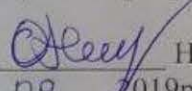


МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ АВТОМОБІЛЬНО-ДОРОЖНИЙ
УНІВЕРСИТЕТ

Кафедра комп'ютерних технологій і мехатроніки

«ЗАТВЕРДЖУЮ»

Гарант освітньо-професійної програми
«Програмне забезпечення систем»
першого (бакалаврського) рівня вищої
освіти, завідувач кафедри КТМ, д.т.н.,

професор  Ніконов О.Я.
« 03 » 09 2019р.

СИЛАБУС
ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ /
OBJECT-ORIENTED PROGRAMMING
SYLLABUS

освітній ступінь	бакалавр / bachelor
галузь знань	12 Інформаційні технології / Information Technology
спеціальність	121 Інженерія програмного забезпечення / Software Engineering
освітня програма	Програмне забезпечення систем / Systems Software

Харків 2019

Автор: Тімонін Володимир Олексійович, кандидат технічних наук,
старший науковий співробітник, доцент

Силабус розглянуто та затверджено на засіданні кафедри комп'ютерних
технологій і мехатроніки, протокол № 18 від «27» червня 2019 р.

СИЛАБУС

ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ /

OBJECT-ORIENTED PROGRAMMING

SYLLABUS

освітній ступінь	бакалавр / bachelor
галузь знань	12 Інформаційні технології / Information Technology
спеціальність	121 Інженерія програмного забезпечення / Software Engineering
освітня програма	Програмне забезпечення систем / Systems Software

ВСТУП

На сучасному етапі розвитку інформаційних технологій об'єктно-орієнтоване програмування (ООП) займає істотну частину всіх доступних засобів розробки, аналізу, проектування і моделювання.

У методології ООП за п'ятдесят років існування сформовані знання і накопичений досвід по якійсній розробки об'єктно-орієнтованого програмного забезпечення. В даний час методологія ООП стає однією з центральних в підготовці майбутніх фахівців.

Особлива роль об'єктно-орієнтованих технологій призводить до необхідності детального вивчення принципів побудови програмних компонент інформаційних систем на базі об'єктних технологій.

Сучасні технології ООП інтенсивно розвиваються - на даний момент програмісту недостатньо розуміти найпростіші принципи ООП. ООП як технологія має реагувати на появу нових вимог сучасного високотехнологічного світу: паралельний характер процесів в інформаційних системах; розподілений характер інформаційних систем; злиття різних технологій розробки додатків і ін.

Програма вивчення навчальної дисципліни **“Об'єктно-орієнтоване програмування”** складена відповідно до освітньо-кваліфікаційної характеристики та навчального плану підготовки **бакалавра спеціальності 121 «Інженерія програмного забезпечення»**.

Викладання дисципліни розраховано на виконання студентами практичних робіт та самостійних завдань у програмному середовищі MS Visual Studio.

1. МЕТА, ПРЕДМЕТ ТА ЗАВДАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

1.1. **Метою** вивчення навчальної дисципліни є вивчення та практичне освоєння методів і засобів об'єктно-орієнтованого програмування як однієї з основних і перспективних моделей програмування, що є в даний час базою для створення програмних систем, придбання базових знань і навичок програмування, проектування і розробки додатків із застосуванням об'єктно-орієнтованого підходу, теоретична і практична підготовка фахівців у галузі сучасних технологій програмування на рівні професійних вимог за напрямом.

1.2. **Предметом** вивчення навчальної дисципліни є концепція об'єктно-орієнтованого програмування, принципи об'єктно-орієнтованого програмування, методи і засоби її реалізації в середовищі візуального програмування, а також методи їх використання при розробленні додатків.

1.3. **Основними завданнями** вивчення навчальної дисципліни є формування у студентів необхідних знань у галузі сучасних технологій програмування; ознайомлення з технічними, алгоритмічними, програмними і технологічними рішеннями у області програмування; вироблення практичних навичок аналітичного та експериментального дослідження основних методів і засобів, що використовуються в області програмування при рішенні обчислювальних задач.

