

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Кафедра комп'ютерних наук

«ЗАТВЕРДЖУЮ»

Декан факультету інформаційних технологій

_____ О. Г. Глазунова

« _____ » _____ 20 ____ р.

«СХВАЛЕНО»

на засіданні кафедри комп'ютерних наук

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

Завідувач кафедри

_____ Б. Л. Голуб

«РОЗГЛЯНУТО»

Гарант ОП «Інженерія програмного
забезпечення»

_____ доцент, к.ф.-м.н. Лялецький О.В.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ОБ'ЄКТНО-ОРІЄНТОВНЕ ПРОГРАМУВАННЯ

Спеціальність – 121 «Інженерія програмного забезпечення»

Освітня програма – «Інженерія програмного забезпечення»

Факультет інформаційних технологій

Розробник: ст. викладач Міловідов Ю. О.

Київ 2021

1. Опис навчальної дисципліни

ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ

Галузь знань, спеціальність, освітня програма, освітній ступінь		
Галузь знань	12 «Інформаційні технології»	
Спеціальність	121 – «Інженерія програмного забезпечення»	
Освітня програма	«Інженерія програмного забезпечення»	
Освітній ступінь	Бакалавр	
Характеристика навчальної дисципліни		
Вид	Обов'язкова	
	3-й семестр	4-й семестр
Загальна кількість годин	90	60
Кількість кредитів ECTS	3	2
Кількість змістових модулів	2	
Форма контролю	Залік	Іспит
Показники навчальної дисципліни для денної форми навчання		
Рік підготовки	2	
Семестр	3	4
Лекційні заняття	30 год	30 год
Практичні, семінарські заняття	-	-
Лабораторні заняття	30 год	30 год
Самостійна робота	30 год	-
Індивідуальні завдання	-	-
Кількість тижневих годин для денної форми навчання: аудиторних самостійної роботи студента –	4 год.	4 год.

2. Мета та завдання навчальної дисципліни

Метою викладання дисципліни “Об’єктно-орієнтоване програмування” є отримання студентами знань з області розробки алгоритмів та програмування. 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 може застосовувати як при подальшому навчанні, так і після отримання вищої освіти у своїй професійній діяльності.

У результаті вивчення дисципліни “Об’єктно-орієнтоване програмування” студенти повинні:

Мати знання з питань:

- основних принципів реалізації класів;
- технології об’єктно-орієнтованої розробки програм;
- технології розробки програм на мові C++;

Оволодіти:

- практичними навичками розробки об’єктно-орієнтованих алгоритмів;
- умінням розробляти програми на мові C++;
- практичними навичками реалізації алгоритмів на мові C++ та налагодження програм;
- вмінням самостійно опановувати нові методи та технології розробки програм.

Викладання дисципліни ґрунтується на знаннях по таких напрямках як:

- іноземна мова (бажано, англійська);
- вища математика;
- числові методи;
- архітектура комп’ютера;

та дисципліни “Основи програмування та алгоритмічні мови”.

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

- ”Системне програмування та операційні системи”;
- “Чисельні методи в інформатиці”;
- ”Технологія програмування та алгоритмічні мови”.

Загальні компетентності (ЗК)

К05. Здатність вчитися і оволодівати сучасними знаннями.

Фахові компетентності спеціальності (ФК)

К14. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.

К15. Здатність розробляти архітектури, модулі та компоненти програмних систем.

К23. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.

К24. Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення.

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 замовчуванням. Деструктори. Послідовність виклику конструкторів та деструкторів.

Лекція №4. Використання елементів-даних та елементів-функцій. Присвоєння за замовчуванням. 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 №1. 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 №2. Спадкування

Лекція №10. Базові та похідні класи. Захищені елементи. Перетворення показників базового класу у показники 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 №3. 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. Програма та 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			8		8								
Тема 2. Основні принципи реалізації класів			6		6								
Разом за змістовим модулем 1			14		14								
Змістовий модуль 2. Реалізація властивостей класу													
Тема 1. Перевантаження операцій			4		4								
Тема 2. Спадкування			6		6								
Тема 3. Віртуальні функції і поліморфізм			6		6								
Разом за змістовим модулем 2			16		16								
Усього годин			30		30								
Курсовий проект (робота) з _____ (якщо є в робочому навчальному плані)			-	-	-		-		-	-	-		-
Усього годин			30		30								

5. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1	Середовище для розробки програм Borland C++ - середовище для розробки програм на мові C++. Розробка структур як простих класів.	4
2	Визначення класу. Область дії класу та доступ до елементів класу. Управління доступом до елементів класу. Стандартні класи C++ для введення/виведення даних. Розробка власних класів (<u>complex</u> , <u>date</u>).	4
3	Створення простих конструкторів для ініціалізації елементів-змінних класів та виділення пам'яті. Використання конструкторів з аргументами за замовчуванням. Розробка програм з використанням власних класів – реалізація «арифметики» для комплексних чисел.	4
4	Створення простих деструкторів для звільнення пам'яті. Присвоєння за замовчуванням. Розробка програм з використанням власних класів – реалізація «арифметики» для комплексних чисел.	4
5	Розробка програм з використанням композиції класів – побудова двох класів (<u>date</u> , <u>student</u>), один з яких використовує інший клас як вкладений.	4
6	Розробка програм із використанням дружньої функції – розробка функції <u>presentTime</u> як дружньої для класу <u>date</u> . Використання операцій виділення пам'яті <i>new</i> , <i>delete</i> .	4
7	Розробка програм із використанням дружнього класу – розробка класу <u>decant</u> як дружнього для класу <u>student</u> .	4
8	Використання основних принципів перевантаження операцій. Розробка програми, яка дозволяє перевантажувати одномісні операції.	4
9	Розробка програми, яка реалізує «арифметику» комплексних чисел і використовує перевантаження операцій додавання та віднімання.	4
10	Розробка базового класу <u>person</u> та реалізація функцій для цього класу.	4
11	Розробка похідного класу <u>student</u> від базового класу <u>person</u> та реалізація функцій для класу <u>student</u> .	4

12	Розробка базового класу <u>state</u> та реалізація функцій для цього класу.	4
13	Розробка похідного класу <u>decant</u> від базового класу <u>state</u> та реалізація функцій для цього класу.	4
14	Використання віртуальних функцій. Розробка похідного класу <u>teacher</u> від базового класу <u>person</u> . Введення віртуальної функції в базовий клас <u>person</u> .	4
15	Застосування поліморфізму при проектуванні класів. Розробка програми по введенню даних щодо викладачів та студентів.	4
16	Контрольна перевірка виконаних програм.	4

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

1. 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.

2. При зверненні до члена класу через ім'я об'єкта використовується операція

- а):
- б) ::
- в) ->
- г).

3. При зверненні до члена класу через вказівник на об'єкт використовується операція

- а):
- б) ::
- в) ->
- г).

4. Що означають елементи опису членів класу `private`, `protected` і `public`?

- а) це директиви елементів класу;
- б) це модифікатори доступу
- в) це базові методи;
- г) інше.

5. За що відповідають модифікатори 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.

6. Коли викликається конструктор ?

- а) при створенні об'єкта;
- б) викликається програмістом;
- в) при запуску програми;
- г) інше.

7. Імені деструктора безпосередньо передуює символ

- а) `::`
- б) `~`
- в) `&`
- г) `#`

8. Якщо програміст не вказав конструктор, компілятор

- а) видасть помилку;
- б) створить абстрактний клас;
- в) створить клас, але видасть помилку при зверненні до об'єкта;

г) створить автоматично конструктор за замовчуванням.

7. 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

8. 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).

9. 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.

2. Згідно із зазначеним Положенням *підготовка і захист курсового проекту (роботи)* оцінюється за 100 бальною шкалою і далі переводиться в оцінки за національною шкалою.

Шкала оцінювання

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

10. 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 вказівки 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 вказівки 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.

11. Рекомендована література

1. Герберт Шилдт. С# 4.0: полное руководство.: – М: ООО "И.Д. Вильямс", 2019. – 1056 с.
2. Герберт Шилдт. С++: базовый курс (C++ 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. Б.Л.Голуб, Є.М.Шукайло. Методичний посібник до вивчення дисципліни “Програмування та алгоритмічні мови”. Методичний посібник. – Видавничий центр НАУ, 2003. – 64 с.

12. Інформаційні ресурси

1. ЕНК по даній дисципліні знаходиться за електронною адресою: <https://elearn.nubip.edu.ua/enrol/index.php?id=1423>