

Міністерство освіти і науки України

Тернопільський національний технічний університет імені Івана Пулюя

(повне найменування вищого навчального закладу)

Факультет комп'ютерно-інформаційних систем і програмної інженерії

/назва факультету/

Кафедра комп'ютерних наук

/назва кафедри/

ЗАТВЕРДЖУЮ

Декан факультету

Ігор БАРАН

(підпис)

(прізвище та ініціали)

« 02 »

09

2024 р.



РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ

/назва дисципліни/

галузь знань

12 Інформаційні технології

/шифр і назва галузі знань/

рівень вищої освіти

перший (бакалаврський)

/назва/

спеціальність

122 Комп'ютерні науки

/шифр і назва/

освітня програма

«Комп'ютерні науки»

/назва/

спеціалізація

/назва/

вид дисципліни

обов'язкова дисципліна циклу професійної підготовки

/обов'язкова/вибіркова/

Тернопіль – 2024 рік

Робоча програма з навчальної дисципліни Об'єктно-орієнтоване програмування
/назва дисципліни/

для студентів факультету комп'ютерно-інформаційних систем і програмної інженерії

/назва факультету(ів)/

Розробник:

доцент кафедри комп'ютерних наук,

кандидат технічних наук, доцент

/посада, науковий ступінь та вчене звання/

Готович
/підпис/

Володимир ГОТОВИЧ

/ініціали та прізвище/

Робоча програма розглянута та схвалена

на засіданні кафедри комп'ютерних наук

/назва/

Протокол від « 26 » серпня 2024 року № 1

Завідувач кафедри

/підпис/

Ігор БОДНАРЧУК

/ініціали та прізвище/

Робоча програма розглянута та схвалена НМК

факультету комп'ютерно-інформаційних систем і програмної інженерії

Протокол від « 02 » вересня 2024 року № 1

Секретар НМК

/підпис/

Богдана МЛИНКО

/ініціали та прізвище/

Робоча програма погоджена:

Спеціальність 122 Комп'ютерні науки

/шифр і назва/

освітня програма «Комп'ютерні науки»

/назва/

Завідувач випускової кафедри

/підпис/

Ігор БОДНАРЧУК

/ініціали та прізвище/

Гарант освітньої програми

/підпис/

Леся ДМИТРОЦА

/ініціали та прізвище/

1. Структура навчальної дисципліни

Показник	Всього годин	
	Денна форма навчання (ДФН)	Заочна (дистанційна) форма навчання (ЗФН)
Кількість кредитів/год	4,5/135	4,5/135
Аудиторні заняття, год.	64	12
Самостійна робота, год.	71	123
Аудиторні заняття:		
– лекції, год.	32	6
– лабораторні заняття, год.	32	6
– практичні заняття, год.	–	–
– семінарські заняття, год.	–	–
Самостійна робота:		
– підготовка до лабораторних (практичних семінарських) занять, год.	20	40
– опрацювання окремих розділів програми, які не виносяться на лекції, год.	43	63
– виконання контрольних завдання, год.	–	–
– виконання індивідуальних завдань, год.	–	–
– виконання курсових проектів (робіт), год.	–	–
– підготовка та складання заліків, екзаменів, контрольних робіт, рефератів, есе, тестування, год.	8	20
Екзамен	Е	Е
Залік	–	–

Частка годин самостійної роботи студента:

денна форма навчання - 53 %;

заочна (дистанційна) форма навчання - 91 %.

2. Мета та завдання навчальної дисципліни

2.1. Мета вивчення навчальної дисципліни

Мета вивчення навчальної дисципліни «Об'єктно-орієнтоване програмування» полягає в отриманні студентами теоретичних знань та практичних навичок по проектуванню та розробці програмного забезпечення на основі застосування принципів об'єктно-орієнтованого підходу. Також під час освоєння дисципліни студентами вивчаються основні засоби та можливості мови програмування C#.

2.2. Завдання навчальної дисципліни

Завданням дисципліни є оволодіння студентами теоретичними основами об'єктно-орієнтованого підходу до проектування та розробки програмного забезпечення та закріплення їх шляхом набуття практичних навичок програмування на мові C#.

