрема C#.
- Принципи наслідування, поліморфізму, інкапсуляції та абстракції в ООП.
- Використання класів, об'єктів, інтерфейсів та засобів взаємодії між ними.

уміти:

- Розробляти та реалізовувати програми за допомогою об'єктно-орієнтованих мов програмування, зокрема C#.
- Застосовувати принципи ООП для ефективного проектування та реалізації програм.
- Використовувати класи, методи, об'єкти та інші компоненти ООП при розробці програм на мові C#.
- Аналізувати задачі та вибирати оптимальний об'єктно-орієнтований підхід для їх рішення.
- Використовувати об'єктно-орієнтовані технології для розробки масштабованих та модульних програмних систем.

та досягти наступних **програмних результатів навчання:**

РН 2 – Мати навички проведення експериментів, збирання даних та моделювання в комп'ютерних системах;

РН 7 – Вміти розв'язувати задачі аналізу та синтезу засобів, характерних для спеціальності;

РН 10 – Вміти розробляти програмне забезпечення для вбудованих і розподілених застосувань, мобільних і гібридних систем, розраховувати, експлуатувати, типове для спеціальності обладнання.

4. Структура навчальної дисципліни

Тематичний план для денної форми навчання

Назва змістових модулів, тем	Усього	Розподіл годин між видами робіт					
		Аудиторна:					Самостійна
		Лекції	Семинари	Практичні	Лабораторні	Індивідуальні	
Змістовий модуль 1. Вступ до ООП							
Тема 1. Основи об'єктно-орієнтованого аналізу та проектування програмного забезпечення. Концепція класів та об'єктів.	4	1		2			1
Тема 2. Основні поняття об'єктно-орієнтованого програмування: інкапсуляція.	4	1			2		1
Тема 3. Основні поняття об'єктно-орієнтованого програмування: спадковість, просте та множинне успадкування.	5	1		2			2
Тема 4. Основні поняття об'єктно-орієнтованого програмування: поліморфізм, абстракція.	4	1			2		1
Модульний контроль	2						
Разом	19	4		4	4		5
Змістовий модуль 2. Аспекти програмування на C#							
Тема 5. Інтерфейси та їх реалізація на прикладі мови програмування C#. Концепція проектування через інтерфейси.	4	1		2			1
Тема 6. Перевантаження функцій та перевантаження операцій на прикладі мови програмування C#.	5	1			2		2
Тема 7. Універсальні шаблони функцій і класів на прикладі мови програмування C#. Обмеження універсальних шаблонів.	5	1		2			2
Тема 8. Делегати; поняття делегатів; призначення та використання делегатів на прикладі мови програмування C#.	4	1			2		1
Модульний контроль	2						
Разом	20	4		4	4		6
Змістовий модуль 3. Розширені концепції C#							
Тема 9. Події. Поняття подій в об'єктно-орієнтованому програмуванні на прикладі мови програмування C#. Концепція використання подій та делегатів.	4	1		2			1
Тема 10. Потoki на прикладі мови програмування C#. Багатопоточність. Основні концепції синхронізації потоків.	7	1			4		2
Тема 11. Методологія дебагу проєкту. Оброблення виняткових ситуацій на прикладі мови програмування C#.	6	1		4			1

Назва змістових модулів, тем	Усього	Розподіл годин між видами робіт					
		Аудиторна:					Самостійна
		Лекції	Семінари	Практичні	Лабораторні	Індивідуальні	
Тема 12. Шаблони проектування: користь та переваги шаблонів проектування, класифікація шаблонів проектування, недоліки шаблонів проектування.	7	1			4		2
Модульний контроль	2						
Разом	26	4		6	8		6
Семестровий контроль	25						
Усього	90	12		14	16		17

5. Програма навчальної дисципліни

Змістовий модуль 1. Вступ до ООП

Тема 1. Основи об'єктно-орієнтованого аналізу та проектування програмного забезпечення. Концепція класів та об'єктів.

