

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Кафедра комп'ютерних наук

«ЗАТВЕРДЖУЮ»

Декан факультету інформаційних технологій

_____ О. Г. Глазунова

«_____» _____ 20__ р.

РОЗГЛЯНУТО І СХВАЛЕНО

на засіданні кафедри комп'ютерних наук

Протокол № _____ від «_____» _____ 20__ р.

Завідувач кафедри

_____ Б. Л. Голуб

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ

Спеціальність – 123 «Комп'ютерна інженерія»

Факультет інформаційних технологій

Розробник: ст.викл. кафедри комп'ютерних наук , Міловідов Ю. О.

Київ – 2021 р.

1. Опис навчальної дисципліни

ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ

Галузь знань, спеціальність, освітньо-кваліфікаційний рівень		
Галузь знань	12 Інформаційні технології	
Спеціальність	123 «Комп'ютерна інженерія»	
Освітньо-кваліфікаційний рівень	бакалавр	
Характеристика навчальної дисципліни		
	денна форма навчання	Заочна форма навчання
Вид	вибіркова	
Семестр	5	
Загальна кількість годин	150	
Кількість кредитів ECTS	5	
Кількість змістових модулів	2	
Форма контролю	іспит	
Показники навчальної дисципліни для денної форми навчання		
Рік підготовки	3	
Лекційні заняття	30	4
Лабораторні заняття	30	10
Практичні, семінарські заняття		
Курсовий проект		
Самостійна робота	90	136
Кількість тижневих годин для денної форми навчання: аудиторних самостійної роботи студента –	4	

2. Мета та завдання навчальної дисципліни

Метою викладання дисципліни «Об'єктно-орієнтоване програмування» (ООП) є 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. Такі знання майбутній спеціаліст зможе застосовувати 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.

Загальні компетентності:

ЗК2. Здатність вчитися й оволодівати сучасними знаннями;

ЗК3. Здатність застосовувати знання у практичних ситуаціях;

ЗК7. Вміння виявляти, ставити та вирішувати проблеми.

Спеціальні(фахові, предметні) компетентності:

СК3. Здатність розробляти алгоритмічне та програмне забезпечення, компоненти комп'ютерних систем та мереж, Інтернет додатків, кібер-фізичних систем з використанням сучасних методів і мов програмування, а також засобів і систем автоматизації проектування.

Це забезпечує досягнення програмних результатів навчання ПР1, ПР2, ПР10, згідно з якими студент повинен знати:

- практичними навичками реалізації алгоритмів та налагодження програм;
- умінням самостійно опановувати нові методи та технології розробки програм. основних принципів реалізації класів;
- технології об'єктно-орієнтованої розробки програм;
- технології розробки програм на мові C++, C#;

Оволодіти:

- практичними навичками розробки об'єктно-орієнтованих алгоритмів;
- навичками застосування об'єктно-орієнтованих технологій при розробці програм;

Викладання дисципліни ґрунтується на знаннях по таких напрямках як:

- іноземна мова (бажано, англійська);
- вища математика;
- архітектура комп'ютера;
- програмування.

Дисципліна “ Об'єктно-орієнтоване програмування ” повинна передувати дисциплінам:

- системний аналіз;
- системне програмування;

3. Програма навчальної дисципліни

Модуль №1. Класи та абстракція даних

Тема №1. Введення в ООП

Лекція №1. Мета та завдання дисципліни. Значення та місце дисципліни в системі підготовки спеціалістів комп'ютерних наук. Загальні відомості про дисципліну, її зв'язок з іншими дисциплінами. Роль комп'ютерної техніки в сучасному сільськогосподарському виробництві.

Лекція №2. Визначення класу. Область дії класу та доступ до елементів класу. Управління доступом до елементів класу. Функції доступу та сервісні функції.

Лекція №3. Конструктори класів. Використання конструкторів з аргументами за замовченням. Деструктори. Послідовність виклику конструкторів та деструкторів.

Лекція №4. Використання елементів-даних та елементів-функцій. Присвоєння за замовченням. Повторне використання програмного забезпечення.

Тема №2. Основні принципи реалізації класів

Лекція №5. Константні об'єкти та константні елементи-функції. Композиція класів.

Лекція №6. Дружні функції та дружні класи. Операції виділення пам'яті *new, delete*.

Лекція №7. Статичні елементи класу. Абстракція даних та укриття інформації. Контейнерні класи та ітератори.

Модуль №2. Реалізація властивостей класу

Тема №3. Перевантаження операцій

Лекція №8. Основні принципи перевантаження операцій. Заборони на перевантаження операцій. Функції-операції як елементи класу і як дружні функції.