1.4. По завершенні вивчення дисципліни студенти повинні:

знати:

поняття об'єктно-орієнтованого аналізу, проектування та програмування; основи об'єктно-орієнтованої мови програмування; принципи структурного та процедурного програмування; принципи об'єктно-орієнтованого програмування; особливості розробки графічних інтерфейсів користувача; особливості програмування, керованого подіями; особливості оброблення виняткових ситуацій; особливості обробки подій від миші, клавіатури, команд меню, елементів управління тощо; правила використання бібліотек класів середовищ розробника програм.

вміти:

розробляти програми на алгоритмічній мові високого рівня для обробки різних типів даних; використовувати бібліотеки класів реалізації функціональних можливостей Windows; використовувати особливості алгоритмічній мови високого рівня для рішення професійних задач; проектувати компоненти програмного забезпечення; проектувати людино-машинний інтерфейс інформаційних систем; реалізувати прототипи архітектури програмного забезпечення; установлювати, налаштовувати та обслуговувати системне, інструментальне і прикладне програмне забезпечення та інформаційні системи.

1.5. Міждисциплінарні зв'язки:

Вивченню дисципліни

передують: “Основи інформатики” середньої школи, “Вища математика”, “Дискретна математика”, “Теорія алгоритмів”, “Алгоритмізація та програмування”;

потребують її вивчення: “Організація баз даних і знань”, “Крос-платформне програмування”, “Веб-технології та веб-дизайн”, “Технологія створення програмних продуктів”, “Технологія розподілених систем та паралельних обчислень”, “Методи та системи штучного інтелекту”, “Проектно-технологічна практика”, “Дипломне проектування”.

2. ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Найменування показників	Характеристика навчальної дисципліни
Кількість кредитів - 8 Кількість годин - 240	<u>нормативна</u> (нормативна, за вибором ВНЗ, за вибором студента)
Семестр викладання дисципліни	3
Вид контролю:	<u>екзамен</u> (залік, екзамен)
Розподіл часу:	
- лекції (годин)	16
- практичні, семінарські (годин)	-
- лабораторні роботи (годин)	64
- самостійна робота студентів (годин)	100
- курсовий проект (годин)	-
- курсова робота (годин)	30

розрахунково-графічна робота (контрольна робота)	-
підготовка та складання екзамену	30

3. РОЗПОДІЛ ДИСЦИПЛІНИ У ГОДИНАХ ЗА ФОРМАМИ ОРГАНІЗАЦІЇ ОСВІТНЬОГО ПРОЦЕСУ ТА ВИДАМИ НАВЧАЛЬНИХ ЗАНЯТЬ

Результати навчання	Навчальна діяльність	Робочий час студента (год.)	Оціню вання (бал)
1	2	3	4
Тема 1. Поняття об'єктно-орієнтованого аналізу, проектування та програмування. Основи об'єктно-орієнтованої мови програмування.			
<i>Знати поняття об'єктно-орієнтованого аналізу, проектування та програмування; підходи до аналізу об'єктно-орієнтованих систем; поняття абстракції даних, інкапсуляції, поліморфізму, успадкування.</i>	Лекція №1. Поняття об'єктно-орієнтованого аналізу, проектування та програмування. Основи об'єктно-орієнтованої мови програмування. <i>Основні питання:</i> 1. Технологія об'єктно-орієнтованого програмування. 2. Основні поняття об'єктно-орієнтованого аналізу. 3. Основні поняття об'єктно-орієнтованого проектування. 4. Принципи об'єктно-орієнтованого програмування. <i>Список рекомендованих джерел:</i> Основний – 1, 6 Додатковий – 11	2	
	Самостійна робота студента (СРС ¹). Поняття об'єктно-орієнтованого аналізу, проектування та програмування. <i>Основні питання:</i> 1. Історія появи і розвитку ООП. 2. Підходи до аналізу об'єктно-орієнтованих систем.	4	
	СРС. Об'єктна модель предметного середовища, принципи і побудови.	4	

¹ Завданнями для самостійної роботи є вивчення та доповнення матеріалів лекції на основі самостійного опрацювання основних літературних джерел, зазначених у списку та електронних матеріалів, які надаються викладачем, а також в практичному відпрацюванні незавершених завдань в практичних роботах.

