

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Кафедра комп'ютерних наук

«ЗАТВЕРДЖУЮ»
Декан факультету інформаційних технологій
Олена ГЛАЗУНОВА
«12» 2023 р.

«СХВАЛЕНО»
на засіданні кафедри комп'ютерних наук
Протокол № 12 від «01» 06 2023
р.

Завідувач кафедри
Белла ГОЛУБ

«РОЗГЛЯНУТО»
Гарант ОП «Інженерія програмного забезпечення»
Гарант ОП
Белла ГОЛУБ

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ

Спеціальність 121 – «Інженерія програмного забезпечення»

Освітня програма «Інженерія програмного забезпечення»

Факультет інформаційних технологій

Розробник: ст.викл. Міловідов Ю. О.

Київ – 2023 р.

Опис навчальної дисципліни

ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ

Галузь знань, спеціальність, освітня програма, освітній ступінь		
Освітній ступінь	Бакалавр	
Спеціальність	121 – «Інженерія програмного забезпечення»	
Освітня програма	Інженерія програмного забезпечення	
Характеристика навчальної дисципліни		
Вид	обов'язкова	
Семестр	3	4
Загальна кількість годин	90	90
Кількість кредитів ECTS	3	3
Кількість змістових модулів	2	2
Курсова робота	30 год. 1 кредит	
Форма контролю	Диф. залік	іспит
Показники навчальної дисципліни для денної та заочної форм навчання		
	денна форма навчання	заочна форма навчання
Курс (рік підготовки)	2	2
Лекційні заняття	60	
Лабораторні заняття	60	
Самостійна робота	60	
Кількість тижневих аудиторних годин для денної форми навчання	4	

1. Мета, завдання та компетентності навчальної дисципліни

Мета: отримання студентами знань в області розробки програм зі застосуванням об'єктно-орієнтованих технологій. Оволодіння такими знаннями дозволить реалізовувати задачі автоматизації обробки інформації, автоматизації керування об'єктами, в тому числі, сільськогосподарськими, за допомогою комп'ютерної техніки.

Завдання: використання і застосування об'єктно-орієнтованих технологій програмування як при подальшому навчанні, так і після отримання вищої освіти у своїй професійній діяльності.

Вивчення дисципліни «Об'єктно-орієнтоване програмування» сприяє формуванню у студентів *наступних компетентностей*.

Інтегральні компетентності:

Здатність розв'язувати складні задачі і проблеми інженерії програмного забезпечення, що передбачає проведення досліджень з елементами наукової новизни та/або здійснення інновацій в умовах невизначеності вимог.

Загальні компетентності:

K05. Здатність вчитися і оволодівати сучасними знаннями.

Фахові компетентності:

K14. Здатність брати участь у проєктуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.

K15. Здатність розробляти архітектури, модулі та компоненти програмних систем.

K23. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.

K24. Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення.

K26. Здатність до алгоритмічного та логічного мислення.

Програмні результати:

ПР05. Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення.

ПР06. Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення, враховуючи специфіку природоохоронної галузі.

ПР07. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.

ПР12. Застосовувати на практиці ефективні підходи щодо проєктування

програмного забезпечення.

ПР13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.

2. Програма і структура навчальної дисципліни для:

- повного терміну денної (заочної) форми навчання;
- скороченого терміну денної (заочної) форми навчання.