По завершенню вивчення дисципліни студент повинен набути таких знань та вмінь:

Знати:

1. Основні характеристики та можливості платформи Microsoft .Net Framework;
2. Теоретичні основи використання системи контролю версій git та роботи з відповідними консольним та графічним клієнтами;
3. Базові засоби мови програмування C# та деякі більш складні теми (обробка виключних ситуацій, робота з файлами, застосування потоків, перевантаження методів та операцій, індексатори, стандартні інтерфейси, стандартні класи-колекцій);
4. Теоретичні основи об'єктно-орієнтованого підходу до розробки програмного забезпечення та як вони реалізуються засобами мови C# (клас, об'єкт, абстракція, інкапсуляція, успадкування, поліморфізм, інтерфейс, класи-прототипи).

Вміти:

1. Використовувати базові засоби мови програмування C# для розробки програм як із консольним, так із графічним інтерфейсом користувача;
2. Використовувати систему контролю версій git;
3. Використовувати сховище (репозиторій) проектів GitLab/GitHub;
4. Проектувати програмне забезпечення на основі використання принципів об'єктно-орієнтованого підходу;
5. Розробляти програмне забезпечення на мові програмування C# та з використанням можливостей інтегрованого середовища розробки MS Visual Studio/MS Visual Studio Code.

Вивчення навчальної дисципліни передбачає формування та розвиток у студентів загальних компетентностей (ЗК) та спеціальних (фахових) компетентностей (СК) згідно освітньої програми «Комп'ютерні науки» першого (бакалаврського) рівня вищої освіти.

Загальні:

- ЗК1.** Здатність до абстрактного мислення, аналізу та синтезу;
- ЗК2.** Здатність застосовувати знання у практичних ситуаціях;
- ЗК6.** Здатність вчитися й оволодівати сучасними знаннями;
- ЗК7.** Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

Спеціальні (фахові):

СК3. Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.

СК8. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-

орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.

Вивчення навчальної дисципліни передбачає формування у студента таких **програмних результатів навчання (ПР):**

ПР5. Проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій.

ПР14. Застосовувати знання методології та CASE-засобів проектування складних систем, методів структурного аналізу систем, об'єктноорієнтованої методології проектування при розробці і дослідженні функціональних моделей організаційноекономічних і виробничотехнічних систем.

3. Опис навчальної дисципліни

3.1. Лекційні заняття

№	Тема заняття та короткий зміст	Кількість годин	
		ДФН	ЗФН
1	2	3	4
1.	Тема 1. Основи технологічної платформи MS .Net Framework. Загальна характеристика платформи MS .Net Framework. Архітектура платформи MS .Net. Структура бібліотеки класів FCL. Простір імен. Загальномовне середовище виконання – динамічна складова MS .NET Framework. Універсальна система типів.	2	0,5
2.	Тема 2. Основи мови програмування C#. Основні поняття C#. Система типів C#. Змінні, операції, вирази. Перетворення вбудованих арифметичних типів-значень. Перетворення із стрічкового типу в числові. Вимоги до оформлення програмного коду: стиль, розбиття на структурні одиниці, найменування змінних, класів, об'єктів. Відлагодження програмного коду в середовищі Visual Studio.	4	0,5
3.	Тема 3. Оператори мови C#. Базові конструкції структурного програмування. Умовні оператори. Оператори циклу. Оператори переходу. Базові конструкції структурного програмування.	2	0,5
4.	Тема 4. Масиви і стандартні класи мови C#. Робота з масивами. Оператор foreach. Клас Random і його методи. Клас Match і його методи.	2	0,5
5.	Тема 5. Стандартні класи мови C#. Тип char. Масиви типу char. Робота із стрічками. Клас String та його методи. Клас StringBuilder і його методи. Обробка помилок. Клас Exception. Структури в мові C#. Тип перерахування в мові C#. Комплексний приклад: вдосконалений консольний калькулятор.	2	0,5
6.	Тема 6. Основи вводу-виводу та робота з файлами. Введення-виведення на консоль. Файлове введення-виведення. Потoki. Абстрактний клас Stream і його нащадки. Класи StreamReader, StreamWriter, StringReader, StringWriter. Робота з дисками. Робота з каталогами. Робота з файлами. Класи File і FileInfo. Клас FileStream. Читання і запис текстових файлів. Робота з бінарними файлами. Класи BinaryWriter і BinaryReader.	2	0,5

