СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
«ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ»

Ступінь освіти	бакалавр
Освітня програма	Комп'ютерні науки
Тривалість викладання	3 семестр (5,6 чверть)
Кількість кредитів	5 кредитів ЄКТС (150 годин)
Заняття:	
лекції:	2 години/тиждень
лабораторні заняття:	2 години/тиждень
Мова викладання	українська

Сторінка курсу в СДО НТУ «ДП»: https://do.nmu.org.ua/course/view.php?id= 3441

Кафедра, що викладає Програмного забезпечення комп'ютерних систем

Викладач:
Приходченко Сергій Дмитрович
Доцент, к.т.н.

Персональна сторінка
https://pzks.nmu.org.ua/ua/teachers/prykhodchenkosd.php

E-mail: Prykhodchenko.s.d@nmu.one

1. Анотація до курсу

Курс з об'єктно-орієнтованого програмування (ООП) пропонує студентам глибоке вивчення концепцій та практичних навичок розробки програмних додатків, використовуючи об'єктно-орієнтований підхід. ООП - це сучасна та потужна парадигма програмування, яка дозволяє створювати модульні, розширювані та підтримувані програми. Під час курсу студенти ознайомляться з основними концепціями ООП, такими як класи, об'єкти, спадкування, поліморфізм, інкапсуляція тощо. Вони отримають практичні навички розробки програм з використанням мови програмування C#, але зможуть застосувати ці навички до інших мов, які підтримують ООП, таких як Java, C++, Python тощо.

Студенти будуть мати можливість розробляти власні програми та проекти, використовуючи принципи ООП, що допоможе їм стати більш

впевненими програмістами та підготувати їх для подальших завдань у галузі розробки програмного забезпечення.

Курс "Об'єктно-орієнтоване програмування" спрямований на студентів з попередніми знаннями з програмування та допоможе їм поглибити свої навички у цій важливій галузі інформатики..

2. Мета та завдання курсу

Мета дисципліни – ознайомити студентів з основними підходами до об’єктно-орієнтованих підходів, методів та засобів створення програмних систем, принципів побудови та функціонування процесу створення програмного забезпечення за допомогою об’єктно-орієнтованих парадигм та підходів; сформувати компетентності та надати сучасні практики щодо автоматизованого створення багатофункціональних додатків.

Завданнями дисципліни є:

- опанування теоретико-понятійної бази курсу;
- ознайомлення здобувачів з основними підходами до об’єктно-орієнтованих підходів, методів та засобів створення програмних систем, принципами побудови та функціонування процесу побудови програмного забезпечення з урахуванням ідеології та підходів об’єктно-орієнтованого програмування;
- ознайомлення здобувачів з сучасними практиками та інструментальними засобами щодо автоматизованого створення багатофункціональних додатків.

Мета цього курсу полягає в тому, щоб надати студентам необхідні знання та навички для успішного використання об'єктно-орієнтованого програмування у своїй майбутній кар'єрі, а також сприяти розвитку їхньої аналітичної та розробницької спроможності.

3. Результати навчання

Дисциплінарні результати навчання:

- розуміти концепції об’єктно-орієнтованого програмування, включаючи класи, об’єкти, спадкування, поліморфізм, інкапсуляцію та абстракцію;
- використовувати сучасні технології, мови програмування, методи та інструменти, що підтримують ООП;
- володіти навичками розробки програм, які використовують принципи ООП для створення модульних, розширюваних та підтримуваних програмних рішень;
- володіти навичками аналізу, проектування та впровадження програмних рішень, що використовують об’єктно-орієнтований підхід.
- володіти навичками ефективної роботи над спільними проектами, що використовують ООП, і впроваджувати кращі практики роботи з об’єктами та класами.

Дисциплінарні результати навчання сформовано на основі ПРН освітньо-професійної програми «Комп’ютерні науки» першого (бакалаврського) рівня вищої освіти (ПР9).

4. Структура курсу

Види та тематика навчальних занять	Внесок в загальну оцінку, %
ЛЕКЦІЇ	**40**
1. Вступ до об'єктно-орієнтованого програмування.	
Основи об'єктно-орієнтованого програмування	
Наслідування, Інкапсуляція, Поліморфізм, Абстракція	
Платформа .NET	
Основні елементи мови C #.	
Концепція класу.	
2. Основні елементи класів: дані, методи, конструктори	
Дані: поля і константи	
Методи	
Вхідні та вихідні параметри	
Конструктори	
3. Властивості, деструктори класів. Обробка виняткових ситуацій	
Збір сміття. Деструктори	
Властивості класу та об'єктів	
Обробка виняткових ситуацій	
Генерування виключень вручну	
4. Використання масивів в C#	
Одновимірний масив	
Методи	
Багатовимірні масиви	
Ступінчасті масиви	
5. Індексатори і операції класів	
Індексатори	
Багатовимірні індексатори	
Операції перетворення типу	
Перевантаження методів	
Операції класу	
6. Рядки, символи і регулярні вирази	
Символи. Методи	
Рядки типу string. Методи	
Регулярні вирази	
Методи класу Regex	
Тестова контрольна робота №1 (за темами 1-6).	**20**
7. Спадкування	
Основи спадкування. Створення похідних класів	
Доступ до елементів базового класу	
Використання конструкторів базового класу	
Перевизначення елементів базового класу. Приховування імен.	
Абстрактні класи	
Віртуальні методи	
8. Інтерфейси	
Синтаксис інтерфейсу. Реалізація інтерфейсів	