Назви змістових модулів і тем	Кількість годин												
	денна форма							Заочна форма					
	тижні	усього	у тому числі					усього	у тому числі				
			л	п	лаб	інд	с.р.		л	п	лаб	інд	с.р.
1	2	3	4	5	6	7	8	9	10	11	12	13	14
1 семестр													
Змістовий модуль 1. Класи та абстракція даних													
Тема 1. Введення в ООП	4	23	8		8		7						
Тема 2. Основні принципи реалізації класів	3	20	6		6		8						
Разом за змістовим модулем 1	7	43	14		14		15						
Змістовий модуль 2. Реалізація властивостей класу													
Тема 3. Перевантаження операцій	2	12	4		4		4						
Тема 4. Спадкування	3	18	6		6		6						
Тема 5. Віртуальні функції і поліморфізм	3	17	6		6		5						
Разом за змістовим модулем 2	8	47	16		16		15						
Разом за 1 сем.	15	90	30		30		30						
2 семестр													
Змістовий модуль 3. Об'єктно-орієнтована технологія програмування на мові C#													
Тема 6. Базові поняття мови C#.	3	19	6		6		7						
Тема 7. Класи, інтерфейси, наслідування, поліморфізм.	5	28	10		10		8						
Разом за змістовим модулем 3	8	47	16		16		15						
Змістовий модуль 4. Додаткові можливості ООП у C#													

Тема 8. Делегати, події, лямбда-вирази.	4	23	8		8		7						
Тема 9. Рефлексія. Серіалізація, десеріалізація	3	20	6		6		8						
Разом за змістовим модулем 4	7	43	14		14		15						
Разом за 2 сем.	15	90	30		30		30						
Усього годин	30	180	60		60		60						
Курсова робота						30							

3. Теми семінарських занять

Не передбачені

4. Теми практичних занять

Не передбачені

5. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1	Класи і об'єкти в C++. Створення простих конструкторів для ініціалізації елементів-змінних класів та виділення пам'яті. Використання конструкторів з аргументами за замовчуванням. Розробка програм з використанням власних класів.	4
2	Дружні функції C++. Доступ до закритих членів класу функціями, які не є членами цього класу.	4
3	Використання основних принципів перевантаження операцій.	4
4	Робота з потоками читання/запису файлів. Обробка масивів об'єктів.	4
5	Використання конструктора копіювання.	4
6	Наслідування. Створення ієрархії класів.	4
7	Віртуальні функції. Абстрактні класи. Поліморфізм.	4
8	Шаблони функцій і класів. Створення шаблонів і використання їх у програмах C++.	4
9	Вступ до C#. Створення консольних додатків на C#.	2
10	Робота з масивами. Використання класу Array.	4
11	Робота з рядковими типами. Використання класів Char, Char[], String та StringBuilder у мові C#.	2
12	Конструювання класів. Програмування класу з декількома конструкторами, функціями-властивостями і	4

	перевантаженими операціями. Перевірка працездатності створеного класу.	
13	Основи роботи з Visual Studio .NET. Створення Windows-додатків на мові C#.	2
14	Перевизначення методів інтерфейсів. Інтерфейс Icomparable.	4
15	Конструювання ієрархії класів у C#. Використання віртуальних функцій.	4
16	Обробка подій. Делегати.	4
17	Серіалізація об'єктів. Рефлексія	2
	Всього годин	60

6. Самостійна робота студентів

Необхідним елементом успішного засвоєння навчального матеріалу дисципліни є самостійна робота студентів з вітчизняною та іноземною спеціальною літературою. Самостійна робота є основним засобом оволодіння навчальним матеріалом у вільний від обов'язкових аудиторних навчальних занять час.

Для самостійного опрацювання виносяться наступні теми.

№ з/п	Назва теми	Кількість годин
1	Обробка винятків C++, типи виключень,	4
2	Послідовні контейнери C++, типи послідовних контейнерів	4
3	Створення класів винятків C#	4
4	Коваріантність і контраваріантних делегатів	4
5	Коваріантність і контраваріантних узагальнених інтерфейсів	8
6	Робота з JSON. Серіалізація в JSON. JsonSerializer	8
7	LINQ (Language-Integrated Query) мова запитів до джерела даних.	12
8	Відкладена ініціалізація і тип Lazy	4
9	Перетворення типів і клас Convert. Методи Parse і TryParse	4
10	Прибирання сміття, управління пам'яттю і покажчики Збиральник сміття в C #	8
	Всього годин	60

7. Контрольні питання, комплекси тестів для визначення рівня засвоєння знань студентами.