Поняття об'єктно-орієнтованого аналізу предметного середовища. Загальні принципи побудови об'єктної моделі предметного середовища. Поняття об'єктно-орієнтованого проектування програмного забезпечення. Поняття класу та об'єкту (екземпляру класу). Поля та методи класу. Взаємовідносини між класами та об'єктами. Поняття успадкування. Композиція та колекція об'єктів. Опис класу. Конструктори, деструктори класів. Створення екземпляру класу. Масиви об'єктів. Статичні, константні члени класів. Огляд діаграми класів. Введення поняття відносин між класами: асоціація, агрегація, композиція, реалізація, самоасоціація залежностей.

Тема 2. Основні поняття об'єктно-орієнтованого програмування: інкапсуляція.

Поняття інкапсуляції. Директиви видимості класів C# за замовчанням. Організація доступу до захищених полів класу. Дружні функції та класи.

Тема 3. Основні поняття об'єктно-орієнтованого програмування: спадковість, просте та множинне успадкування.

Поняття спадковості. Опис класів-нащадків в C# при одиничній спадковості. Директиви видимості при спадкуванні в C#. Явний виклик конструкторів-предків. Послідовність виклику конструкторів/ деструкторів для ієрархії класів. Множинна спадковість в C#. Проблеми ромбовидної спадковості, віртуальні предки.

Тема 4. Основні поняття об'єктно-орієнтованого програмування: поліморфізм, абстракція.

Поняття поліморфізму. Види поліморфізму. Практичне застосування основних властивостей та переваг поліморфізму на прикладі мови програмування C#. Вивчення поняття ієрархії класів. Поняття абстракції. Логічне виділення та реалізація/створення абстрактних класів та методів (функцій) зважаючи на чіткі концептуальні межі об'єкта. Використання абстрактних класів та методів.

Змістовий модуль 2. Аспекти програмування на C#

Тема 5. Інтерфейси та їх реалізація на прикладі мови програмування C#. Концепція проектування через інтерфейси.

Поняття інтерфейсу. Принципи створення і визначення інтерфейсів. Реалізація інтерфейсів. Особливості явної та неявної імплементації інтерфейсів. Спадкування інтерфейсів. Огляд концепції швидкого проектування додатку через інтерфейси.

Тема 6. Перевантаження функцій та перевантаження операцій на прикладі мови програмування C#.

Неоднозначність при перевантаженні (сумісність типів параметрів, параметри-посилання, параметри за замовчуванням). Перевантаження методів класу та конструкторів. Перевантаження бінарних та унарних операцій. Перевантаження операції індексування. Перевантаження операції присвоєння. Особливості присвоєння об'єктів мови C#. Перевантаження операцій потокового введення-виведення.

Тема 7. Універсальні шаблони функцій і класів на прикладі мови програмування C#. Обмеження універсальних шаблонів.

Поняття шаблону функції та класів. Опис шаблону. Створення шаблону функції. Відмінності перевантаження функції та шаблонів функції. Опис шаблону-класу. Створення екземплярів шаблону класу. Принципи та механізми обмеження шаблонів. Робота з Nullable типами даних. Практичні приклади роботи з операціями поглинання.

Тема 8. Делегати; поняття делегатів; призначення та використання делегатів на прикладі мови програмування C#.

Поняття делегату та лямбда-виразів. Можливість передачі частини виконуваного коду через аргументи. Створення та використання делегатів. Особливості та обмеження можливостей використання делегатів.

Змістовий модуль 3. Розширені концепції C#

Тема 9. Події. Поняття подій в об'єктно-орієнтованому програмуванні на прикладі мови програмування C#. Концепція використання подій та делегатів.

Поняття події. Створення та робота з подіями. Концепція “pub-sub” (publisher - subscriber). Огляд делегата EventHandler та базового класу EventArgs; робота з ними. Концептуальні особливості вибору доцільності використання подій.

Тема 10. Потoki на прикладі мови програмування C#. Багатопоточність. Основні концепції синхронізації потоків.

Поняття потоків та багатопоточності. Вивчення можливостей багатозадачного програмування в середовищі .NET. Використання делегатів ThreadStart та ParameterizedThreadStart. Критичні секції, як методи синхронізації доступу потоків до різноманітних роздільних ресурсів. Огляд, реалізація.