	<i>Основні питання:</i> 1. Принципи об'єктно-орієнтованого програмування. 2. Поняття абстракції даних, інкапсуляції, поліморфізму, успадкування. 3. Основні поняття ООП.		
Тема 2. Введення в класи, об'єкти, методи, властивості.			
Знати принципи об'єктно-орієнтоване програмування; правила і особливості розробки класів і його елементів. Вміти проектувати компоненти програмного забезпечення; розробляти класи і його елементи; використовувати об'єкти для рішення професійних задач.	Лекція №2. Введення в класи, об'єкти, методи, властивості. <i>Основні питання:</i> 1. Поняття класів та об'єктів. 2. Методи класів 3. Властивості класів. <i>Список рекомендованих джерел:</i> Основний – 1, 6, 8, 9 Додатковий – 11 Інтернет-ресурси – 14	2	
	СРС. Поняття об'єктів і класів та їх взаємовідносин. <i>Основні питання:</i> 1. Діаграми об'єктів. 2. Відносини між об'єктами (успадкування, реалізація, асоціація, композиція і агрегація).	4	
	СРС. Розробка класів. Створення об'єктів. <i>Основні питання:</i> 1. Модифікатори доступу. 2. Конструктори класу. 3. Деструктор класу. 4. Використання ключового слова this.	8	
	Практична робота №1 (ПР1). Дослідження можливостей інтегрованого середовища розробки Visual C# для створення додатків, що використовують класи та об'єкти. <i>Мета роботи - досліджувати можливості інтегрованого середовища розробки Visual Studio 2017 і отримати практичні навички зі створення додатків, що використовують класи і об'єкти.</i> План заняття:	4	0...100

	1. Актуалізація теоретичного матеріалу. 2. Виконання завдань практичної роботи (розробка алгоритму, написання додатків, обробка помилок, тестування додатків). 3. Презентація і захист виконаної роботи. 4. Тестування знань.		
	СРС. Методи і модифікатори параметрів. <i>Основні питання:</i> 1. Модифікатор out. 2. Модифікатор ref. 3. Модифікатор params. 4. Визначення необов'язкових параметрів.	8	
	ПР2. Дослідження можливостей інтегрованого середовища розробки Visual C# для створення додатків, що використовують методи. <i>Мета роботи - досліджувати можливості інтегрованого середовища розробки Visual Studio 2017 і отримати практичні навички зі створення додатків, що використовують методи.</i> План заняття: 1. Актуалізація теоретичного матеріалу. 2. Виконання завдань практичної роботи (розробка алгоритму, написання додатків, обробка помилок, тестування додатків). 3. Презентація і захист виконаної роботи. 4. Тестування знань.	4	0...100
	ПР3. Дослідження можливостей інтегрованого середовища розробки Visual C # для створення додатків з класами. Властивості класу. <i>Мета роботи - досліджувати можливості інтегрованого середовища розробки Visual Studio 2017 і отримати практичні навички зі створення додатків, що використовують</i>	4	0...100

	<i>властивості класу.</i> План заняття: 1. Актуалізація теоретичного матеріалу. 2. Виконання завдань практичної роботи (розробка алгоритму, написання додатків, обробка помилок, тестування додатків). 3. Презентація і захист виконаної роботи. 4. Тестування знань.		
	СРС. Статичні, константні члени класів, дружні функції та класи. <i>Основні питання:</i> 1. Визначення статичних полів даних. 2. Визначення статичних методів. 3. Визначення статичних класів.	8	
Тема 3. Перевантаження операцій та функцій.			
Знати принципи об'єктно-орієнтоване програмування; механізм і особливості перевантаження операторів. Вміти розробляти і застосовувати перевантажені оператори; проектувати компоненти програмного забезпечення.	Лекція №3. Перевантаження операцій та функцій. <i>Основні питання:</i> 1. Синтаксис перевантажених операторів. 2. Перевантаження бінарних операторів. 3. Перевантаження унарних операторів. 4. Виконання операцій над значеннями вбудованих C#-типів. <i>Список рекомендованих джерел:</i> Основний – 1, 6, 8, 9 Додатковий – 11 Інтернет-ресурси – 14	2	
	СРС. Перевантаження операцій та функцій. <i>Основні питання:</i> 1. Перевантаження операторів відносини. 2. Перевантаження операторів true і false. 3. Перевантаження логічних операторів. 4. Оператори перетворення.	8	