Лекція №9. Перевантаження одномісних операцій. Перевантаження двомісних операцій. Перетворення типів.

Тема №4. Спадкування

Лекція №10. Базові та похідня класи. Захищені елементи. Перетворення покажчиків базового класу у покажчики на похідний клас.

Лекція №11. Перевизначення елементів базового класу у похідному класі. Відкриті, захищені та закриті базові класи.

Лекція №12. Безпосередні та непрямі базові класи. Використання конструктора і деструктора у похідному класі.

Лекція №13. Неявне перетворення об'єктів похідного класу до базового. Спадкування у конструюванні програмного забезпечення. Складне спадкування.

Тема №5. Віртуальні функції і поліморфізм

Лекція №14. Віртуальні функції. Абстрактні базові класи і реальні класи.

Лекція №15. Поліморфізм. Нові класи і динамічне зв'язування. Приклади програм.

Лекція №16. Віртуальні деструктори. Приклад програми, яка демонструє інтерфейс та реалізацію.

4. Структура навчальної дисципліни

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	
Змістовий модуль 1. Класи та абстракція даних														
Тема 1. Введення в ООП	4	24	8		8		15							
Тема 2. Основні принципи реалізації класів	3	18	6		6		30							
Разом за змістовим модулем 1	7	42	14		14		45							
Змістовий модуль 2. Реалізація властивостей класу														
Тема 3. Перевантаження операцій	2	12	4		4		15							
Тема 4. Спадкування	3	18	6		6		15							
Тема 5. Віртуальні функції і поліморфізм	3	18	6		6		15							
Разом за змістовим модулем 2	8	48	16		16		45							
Всього годин	15	150	30		30		90							

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 замовченням. Розробка програм з використанням власних класів.	2
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
	Всього годин	30

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	4
4	Коваріантність і контраваріантних делегатів	4
5	Коваріантність і контраваріантних узагальнених інтерфейсів C#	16
6	Робота з JSON. Сериалізація в JSON. JsonSerializer	16
7	LINQ (Language-Integrated Query) мова запитів до джерела даних.	12
8	Відкладена ініціалізація і тип Lazy	10

9	Перетворення типів і клас Convert. Методи Parse і TryParse	8
10	Прибирання сміття, управління пам'яттю і покажчики Збиральник сміття в C #	12
	Всього годин	90

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

Екзаменаційні запитання <i>(максимальна оцінка 10 балів за відповідь на кожне запитання)</i>
1. Еволюція методологій програмування. Парадигми програмування
2. Побудувати систему класів для опису плоских геометричних фігур: коло, квадрат, прямокутник. Передбачити методи для обчислення площі об'єктів
Тестові завдання різних типів <i>(максимальна оцінка 10 балів за відповіді на тестові завдання)</i>
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

8. Методи навчання

При викладанні дисципліни використовуються наступні методи навчання:

М1. Лекція (проблемна, інтерактивна)

М2. Лабораторна робота – для використання набутих знань до розв’язування практичних завдань;

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

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

М5. Он-лайн навчання

9. Форми контролю

При викладанні дисципліни передбачені такі форми контролю:

МК1. Тестування

МК2. Контрольне завдання

МК4. Методи усного контролю

МК5. Екзамен

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

Поточний контроль		Рейтинг з навчальної роботи $R_{НР}$	Рейтинг з додаткової роботи $R_{ДР}$	Рейтинг штрафний $R_{ШТР}$	Підсумкова атестація (екзамен чи залік)	Загальна кількість балів
Змістовий модуль 1	Змістовий модуль 2					
0-100	0-100	0-70	0-20	0-5	0-30	0-100

Примітки. 1. Відповідно до «Положення про кредитно-модульну систему навчання в НУБіП України», затвердженого ректором університету 03.04.2009 р., рейтинг студента з навчальної роботи $R_{НР}$ стосовно вивчення певної дисципліни визначається за формулою

$$R_{НР} = \frac{0,7 \cdot (R_{ЗМ}^{(1)} \cdot K_{ЗМ}^{(1)} + \dots + R_{ЗМ}^{(n)} \cdot K_{ЗМ}^{(n)})}{K_{ДИС}} + R_{ДР} - R_{ШТР},$$

де $R_{ЗМ}^{(1)}, \dots, R_{ЗМ}^{(n)}$ – рейтингові оцінки змістових модулів за 100-бальною шкалою;

n – кількість змістових модулів;

$K_{ЗМ}^{(1)}, \dots, K_{ЗМ}^{(n)}$ – кількість кредитів ECTS, передбачених робочим навчальним планом для відповідного змістового модуля;

$K_{ДИС} = K_{ЗМ}^{(1)} + \dots + K_{ЗМ}^{(n)}$ – кількість кредитів ECTS, передбачених робочим навчальним планом для дисципліни у поточному семестрі;

$R_{ДР}$ – рейтинг з додаткової роботи;

$R_{ШТР}$ – рейтинг штрафний.