1	2	3	4
7.	Тема 7. Класи. Дві ролі класів в C#. Клас як модуль і клас як тип даних. Інтерфейс і реалізація класу. Основні принципи об'єктно-орієнтованого програмування та їх реалізація в C#. Порівняння процедурної та об'єктно-орієнтованої парадигм програмування. Методи класу. Використання вказівника this в тілі класу. Конструктори. Типи конструкторів. Властивості get і set. Порівняння класів та структур в мові C#.	4	0,5
8.	Тема 8. Реалізація принципу успадкування. Зв'язки (відношення) між класами в ООП: успадкування, реалізація, асоціація, композиція, агрегація. Організація ієрархії успадкування класів. Використання в похідному класі конструктора базового класу з параметрами. Віртуальні методи. Пізні і ранні зв'язування об'єктів класу. Абстрактні класи і методи. Приховані (закриті) класи. Клас Object – базовий клас ієрархії класів C#.	4	0,5
9.	Тема 9. Поліморфізм. Перевантаження методів. Методи зі змінною кількістю аргументів. Індиксатори. Переважання операцій. Приклад класу з перевантаженими методами і операціями.	4	0,5
10.	Тема 10. Інтерфейси. Інтерфейс як окремий випадок абстрактного класу. Стратегії реалізації інтерфейсу. Інтерфейси і поля. Інтерфейси та успадкування. Типи відношень між класами.	2	0,5
11.	Тема 11. Вбудовані інтерфейси мови C#. Впорядкованість об'єктів і інтерфейс IComparable. Впорядкування об'єктів за кількома критеріями. Інтерфейс IComparer. Перелічуваність об'єктів і інтерфейси. Інтерфейс IEnumerable. Ітератори та інтерфейс IEnumerable.	2	0,5
12.	Тема 12. Класи-колекції. Поняття абстрактного типу даних. Абстрактні структури даних в мові C#. Колекції. Простір імен System.Collections. Клас Stack (Стек). Клас Queue (Черга). Клас ArrayList (Динамічний масив). Класи-прототипи (generics).	2	0,5
Усього годин		32	6

3.2. Лабораторні заняття

№	Тема заняття	Кількість годин	
		ДФН	ЗФН
1.	Знайомство з системою контролю версій Git.	2	1
2.	Реалізація розгалужених і циклічних алгоритмів. Знайомство з додатками Windows Presentation Foundation.	4	1
3.	Робота з масивами в мові C#. Знайомство з графічними git-клієнтами.	3	1
4.	Структури. Масиви структур. Передача параметрів у методи. Модифікатори параметрів. Обробка виключних ситуацій.	4	–
5.	Введення/виведення даних з використанням файлів. Робота із типом даних enum.	2	–
6.	Класи та об'єкти. Реалізація конструкторів та методів класів.	4	1
7.	Створення та використання бібліотеки класів.	4	1
8.	Успадкування класів. Поліморфізм та віртуальні методи.	4	1
9.	Інтерфейси в мові C#.	2	–
10.	Класи-колекції та класи-прототипи в мові C#.	3	–
Усього годин		32	6

3.3. Самостійна робота

№	Найменування робіт	Кількість годин	
		ДФН	ЗФН
1	2	3	4
1.	Тема 1. Основи технологічної платформи MS .Net Framework. Платформа .Net Core як розвиток платформи .Net Framework.	6	9
2.	Тема 2. Основи мови програмування C#. Пріоритет виконання операцій в мові C#. Приведення типів.	6	9