	ПР4. Дослідження можливостей інтегрованого середовища розробки Visual C# для створення додатків з класами. Перевантаження операторів. <i>Мета роботи - досліджувати можливості інтегрованого середовища розробки Visual Studio 2017 і отримати практичні навички зі створення додатків, що використовують перевантаження операторів в класах.</i> План заняття: 1. Актуалізація теоретичного матеріалу. 2. Виконання завдань практичної роботи (розробка алгоритму, написання додатків, обробка помилок, тестування додатків). 3. Презентація і захист виконаної роботи. 4. Тестування знань.	4	0...100
Тема 4. Успадкування.			
Знати принципи об'єктно-орієнтоване програмування; правила успадкування об'єктів; призначення і використання віртуальних методів; призначення делегатів; правила і механізм створення і використання подій об'єкта. Вміти розробляти і застосовувати	Лекція №4. Успадкування. <i>Основні питання:</i> 1. Основи спадкування. 2. Доступ до членів базового класу при спадкуванні. 3. Конструктори при спадкуванні. 4. Спадкування і приховування імен. <i>Список рекомендованих джерел:</i> Основний – 1, 6, 8, 9 Додатковий – 11 Інтернет-ресурси – 14	2	
	СРС. Поняття спадкування та поліморфізму. <i>Основні питання:</i> 1. Використання властивостей класу при спадкуванні. 2. Використання захищених членів класу. 3. Застосування конструктора довільного класу. 4. Застосування конструктора базового класу.	8	

<i>похідні класи; розробляти і застосовувати події класів; проектувати компоненти програмного забезпечення.</i>	СРС. Віртуальні методи. <i>Основні питання:</i> 1. Оголошення віртуальних методів. 2. Застосування віртуальних методів. 3. Застосування абстрактних класів.	4	
	ПР5. Дослідження можливостей інтегрованого середовища розробки Visual C # для створення додатків з похідними класами. <i>Мета роботи - досліджувати можливості інтегрованого середовища розробки Visual Studio 2017 і отримати практичні навички зі створення додатків з похідними класами.</i> План заняття: 1. Актуалізація теоретичного матеріалу. 2. Виконання завдань практичної роботи (розробка алгоритму, написання додатків, обробка помилок, тестування додатків). 3. Презентація і захист виконаної роботи. 4. Тестування знань.	4	0...100
	СРС. Делегати і події <i>Основні питання:</i> 1. Поняття делегата. 2. Багатоадресне передача. 3. Створення події. 4. Оформлення події. 5. Підписка події.	8	
	ПР6. Дослідження можливостей інтегрованого середовища розробки Visual C# для створення додатків, що використовують події. <i>Мета роботи - досліджувати можливості інтегрованого середовища розробки Visual Studio 2017 і отримати практичні навички зі створення додатків, що використовують події.</i> План заняття: 1. Актуалізація теоретичного матеріалу. 2. Виконання завдань практичної	4	0...100