Види та тематика навчальних занять	Внесок в загальну оцінку, %
Операції is та as	
Стандартні інтерфейси .NET	
Використання операції typeof	
9. Структури та переліки	
Структури	
Перелік (enumeration)	
Методи	
10. Делегати і події	
Поняття делегата. Використання делегата	
Багатоадресатна передача. Передача делегатів в методи	
Забезпечення зв'язку «джерело – спостерігач» між об'єктами	
Події	
Правила обробки подій в середовищі .Net і використання вбудованого делегата EventHandler	
11. Створення Windows-додатків	
Порядок створення Windows-додатки	
Введення даних за допомогою елементів управління	
Створення і використання діалогових вікон	
12. Колекції класів	
Клас Stack	
Клас HashTable	
Тестова контрольна робота №2 (за темами 7-12).	**20**
ЛАБОРАТОРНІ ЗАНЯТТЯ	**60**
1. Основи мови програмування C#	
Підключення стандартних бібліотек.	
Програмування пробного додатку.	
Звіт з роботи № 1 та захист лабораторної роботи.	10
2. Робота з класами у проектах MS Visual C#	
Ініціалізація	
Оголошення класу, реалізацію конструкторів, set- та get-методів та деструкторів	
Звіт з роботи № 2 та захист лабораторної роботи.	10
3. Вивчення успадкування як методики повторного Використання коду у класах C#	
Методика автоматичної генерації класів C# з діаграм класів	
Програмування додатку	
Звіт з роботи № 3 та захист лабораторної роботи.	10
4. Вивчення поліморфізму у класах C#	
Створення віртуальних методів	
Перевантаження арифметичних операцій та операцій порівняння	
Робота з GUI	
Звіт з роботи № 4 та захист лабораторної роботи.	10
5. Вивчення інтерфейсів C#	
Зміна реалізації інтерфейсів у похідних класах	
Явне застосування інтерфейсів	
Звіт з роботи № 5 та захист лабораторної роботи.	10

Види та тематика навчальних занять	Внесок в загальну оцінку, %
6. Вивчення інтерфейсів, делегатів та колекцій у класах C#	
Створення делегатів	
Створення колекцій	
Звіт з роботи № 6 та захист лабораторної роботи.	**10**
РАЗОМ	**100**

5. Технічне обладнання та/або програмне забезпечення

Використовуються лабораторії кафедри програмного забезпечення комп'ютерних систем (комп'ютерне та мультимедійне обладнання). Дистанційна платформа Moodle, MS Office 365, Microsoft Teams, GIT, Visual Studio Community.

6. Система оцінювання та вимоги

6.1. Навчальні досягнення здобувачів вищої освіти за результатами вивчення курсу оцінюватимуться за шкалою, що наведена нижче:

Рейтингова шкала	Інституційна шкала
90 – 100	відмінно
74-89	добре
60-73	задовільно
0-59	незадовільно

6.2. Здобувачі вищої освіти можуть отримати **підсумкову оцінку** з навчальної дисципліни **на підставі поточного оцінювання знань** за умови, якщо набрана кількість балів з поточного тестування та виконання і захисту лабораторних робіт складатиме не менше 60 балів.

Теоретична частина оцінюється за результатами здачі двох контрольних тестових робіт, кожна з яких містить тестові закриті запитання з однією вірною відповіддю (максимальна кількість – 20 балів за кожною тестовою роботою). Загалом за дві контрольні тестові робіти отримується **максимум 40 балів**, тобто 40% від оцінки за дисципліну.

Лабораторні роботи (шість робіт – у вигляді індивідуального завдання з кожної, розподіл % див. в таблиці розділу 4) виконуються у письмовому вигляді (звіт з кожної роботи оцінюється в межах балів, представлених в таблиці розділу 4, загалом лабораторні враховуються як 60% (максимум 60 балів). При несвоєчасному здаванні роботи оцінка знижується вдвічі. У сумі за лабораторну частину курсу при поточному оцінюванні отримується **максимум 60 балів**.

Отримані бали за теоретичну частину та лабораторні роботи додаються і є поточною успішністю за вивчення навчальної дисципліни. Максимально за поточною успішністю здобувач вищої освіти може набрати 100 балів.