Тема 11. Методологія дебагу проекту. Оброблення виняткових ситуацій на прикладі мови програмування C#.

Введення в основи мануального дебагу проекту. Використання оператора try-catch. Повторне збудження виключної ситуації throw.

Тема 12. Шаблони проектування: користь та переваги шаблонів проектування, класифікація шаблонів проектування, недоліки шаблонів проектування.

Введення в поняття шаблони проектування. Історія шаблонів проектування. Огляд канонічних шаблонів проектування. Основні принципи вибору шаблонів проектування. Розгляд питання класифікації патернів в залежності від складності, охоплення та деталізації системи. Розбір питань корисності шаблонів проектування. Доцільність використання патернів. Можливий зворотній ефект від використання патернів.

6. Контроль навчальних досягнень

Навчальні досягнення студентів з дисципліни оцінюються за модульно-рейтинговою системою, в основу якої покладено принцип поопераційної звітності, обов'язковості модульного контролю, накопичувальної системи оцінювання рівня знань, умінь та навичок, розширення кількості підсумкових балів до 100.

Оцінка за кожний змістовий модуль включає бали за поточну роботу студента на практичних та лабораторних заняттях, за виконання індивідуальних завдань, за модульну контрольну роботу. Виконання модульних контрольних робіт здійснюється в електронному

вигляді. Модульний контроль знань студентів здійснюється після завершення вивчення навчального матеріалу змістового модуля.

У процесі оцінювання навчальних досягнень студентів застосовуються такі методи:

- *Методи усного контролю:* індивідуальне опитування, фронтальне опитування, співбесіда, екзамен.
- *Комп'ютерного контролю:* тестові програми.
- *Методи самоконтролю:* уміння самостійно оцінювати свої знання, самоаналіз.

Кількість балів за роботу з теоретичним матеріалом, на практичних заняттях, під час виконання самостійної роботи залежить від дотримання таких вимог:

- систематичність відвідування занять;
- своєчасність виконання навчальних і індивідуальних завдань;
- повний обсяг їх виконання;
- якість виконання навчальних і індивідуальних завдань;
- самостійність виконання;
- творчий підхід у виконанні завдань;
- ініціативність у навчальній діяльності;
- виконання тестових завдань.

Контроль успішності студентів з урахуванням поточного і підсумкового оцінювання здійснюється відповідно до навчально-методичної карти дисципліни, де зазначено види контролю і кількість балів за видами. Систему рейтингових балів для різних видів контролю та порядок їх переведення у національну (4-бальну) та європейську (ECTS) шкалу подано нижче у таблицях.

Розрахунок рейтингових балів за видами поточного (модульного) контролю

Вид діяльності студента	Максимальна к-сть балів за одиницю	Модуль 1		Модуль 2		Модуль 3	
		Кількість одиниць	Максимальна кількість балів	Кількість одиниць	Максимальна кількість балів	Кількість одиниць	Максимальна кількість балів
Відвідування лекцій	1	2	2	2	2	2	2
Відвідування практичних занять	1	2	2	2	2	3	3
Відвідування лабораторних занять	1	2	2	2	2	4	4
Робота на практичному занятті	10	2	20	2	20	3	30
Лабораторна робота (в тому числі допуск, виконання, захист)	10	2	20	2	20	4	40
Виконання завдань для самостійної роботи	5	1	5	1	5	1	5
Виконання модульної роботи	25	1	25	1	25	1	25
	Разом	-	76	-	76	-	109
Максимальна кількість балів: 261							
Розрахунок коефіцієнта: $261/100=2,61$							

Завдання для самостійної роботи та критерії її оцінювання

Самостійна робота є видом поза аудиторної індивідуальної діяльності студента, результати якої використовуються у процесі вивчення програмового матеріалу навчальної дисципліни та містить результати дослідницького пошуку, відображає певний рівень його навчальної компетентності.