	роботи (розробка алгоритму, написання додатків, обробка помилок, тестування додатків). 3. Презентація і захист виконаної роботи. 4. Тестування знань.		
	СРС. Інтерфейси. Реалізація інтерфейсів. <i>Основні питання:</i> 1. Поняття інтерфейсу. 2. Реалізація інтерфейсу. 3. Інтерфейсні властивості. 4. Спадкування інтерфейсів.	8	
Тема 5. Бібліотеки класів реалізації функціональних можливостей Windows.			
Знати правила використання бібліотек класів середовищ розробника програм; особливості розробки графічних інтерфейсів користувача; основи візуального проектування; основні принципи програмування в середовищі Windows. Вміти використовувати бібліотеки класів реалізації функціональних можливостей Windows; розробляти графічний інтерфейс користувача;	Лекція №5. Бібліотеки класів реалізації функціональних можливостей Windows. <i>Основні питання:</i> 1. Поняття загальної бібліотеки в NET. 2. Специфічні для платформи бібліотеки класів. 3. Переносяться бібліотеки класів. 4. Бібліотеки класів .NET Standard. <i>Список рекомендованих джерел:</i> Основний – 1, 2, 4, 5, 6, 8, 9 Додатковий – 13 Інтернет-ресурси – 14, 15	2	
	СРС. Стандартні бібліотеки класів середовищ розробника програм. <i>Основні питання:</i> 1. Підтримка реалізації .NET. 2. Специфікація .NET Стандарт. 3. Бібліотеки .NET Standard і Visual Studio 2017. 4. Інтегроване середовище розробки Visual Studio 2017.	8	
	ПР7. Дослідження можливостей ICP Visual C# для створення додатків, що використовують прості компоненти введення і відображення текстової інформації. <i>Мета роботи - досліджувати</i>	4	0...100

<i>застосовувати стандартні компоненти при розробці додатків, що обробляють текстові дані.</i>	<i>можливостей інтегрованого середовища розробки Visual C # і отримати практичні навички зі створення додатків, що використовують прості компоненти введення і відображення інформації.</i> План заняття: 1. Актуалізація теоретичного матеріалу. 2. Виконання завдань практичної роботи (розробка алгоритму, написання додатків, обробка помилок, тестування додатків). 3. Презентація і захист виконаної роботи. 4. Тестування знань.		
	ПР8. Дослідження можливостей ICP Visual C # для створення додатків по обробці списків даних. <i>Мета роботи - досліджувати можливостей інтегрованого середовища розробки Visual C # і отримати практичні навички зі створення додатків по обробці списків даних.</i> План заняття: 1. Актуалізація теоретичного матеріалу. 2. Виконання завдань практичної роботи (розробка алгоритму, написання додатків, обробка помилок, тестування додатків). 3. Презентація і захист виконаної роботи. 4. Тестування знань.	4	0...100
Тема 6. Розробка графічних інтерфейсів користувача (меню, панелі інструментів, шаблони діалогових вікон тощо).			
Знати структуру проекту; особливості розробки графічних інтерфейсів користувача; особливості	Лекція №6. Розробка графічних інтерфейсів користувача (меню, панелі інструментів, шаблони діалогових вікон тощо). <i>Основні питання:</i> 1. Структура проекту Visual Studio 2017. 2. Форма додатка, її властивості,	2	

програмування, керованого подіями; особливості обробки подій від команд меню, елементів управління тощо. Вміти проектувати людино-машинний інтерфейс інформаційних систем; використовувати бібліотеки класів реалізації функціональних можливостей Windows; застосовувати стандартні компоненти при розробці додатків з елементами управління, меню, панелями інструментів.	методи і події. 3. Компоненти .NET. Список рекомендованих джерел: Основний – 1, 2, 4, 5, 6, 8, 9 Додатковий – 13 Інтернет-ресурси – 14, 15		
	ПР9. Дослідження можливостей ІСР Visual C # для створення додатків з використанням керуючих елементів. Мета роботи - досліджувати можливостей інтегрованого середовища розробки Visual C # і отримати практичні навички зі створення додатків з використанням керуючих елементів. План заняття: 1. Актуалізація теоретичного матеріалу. 2. Виконання завдань практичної роботи (розробка алгоритму, написання додатків, обробка помилок, тестування додатків). 3. Презентація і захист виконаної роботи. 4. Тестування знань.	4	0...100
	ПР10. Дослідження можливостей ІСР Visual C # для створення додатків з використанням меню. Мета роботи - досліджувати можливостей інтегрованого середовища розробки Visual C # і отримати практичні навички зі створення додатків з використанням меню. План заняття: 1. Актуалізація теоретичного матеріалу. 2. Виконання завдань практичної роботи (розробка алгоритму, написання додатків, обробка помилок, тестування додатків). 3. Презентація і захист виконаної роботи. 4. Тестування знань.	4	0...100
	ПР11. Дослідження можливостей ІСР	4	0...100