Максимальне оцінювання поточного контролю в балах:

Теоретична частина	Лабораторні заняття	**Разом**
40	60	**100**

6.3. Критерії оцінювання підсумкової роботи. У випадку якщо здобувач вищої освіти за поточною успішністю отримав менше 60 балів та/або прагне поліпшити оцінку, проводиться **підсумкове оцінювання (екзамен)** під час сесії.

Екзамен проводиться у вигляді комплексної контрольної роботи, яка включає запитання з теоретичної та лабораторної частини курсу. Білет складається з **30 тестових завдань** з чотирма варіантами відповідей, одна правильна відповідь оцінюється в 2 бали (**разом 60 балів**) та **2 завдання** з практичної частини, кожне з запитань оцінюється максимум у 20 балів (**разом 40 балів**). Отримані бали за тестові завдання та завдання з практичної частини додаються і є підсумковою оцінкою за вивчення навчальної дисципліни. Максимально за підсумковою роботою здобувач вищої освіти може набрати 100 балів.

7. Політика курсу

7.1. Політика щодо академічної доброчесності. Академічна доброчесність здобувачів вищої освіти є важливою умовою для опанування результатами навчання за дисципліною і отримання задовільної оцінки з поточного та підсумкового контролів. Академічна доброчесність базується на засудженні практик списування (виконання письмових робіт із залученням зовнішніх джерел інформації, крім дозволених для використання), плагіату (відтворення опублікованих текстів інших авторів без зазначення авторства), фабрикації (вигадування даних чи фактів, що використовуються в освітньому процесі). Політика щодо академічної доброчесності регламентується положенням «Положення про систему запобігання та виявлення плагіату у Національному технічному університеті «Дніпровська політехніка» https://cutt.ly/MCfh5kv

У разі порушення здобувачем вищої освіти академічної доброчесності (списування, плагіат, фабрикація), робота оцінюється незадовільно та має бути виконана повторно. При цьому викладач залишає за собою право змінити тему завдання.

7.2. Комунікаційна політика. Здобувачі вищої освіти повинні мати активовану корпоративну університетську пошту.

Усі письмові запитання до викладачів стосовно курсу мають надсилатися на університетську електронну пошту.

7.3. Політика щодо перескладання. Роботи, які здаються із порушенням термінів без поважних причин оцінюються на нижчу оцінку. Перескладання підсумкового оцінювання відбувається із дозволу деканату за наявності поважних причин (наприклад, лікарняний).

7.4 Політика щодо оскарження оцінювання. Якщо здобувач вищої освіти не згоден з оцінюванням його знань він може опротестувати виставлену викладачем оцінку у встановленому порядку.

8. Рекомендовані джерела інформації

1. Гаркуша І.М. Конспект лекцій з дисципліни «Програмування». Частина 1. Для студентів галузі знань 12 «Інформаційні технології». Дніпро: НТУ «ДП», 2020. 103 с.
2. Бублик В.В. Об’єктно-орієнтоване програмування: [Підручник] / В.В. Бублик. – К.: ІТ книга, 2015. – 624 с.
3. Алхімова С. М. Об’єктно-орієнтоване програмування : підручник. У 2-х ч. Ч. 2. Об’єктно-орієнтований підхід до розробки програмного забезпечення / С. М. Алхімова. – Київ: КГП ім. Ігоря Сікорського, Вид-во «Політехніка», 2019.-192 с.
4. Основи об’єктно-орієнтованого програмування: навч. посібник / Гришанович Т. О., Глинчук Л. Я.; ВНУ імені Лесі Українки. Луцьк: ВНУ імені Лесі Українки, 2022. – 120 с.

Додаткова література

1. Balagurusamy E. Object Oriented Programming with C++. Gautam Buddha Nagar, Uttar Pradesh: Mc Graw Hill India, 2019. 537 p.
2. Bancila M. Modern C++ Programming Cookbook: Master C++ core language and standard library features. Birmingham, UK: Packt Publishing, 2020. 750 p.
3. Deitel P., Deitel H. C++20 for Programmers. London, UK: Pearson Education Limited, 2021.1008 p.
4. Kirk D. R. Deciphering Object-Oriented Programming with C++: A practical, indepth guide to implementing object-oriented design principles to create robust code. Birmingham, UK: Packt Publishing, 2022. 594 p.
5. Kusswurm D. Modern Parallel Programming with C++ and Assembly Language. New York, NY: Apress, 2022. 656 p.
6. Meyers S. Effective Modern C++: 42 Specific Ways to Improve Your Use of C++11 and C++14. Sebastopol, CA: O'reilly Media, 2014. 334 p.
9. Nibedit D. Cross-Platform Development with Qt 6 and Modern C++. Birmingham, UK: Packt Publishing, 2021. 442 p.