Перелік тем та оцінювання самостійної роботи студента

№ з/п	Назва теми	Кількість годин	Бали
Змістовий модуль 1. Вступ до ООП		5	5
1	Провести порівняльний аналіз модифікаторів доступу в С#, а саме public, protected, internal, private, protected internal. Порівняння підкріпити прикладами	5	5
Змістовий модуль 2. Аспекти програмування на С#		6	5
2	Розкрити тему у власних прикладах: Що відбувається при виклику функції, визначеної шаблоном? Перевантаження шаблонів може призвести до складних і неоднозначних виборів функцій-кандидатів	6	5
Змістовий модуль 3. Розширені концепції С#		6	5
3	Шаблони проектування: •виконання завдань відповідно до теми; •опрацювання фахових видань.	6	5
Разом		17	15

Критерії оцінювання самостійної роботи студента

№ п/п	Критерії оцінювання роботи	Максимальна кількість балів за кожним критерієм
1	Критичний аналіз суті та змісту першоджерел. Виклад фактів, ідей, результатів досліджень в логічній послідовності. Аналіз сучасного стану дослідження проблеми, розгляд тенденцій подальшого розвитку даного питання.	2 бали
2	Доказовість висновків, обґрунтованість власної позиції, пропозиції щодо розв'язання проблеми, визначення перспектив дослідження	2 бали
3	Дотримання вимог щодо технічного оформлення	1 бал
Разом		5 балів

Форми проведення модульного контролю та критерії оцінювання

Модульний контроль здійснюється відповідно до навчально-методичної карти дисципліни та перевіряє рівень досягнення результатів навчання студентів. Форма проведення – комп'ютерний тест, що складається 15 запитань закритої та відкритої форм.

Модульна контрольна робота оцінюється у 25 балів.

Форми проведення семестрового контролю та критерії оцінювання

Семестрове (підсумкове) оцінювання здійснюється у формі заліку, умовою отримання якого є отриманням студентом 60 балів (з врахуванням коефіцієнту) за результатами поточного контролю.

Орієнтовний перелік питань для самоперевірки

1. Що таке клас та об'єкт? Що таке екземпляр класу?
2. Що таке властивість? Які види властивостей бувають?
3. Що таке модифікатори доступу та де їх використовують?
4. Які типи конструкторів ви знаєте? Навіщо використовувати конструктори за замовчуванням?
5. Перевантаження функцій - простий поліморфізм.
6. Поняття сигнатур. Операторні функції. Бінарні операції.
7. Що таке інтерфейс?
8. Чим абстрактний клас відрізняється від інтерфейсу?
9. Що таке множинне спадкування?
10. Що таке делегат?

11. Що таке подія?
12. Що таке узагальнення?
13. Поясніть поняття коваріантності та контраваріантності узагальнень.
14. Які переваги використання узагальнень?
15. Які недоліки використання узагальнень?
16. Що таке тип Nullable?
17. Що таке операція поглинання?
18. Які ви знаєте типи обмежень для узагальнення?
19. Що таке потік?
20. Який клас потрібно використовувати для створення екземпляра потоку?
21. Який делегат потрібно використовувати для передачі методу до потоку?
22. Яким чином можна передати параметри в потік?
23. Що таке виняток? Як створити виняток користувача?
24. Що таке конструкція try-catch та try-catch-finally?
25. У яких випадках не спрацьовує блок.
26. Типи паттернів.

Шкала відповідності оцінок

Рейтингова оцінка	Сума балів за всі види навчальної діяльності	Значення оцінки
A	90-100	Відмінно — відмінний рівень знань (умінь) в межах обов'язкового матеріалу з, можливими, незначними недоліками
B	82-89	Дуже добре - достатньо високий рівень знань (умінь) в межах обов'язкового матеріалу без суттєвих (грубих) помилок
C	75-81	Добре - в цілому добрий рівень знань (умінь) з незначною кількістю помилок
D	69-74	Задовільно - посередній рівень знань (умінь) із значною кількістю недоліків, достатній для подальшого навчання або професійної діяльності
E	60-68	Достатньо - мінімально можливий допустимий рівень знань (умінь)
FX	35-59	Незадовільно з можливістю повторного складання - незадовільний рівень знань, з можливістю повторного перескладання за умови належного самостійного доопрацювання
F	1-34	Незадовільно з обов'язковим повторним вивченням курсу - досить низький рівень знань (умінь), що вимагає повторного вивчення дисципліни

7. Навчально-методична картка дисципліни

Разом: 90 год., лекції – 12 год., практичні заняття – 14 год., лабораторні роботи – 16 год., модульний контроль – 6 год., самостійна робота – 17 год.