	Visual C # для створення додатків з використанням панелей інструментів. <i>Мета роботи - досліджувати можливостей інтегрованого середовища розробки Visual C # і отримати практичні навички зі створення додатків з використанням панелей інструментів.</i> План заняття: 1. Актуалізація теоретичного матеріалу. 2. Виконання завдань практичної роботи (розробка алгоритму, написання додатків, обробка помилок, тестування додатків). 3. Презентація і захист виконаної роботи. 4. Тестування знань.		
	ПР12. Дослідження можливостей ICP Visual C # для створення додатків з декількома формами. <i>Мета роботи - досліджувати можливостей інтегрованого середовища розробки Visual C # і отримати практичні навички зі створення додатків з декількома формами.</i> План заняття: 1. Актуалізація теоретичного матеріалу. 2. Виконання завдань практичної роботи (розробка алгоритму, написання додатків, обробка помилок, тестування додатків). 3. Презентація і захист виконаної роботи. 4. Тестування знань.	4	0...100
Тема 7. Основи програмування динамічних процесів.			
Знати поняття імітаційного моделювання; компоненти для розробки додатків, що відображають	Лекція №7. Основи програмування динамічних процесів. <i>Основні питання:</i> 1. Поняття імітаційного моделювання. 2. Переваги та недоліки імітаційного моделювання.	2	

динамічні процеси; методи роботи з графікою; основи обробки зображень. Вміти проектувати людино-машинний інтерфейс додатків; застосовувати стандартні компоненти при розробці додатків, що відображають динамічні процеси; створювати додатки, що використовують графічні примітиви і зображення.	3. Компоненти для розробки додатків, що відображають динамічні процеси. <i>Список рекомендованих джерел:</i> Основний – 1, 2, 4, 5, 6, 8, 9 Додатковий – 13 Інтернет-ресурси – 14, 15		
	СРС. Побудова графічних зображень. <i>Основні питання:</i> 1. Поняття про графічної поверхні. 2. Інструменти малювання. 3. Графічні примітиви. 4. Відображення текстової інформації.	4	
	ПР13. Дослідження можливостей ICP Visual C# для створення додатків побудови графічних зображень. <i>Мета роботи - досліджувати можливості інтегрованого середовища розробки Visual C # і отримати практичні навички зі створення додатків по роботі з графікою.</i> План заняття: 1. Актуалізація теоретичного матеріалу. 2. Виконання завдань практичної роботи (розробка алгоритму, написання додатків, обробка помилок, тестування додатків). 3. Презентація і захист виконаної роботи. 4. Тестування знань.	4	0...100
	СРС. Робота з зображеннями <i>Основні питання:</i> Клас Bitmap і його можливості. Малювання існуючого точкового малюнка на екрані. Обрізка і масштабування зображень в GDI +. Поворот, відображення і нахил зображень.	8	
	ПР14. Дослідження можливостей Visual C # для створення додатків, що відображають динамічні процеси. <i>Мета роботи - досліджувати</i>	4	0...100

	<i>можливості інтегрованого середовища розробки Visual C # і отримати практичні навички зі створення додатків, що відображають динамічні процеси.</i> План заняття: 1. Актуалізація теоретичного матеріалу. 2. Виконання завдань практичної роботи (розробка алгоритму, написання додатків, обробка помилок, тестування додатків). 3. Презентація і захист виконаної роботи. 4. Тестування знань.		
Тема8. Обробники подій від миші, клавіатури, елементів управління тощо.			
Знати <i>особливості обробки подій від миші та клавіатури.</i> Вміти <i>використовувати бібліотеки класів реалізації функціональних можливостей Windows; створювати додатки, що використовують події миші і клавіатури.</i>	Лекція №8. Обробники подій від миші, клавіатури, елементів управління тощо. <i>Основні питання:</i> 1. Події миші. 2. Розпізнавання джерела події, натиснутих кнопок, координат курсору. 3. Події клавіатури. 4. Розпізнавання натиснутих клавіш. <i>Список рекомендованих джерел:</i> Основний – 1, 2, 4, 5, 6, 8, 9 Додатковий – 13 Інтернет-ресурси – 14, 15	2	
	СРС. Обробники подій від подій миші. <i>Основні питання:</i> 1. Обробники події миші. 2. Властивості об'єкта MouseEventArgs.	4	
	ПР15. Дослідження можливостей ICP Visual C # для створення додатків по обробці подій миші. <i>Мета роботи - досліджувати можливостей інтегрованого середовища розробки Visual C # і отримати практичні навички зі створення додатків по обробці подій миші.</i>	4	0...100