Наведену формулу можна спростити, якщо прийняти $K_{ЗМ}^{(1)} = \dots = K_{ЗМ}^{(n)}$. Тоді вона буде мати вигляд

$$R_{НР} = \frac{0,7 \cdot (R_{ЗМ}^{(1)} + \dots + R_{ЗМ}^{(n)})}{n} + R_{ДР} - R_{ШТР}.$$

Рейтинг з додаткової роботи $R_{ДР}$ додається до $R_{НР}$ і не може перевищувати 20 балів. Він визначається лектором і надається студентам рішенням кафедри за виконання робіт, які не передбачені навчальним планом, але сприяють підвищенню рівня знань студентів з дисципліни.

Рейтинг штрафний $R_{ШТР}$ не перевищує 5 балів і віднімається від $R_{НР}$. Він визначається лектором і вводиться рішенням кафедри для студентів, які матеріал змістового модуля засвоїли невчасно, не дотримувалися графіка роботи, пропускали заняття тощо.

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 введення в дію від 27.12.2019 р. № 1371).

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 $R_{ДИС}$ (до 100 балів) одержаний рейтинг з атестації (до 30 балів) додається до рейтингу студента (слухача) з навчальної роботи $R_{НР}$ (до 70 балів): $R_{ДИС} = R_{НР} + R_{АТ}$.

11. Методичне забезпечення

1. Ю.О. Міловідов. «Об'єктно-орієнтоване програмування» Навчальний посібник. – Видавничий центр НУБіП України, 2019. – 301 с.
2. Б.Л.Голуб, О.І.Примак. Методичні вказавки до виконання лабораторних робіт з дисциплін “Основи програмування”, “Програмування та алгоритмічні мови” (частина І). – Видавничий центр НУБіП, 2009. – 38 с.
3. Б.Л.Голуб, О.І.Примак. Методичні вказавки до виконання лабораторних робіт з дисциплін “Основи програмування”, “Програмування та алгоритмічні мови” (частина ІІ). – Видавничий центр НУБіП, 2009. – 50 с.
4. Басараб Р.М., Міловідов Ю.О. Методичні вказівки до виконання лабораторних робіт з дисципліни «Об'єктно-орієнтоване програмування» – Видавничий центр НУБіП України, 2016. – 44 с.
5. Міловідов Ю.О. Методичні вказівки до написання курсових робіт з дисципліни «Об'єктно-орієнтоване програмування» – Видавничий центр НУБіП України, 2016. – 44 с.

12. Рекомендована література

1. Герберт Шилдт. С# 4.0: полное руководство.: – М: ООО "И.Д. Вильямс", 2019. – 1056 с.
2. Герберт Шилдт. С++: базовый курс (С++ from the Ground Up) – 4-е изд. – М: ООО "И.Д. Вильямс", 2012, – 624с.
3. Роберт Лафоре. Объектно-ориентированное программирование в С++ – 4-е изд. СПб: «Питер», 2018. – 928с.
4. Полное руководство по языку программирования С# 7.0 и платформе .NET4.7 [Електронний ресурс] – Режим доступу: <https://metanit.com/sharp/tutorial/> – Назва з екрана.

13. Інформаційні ресурси

1. Електронний навчальний курс «Об'єктно-орієнтоване програмування» – Режим доступу: <https://elearn.nubip.edu.ua/enrol/index.php?id=876>
2. Полное руководство по языку программирования С# 7.0 и платформе .NET4.7 [Електронний ресурс] – Режим доступу: <https://metanit.com/sharp/tutorial/> – Назва з екрана.
3. Библиотека MSDN [Електронний ресурс]. – Режим доступу: <http://msdn.microsoft.com/ru-ru/library/default.aspx> – Назва з екрана.
4. .NET Framework [Електронний ресурс]. – Режим доступу: [https://msdn.microsoft.com/ru-ru/library/w0x726c2\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/w0x726c2(v=vs.110).aspx) – Назва з екрана.
5. Справочник по С# [Електронний ресурс]. – Режим доступу: <https://msdn.microsoft.com/ru-ru/library/618ayhy6.aspx> – Назва з екрана.