Навайжливіші контрольні питання

1. Еволюція методологій програмування. Парадигми програмування.
2. Поняття класу. Оголошення класу. специфікатори доступу.
3. Конструктори і деструктори. Порядок виклику конструкторів при спадкуванні.
4. Основні принципи об'єктного підходу. Абстрагування, інкапсуляція, спадкування, поліморфізм.
5. Відносини між класами. Асоціація та агрегація.
6. Перевантаження операторів.
7. Віртуальні методи. Механізм пізнього (відкладеного) або динамічного зв'язування.
8. Шаблони класів і шаблони функцій.
9. Порядок виклику конструкторів і деструкторів при наслідуванні.
10. Статичні і нестатичні методи класів. Виклик методів.

Тестові завдання

1. Під час виконання програми об'єкт класу Book створюється в рядку: А) Book * pnt; Б) pnt = new Book (); В) pnt -> GetYear (2010); Г) при запуску програми.
2. При створенні об'єкта класу А) створюються нові копії членів-даних і членів-функцій класу; Б) автоматично викликається конструктор; В) підключається бібліотека з файлами класу; Г) інше.
3. При зверненні до члена класу через ім'я об'єкта використовується операція (C++) А): Б) :: В) -> Г).
4. При зверненні до члена класу через покажчик на об'єкт використовується операція (C++) А): Б) :: В) -> Г).
5. Скільки об'єктів класу створюється в даному прикладі: monster Vasia; monster Super (200, 300); monstr stado [100]; monstr * beavis = new monstr (10); А) 611; Б) 103; В) 4; Г) інше.
6. Які файли описують клас? А) файл специфікації і файл реалізації; Б) файл з головною функцією; В) файл, в якому створюються об'єкти класу; Г) інше.
7. Що означають елементи опису членів класу private, protected і public?

А) це директиви елементів класу; Б) це специфікатори доступу В) це базові методи; Г) інше.
8. За що відповідають специфікатори доступу private і public? А) у private оголошуються тільки змінні; Б) public відповідає тільки за область видимості методів; В) private і public відповідають за область видимості зазначених у них елементів класу; Г) інше.
9. Що називається елементами класу? А) тільки члени-дані та члени-функції; Б) тільки конструктор і деструктор; В) тільки члени Дані; Г) тільки члени-функції.
10. monster :: ~ monstre () {}? А) конструктор; Б) деструктор; В) перевизначення; Г) перевантаження
11. Якщо клас містить конструктор за замовчуванням і конструктор з параметрами (C#), ці конструктори А) викликаються послідовно в порядку оголошення; Б) спочатку викликається конструктор за замовчуванням, потім - з параметрами; В) викликається тільки один з них; Г) при наявності завжди викликається конструктор з параметрами.
12. Коли перевантажується оператор, перевантаження діє (C#) А) для всіх класів, які є в програмі; Б) тільки для класу, в якому оператор визначається; В) для всіх випадків використання оператора в програмі; Г) також на стандартне визначення оператора з некласовим змінними
13. this - це (C#) А) неявне посилання на поточний об'єкт класу; Б) поточний клас; В) адреса поточного методу класу; Г) інше.
14. Який синтаксис використовується для вказівки класу батька в C #? А) class ChildClass : ParentClass Б) class ChildClass : base ParentClass В) class ChildClass : parent ParentClass Г) new ChildClass : ParentClass
15. Яким буде результат виконання наступного коду? <pre>class A { public int inc(int x) { return ++x; } } A obj=new A(); int y = 5; ConsoleWrite(obj.inc(y));</pre> А) 8; Б) 7; В) 6; Г) 5
16. Під час виконання програми об'єкт класу Book створюється в рядку: А) Book pnt; Б) pnt = new Book (); В) Book B=pnt; Г) Book []B= new Book [100];
17. Що означають елементи опису членів класу private, protected і public? А) це директиви елементів класу; Б) це специфікатори доступу В) це базові методи;