	План заняття: 1. Актуалізація теоретичного матеріалу. 2. Виконання завдань практичної роботи (розробка алгоритму, написання додатків, обробка помилок, тестування додатків). 3. Презентація і захист виконаної роботи. 4. Тестування знань.		
	CPC. Обробники подій від подій клавіатури. <i>Основні питання:</i> 1. Обробники події миші. 2. Властивості об'єкта KeyEventArgs. 3. Властивості об'єкта KeyPressEventArgs.	4	
	ПР16. Дослідження можливостей ICP Visual C # для створення додатків по обробці подій клавіатури. <i>Мета роботи - досліджувати можливостей інтегрованого середовища розробки Visual C # і отримати практичні навички зі створення додатків по обробці подій клавіатури.</i> План заняття: 1. Актуалізація теоретичного матеріалу. 2. Виконання завдань практичної роботи (розробка алгоритму, написання додатків, обробка помилок, тестування додатків). 3. Презентація і захист виконаної роботи. 4. Тестування знань.	4	0...100
Курсова робота «Підготовка і рішення прикладної задачі на ПК»			
	Тема «Обробка результатів вимірювань, отриманих в результаті проведення N експериментів» (вхідні дані і перелік завдань визначаються відповідно заданому варіанту).	30	
Разом		210	0...100

5. КОНТРОЛЬ ТА ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ

Під час вивчення дисципліни «Об'єктно-орієнтоване програмування» викладачем здійснюється поточний та підсумковий контроль з використанням тестових завдань. Поточний контроль та оцінювання передбачає:

- перевірку рівня засвоєння теоретичного матеріалу (тестування за матеріалами лекції);
- захист практичних робіт (перевірка розроблених проектів та тестування за матеріалами практичних робіт);
- перевірка засвоєння матеріалу, що винесений на самостійне опрацювання під час фронтального опитування на лекції.

Тестування знань студентів здійснюється під час проведення практичних занять за допомогою автоматизованої системи знань. Підсумкова оцінка (ПО) знань з навчальної дисципліни визначається як середньозважена результатів засвоєння окремих блоків в інтервалі (1...100) балів:

$$ПО = \frac{\sum_{i=1}^I K_i}{N},$$

де K – кількість балів за кожен блок,

I – кількість зданих блоків,

N – загальна кількість блоків за дисципліну.

У разі незгоди студента з підсумковою оцінкою, підсумковий контроль знань та компетентностей студента з навчальної дисципліни здійснюється на підставі екзамену.