Модулі (назви, бали)	Змістовий модуль 1. Вступ до ООП (76 балів)		Змістовий модуль 2. Аспекти програмування на С# (76 балів)		Змістовий модуль 3. Розширені концепції С# (109 балів)	
Лекції (теми, бали)	№ 1-2 Концепція класів та об'єктів. Діаграми класів. Розробка класів та об'єктів мовою С#. Основні поняття об'єктно-орієнтованого програмування: інкапсуляція (1 бал)		№ 3-4 Основні поняття об'єктно-орієнтованого програмування: спадковість, просте та множинне успадкування. Основні поняття об'єктно-орієнтованого програмування: поліморфізм, абстракція (1 бал)		№ 5-6 Інтерфейси та їх реалізація на прикладі мови програмування С#. Концепція проектування через інтерфейси. Перевантаження функцій та перевантаження операцій на прикладі мови програмування С# (1 бал)	
Практичні, заняття (теми, бали)	№ 1 Робота з класами та об'єктами мовою С# (11 балів)		№ 2 Спадковість, просте та множинне успадкування (11 балів)		№ 3 Інтерфейси та їх реалізація (11 балів)	
Лабораторні заняття (теми, бали)	№ 1 Застосування принципу інкапсуляції (11 балів)		№ 2 Прикладне використання поліморфізму і абстракції (11 балів)		№ 3 Перевантаження функцій та операцій (11 балів)	
Самостійна робота	Самостійна робота (5 балів)		Самостійна робота (5 балів)		Самостійна робота (5 балів)	
Поточний контроль (вид, бали)	Модульна контрольна робота 1 (25 балів)		Модульна контрольна робота 2 (25 балів)		Модульна контрольна робота 3 (25 балів)	
Підсумковий контроль (вид, бали)	Залік					

8. Рекомендовані джерела

Основна (базова):

1. Настенко, Д. В. Об'єктно-орієнтоване програмування. Частина 1. Основи об'єктно-орієнтованого програмування на мові С#: навчальний посібник для бакалаврів напряму підготовки 6.050701 «Електротехніка та електротехнології» програми професійного спрямування «Системи управління виробництвом та розподілом електроенергії»– Київ : НТУУ «КПІ», 2016. – 76 с.
2. Дібрівний О.А., Гребенюк В.В. Вступ до об'єктно орієнтованого програмування С#: Навчальний посібник. – К.: Державний університет телекомунікацій, 2018, - 190с.
3. Asaad, Renas. (2020). Object Oriented Programming C-Sharp Programming C# provides full support for object-oriented programming including encapsulation, inheritance, and polymorphism.
4. Eric Freeman, Elisabeth Robson. Head First Design Patterns, 2nd Edition, December 2020, O'Reilly Media, Inc., ISBN: 9781492078005

Додаткова:

5. Simon Kendal. Object Oriented Programming using C#, 2011, Simon Kendal & Ventus Publishing ApS. ISBN 978-87-7681-814-2
6. Benjamin Perkins, Jon D. Reid. Beginning C# and .NET, 2021 Edition ISBN: 978-1-119-79578-0
7. Rishabh Verma, Neha Shrivastava, Ravindra Akella. Parallel Programming with C# and .NET Core: Developing Multithreaded Applications Using C# and .NET Core 3.1 from Scratch (English Edition). BPB Publications (June 26, 2020). ISBN-13: 978-9389423327

9. Додаткові ресурси (інформаційні ресурси)

1. Microsoft Visual Studio. URL: <https://visualstudio.microsoft.com/vs/>
2. Object-Oriented programming (C#). URL: <https://docs.microsoft.com/en-us/dotnet/csharp/fundamentals/tutorials/oop>
3. C# Tutorial. URL: https://www.w3schools.com/cs/cs_oop.php
4. C# Corner. URL: <https://www.c-sharpcorner.com/>