Г) інше.
18. Якщо програміст не вказав конструктор, компілятор (C#) А) видасть помилку; Б) створить абстрактний клас; В) створить клас, але видасть помилку при зверненні до об'єкта; Г) створить автоматично конструктор за замовчуванням.
19. Для чого можливо множинне наслідування (C#)? А) для класів; Б) для об'єктів; В) для інтерфейсів; Г) У мові C # множинне наслідування заборонено.
20. Імені деструктора безпосередньо передус символ (C#) А) :: Б) ~ В) & Г) #

8. Методи навчання

При викладанні дисципліни використовуються наступні методи навчання:

М1. Лекція (проблемна, інтерактивна) На лекціях застосовуються мультимедійні засоби та дискусії.

М2. Лабораторна робота – для використання набутих знань до розв'язування практичних завдань. Лабораторні роботи проводяться у двох частинах – пояснення завдання та захист студентами їхнього виконання.

М3. Проблемне навчання – створення проблемної ситуації для зацікавленого і активного сприйняття матеріалу.

М4. Проектне навчання (індивідуальне, малі групи, групове)

М5. Он-лайн навчання

Для розповсюдження усіх матеріалів як з боку викладача, так і з боку студента, використовується платформа *moodle*, що розташована за посиланням *elearn.nubip.ua*.

9. Форми контролю

При викладанні дисципліни передбачені такі форми контролю:

МК1. Тестування

МК2. Контрольне завдання

МК4. Методи усного контролю

МК5. Екзамен

МК6. Залік

МК7. Звіт

Для студентів денної форми навчання: усне опитування (МК4) та експрес контроль (МК1) на лабораторних заняттях, захист індивідуальних лабораторних завдань (МК7), аудиторні модульні контрольні роботи (МК2).

10. Розподіл балів, які отримують студенти

Оцінювання знань студента відбувається за 100-бальною шкалою і переводиться в національні оцінки згідно з табл. 1 «Положення про екзамени та заліки у НУБіП України» (наказ про уведення в дію від 26.04.2023р. протокол № 8)

Рейтинг здобувача вищої освіти, бали	Оцінка національна за результати складання екзаменів заліків	
	екзаменів	заліків
90-100	відмінно	зараховано
74-89	добре	
60-73	задовільно	
0-59	незадовільно	не зараховано

Для визначення рейтингу студента (слухача) із засвоєння дисципліни $R_{\text{дис}}$ (до 100 балів) одержаний рейтинг з атестації (до 30 балів) додається до рейтингу студента (слухача) з навчальної роботи $R_{\text{нр}}$ (до 70 балів): $R_{\text{дис}} = R_{\text{нр}} + R_{\text{ат}}$.

11. Навчально-методичне забезпечення

1. Ю.О. Міловідов. Методичні рекомендації до виконання лабораторних робіт з дисципліни «Об'єктно-орієнтоване програмування» – Видавничий центр НУБіП України, 2021. – 163 с.

2. Міловідов Ю.О. Методичні вказівки до написання курсових робіт з дисципліни «Об'єктно-орієнтоване програмування» – Видавничий центр НУБіП України, 2016. – 44 с.

3. Електронний навчальний курс «Об'єктно-орієнтоване програмування» – Режим доступу: <https://elearn.nubip.edu.ua/course/view.php?id=1423>

12. Рекомендовані джерела інформації

1. В.В. Бублик. «Об'єктно-орієнтоване програмування» Підручник – Київ, ІТ книга, 2015. – 637 с.

2. Ю.О. Міловідов. «Об'єктно-орієнтоване програмування» Навчальний посібник друге видання – Видавничий центр НУБіП України, 2022. – 323 с.