**Відповідність підсумкових семестрових рейтингових оцінок у балах
оцінкам за національною шкалою та шкалою ECTS**

Оцінка в балах	Оцінка за національною шкалою	Оцінка за шкалою ECTS	
		Оцінка	Критерії
90-100	Відмінно	A	“Відмінно” - теоретичний зміст курсу освоєний цілком, без прогалин, необхідні практичні навички роботи з освоєним матеріалом сформовані, усі передбачені програмою навчання навчальні завдання виконані, якість їхнього виконання оцінено числом балів, близьким до максимального.
82 – 89	Добре	B	“Дуже добре” - теоретичний зміст курсу освоєний цілком, без прогалин, необхідні практичні навички роботи з освоєним матеріалом в основному сформовані, усі передбачені програмою навчання навчальні завдання виконані, якість виконання більшості з них оцінено числом балів, близьким до максимального.
75 – 81		C	“Добре” - теоретичний зміст курсу освоєний цілком, без прогалин, деякі практичні навички роботи з освоєним матеріалом сформовані недостатньо, усі передбачені програмою навчання навчальні завдання виконані, якість виконання жодного з них не оцінено мінімальним числом балів, деякі види завдань виконані з помилкам
67 – 74	Задовільно	D	“Задовільно” - теоретичний зміст курсу освоєний частково, але прогалини не носять істотного характеру, необхідні практичні навички роботи з освоєним матеріалом в основному сформовані, більшість передбачених програмою навчання навчальних завдань виконано, деякі з виконаних завдань, можливо, містять помилки.
60 – 66		E	“Достатньо” - теоретичний зміст курсу освоєний частково, деякі практичні навички роботи не сформовані, багато передбачені програмою навчання навчальні завдання не виконані, або якість виконання деяких з них оцінено числом балів, близьким до мінімального.
35 – 59	Незадовільно	FX	“Незадовільно” - теоретичний зміст курсу освоєний частково, необхідні практичні навички роботи не сформовані, більшість передбачених програм навчання навчальних завдань не виконано, або якість їхнього виконання оцінено числом балів, близьким до мінімального; при додатковій самостійній роботі над матеріалом курсу можливе підвищення якості виконання навчальних завдань(з можливістю повторного складання)
1 – 34		F	“Неприйнятно” - теоретичний зміст курсу не освоєно, необхідні практичні навички роботи не сформовані, усі виконані навчальні завдання містять грубі помилки, додаткова самостійна робота над матеріалом курсу не приведе до якого-небудь значимого підвищення якості виконання навчальних завдань.(з обов'язковим повторним курсом)

6. ЧАС ТА МІСЦЕ ПРОВЕДЕННЯ:

- аудиторні заняття – відповідно до розкладу ХНАДУ, ауд. 214, 216;
- позааудиторна робота – самостійна робота студента, результат виконання якої висвітлено засобами Visual Studio.

7. СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ

Основний

1. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений. Пер. с англ. - М.: ООО «Вильямс», 2008. – 720 с.: ил.
2. Ватсон Б. С# 4.0 на примерах. - СПб.: БХВ-Петербург, 2011. – 608с.: ил.
3. Климов А.П. С#. Советы программистам. - СПб.: БХВ-Петербург, 2008. – 544с.: ил.
4. Петцольд Ч. Программирование для Microsoft Windows на С#. В 2-х томах. Том 1. Пер. с англ. - М.: «Русская Редакция», 2002.- 576 с.: ил.
5. Петцольд Ч. Программирование для Microsoft Windows на С#. В 2-х томах. Том 2. Пер. с англ. - М.: «Русская Редакция», 2002.- 624 с.: ил.
6. Тімонін В.О. Конспект лекцій з дисципліни «Об'єктно-орієнтоване програмування» для студентів за спеціальністю 121 «Інженерія програмного забезпечення». – Харків: ХНАДУ, 2018. – 128 с. (в електронній формі).
7. Тімонін В.О. Методичні вказівки до проведення практичних робіт з дисципліни «Об'єктно-орієнтоване програмування» для студентів за спеціальністю 121 «Інженерія програмного забезпечення». – Харків: ХНАДУ, 2018. – 84 с.
8. Троелсен Э. Язык программирования С# 2010 и платформа .NET 4.0. Пер. с англ. - М.: Издательский дом "Вильямс", 2011. — 1392 с.: ил.
9. Шилдт Г. С# 4.0: полное руководство. – М.: ООО «Вильямс», 2012. – 1056 с.: ил.

Додатковий

10. Берковський В.В., Левтеров А.І., Костикова М.В., Онуфрей Ю.Є., Подоляка О.О., Попеленко А.А. Програмування в середовищі С(С++). Збірник задач. – Харків: ХНАДУ, 2006. – 224 с.
11. Лафоре Р. Объектно-ориентированное программирование в С++. Класика Computer Science. 4-е изд. – СПб.: Питер, 2008. – 928 с.: ил.
12. Коноваленко І.В. Програмування мовою С# 6.0. – Тернопіль, ТНТУ, 2016. – 227 с.: ил.
13. Петцольд Ч. Программирование для Microsoft Windows 8. Разработка для Windows Store на С# - М.: «Русская Редакция», 2014.- 1008 с.: ил.

Інтернет-ресурси

14. <https://docs.microsoft.com>
15. <https://visualstudio.microsoft.com>