1	2	3	4
3.	Тема 3. Оператори мови C#. Базові конструкції структурного програмування. Поняття рефлексії. Метадані.	5	8
4.	Тема 4. Масиви і стандартні класи мови C#. Зубчасті масиви. Стандартна бібліотека класів.	5	9
5.	Тема 5. Робота з стрічками. Обробка виключних ситуацій. Стандартні методи для роботи із стрічками. Клас System.Exception.	5	9
6.	Тема 6. Робота з файлами. Стандартні потоки C#. Архітектура потоків.	5	8
7.	Підготовка до тестування	4	10
8.	Тема 7. Класи. Клас як шаблон створення об'єктів. Порівняння процедурної та об'єктно-орієнтованої парадигм програмування.	6	9
9.	Тема 8. Реалізація принципу успадкування. Клас Object.	5	8
10.	Тема 9. Поліморфізм. Статичний та динамічний поліморфізм.	5	8
11.	Тема 10. Інтерфейси. Затосування інтерфейсів для розширення поведінки класів.	5	9
12.	Тема 11. Вбудовані інтерфейси мови C#. Особливості реалізації членів інтерфейсу.	5	8
13.	Тема 12. Класи-колекції. Реалізація засобами C# колекцій: список, черга, стек, словник, хеш-таблиця, граф, вектор, кортеж.	5	9
14.	Підготовка до тестування	4	10
Усього годин		71	123

4. Критерії оцінювання результатів навчання студентів

Форма підсумкового семестрового контролю – **екзамен**.

Підсумкова оцінка складається з суми балів, отриманих студентом при захисті лабораторних робіт, двох тестувань (за перший та другий модульний контроль) та отриманих балів за екзамен.

За виконання 10-ти лабораторних робіт студент отримує максимально 60 балів ($10 * 6$). Захист звіту з лабораторної роботи оцінюється відповідною кількістю балів поданою в таблиці.

За результатами проходження модульного контролю №1 студент отримує від 0 до 7 балів, модульного контролю №2 – від 0 до 8 балів.

Підсумковий семестровий контроль (екзамен) може відбуватися у формі підсумкового тестування або письмового іспиту (максимум – 25 балів).

Контроль здійснюється засобами електронного навчального курсу (ЕНК) на сервері дистанційного навчання <http://dl.tntu.edu.ua>, ідентифікатор дисципліни ID 4773.

МОДУЛЬ 1			МОДУЛЬ 2			ПІДСУМКОВА СЕМЕСТРОВА ОЦІНКА (ЕКЗАМЕН)	РАЗОМ З ДИСЦИПЛІНИ
АУДИТОРНА ТА САМОСТІЙНА РОБОТА			АУДИТОРНА ТА САМОСТІЙНА РОБОТА				
ТЕОРЕТИЧНИЙ КУРС (ТЕСТУВАННЯ)	ЛАБОРАТОРНА РОБОТА		ТЕОРЕТИЧНИЙ КУРС (ТЕСТУВАННЯ)	ЛАБОРАТОРНА РОБОТА			
СЕМЕСТРОВА ОЦІНКА						25	100
7	30		8	30			
№ ЛЕКЦІЙ	ВИД РОБІТ	БАЛ	№ ЛЕКЦІЙ	ВИД РОБІТ	БАЛ		
Лекція 1	Лаб. роб. №1	6	Лекція 7	Лаб. роб. №6	6		
Лекція 2			Лекція 8	Лаб. роб. №7	6		
Лекція 3	Лаб. роб. №2	6	Лекція 9	Лаб. роб. №8	6		
Лекція 4	Лаб. роб. №3	6	Лекція 10	Лаб. роб. №9	6		
Лекція 5	Лаб. роб. №4	6	Лекція 11				
Лекція 6	Лаб. роб. №5	6	Лекція 12	Лаб. роб. №10	6		

До підсумкового семестрового контролю (складання екзамену) допускаються студенти, які протягом семестру виконали всі види навчальної роботи, успішно

пройшли проміжні (модульні) контролі і набрали не менше 45 балів семестрової оцінки та за умови отримання не менше 60% балів за результатами кожного проміжного (модульного) контролю рівня знань.

Підсумкова оцінка записується за 100-бальною шкалою із подальшим переведенням її у шкалу Європейської кредитно-трансферної системи (ECTS) відповідно A, B, C, D, E, F, FX. При цьому виставляється також національна оцінка по чотирьохбальній шкалі (A – «відмінно», B, C – «добре», D, E – «задовільно» і F, FX – «незадовільно»).

5. Навчально-методичне забезпечення

1. Методичні вказівки до виконання лабораторних робіт з дисципліни «Об'єктно-орієнтоване програмування» для здобувачів освітнього ступеня «бакалавр» спеціальності 122 «Комп'ютерні науки» всіх форм навчання / укл.: Готович В.А. – Тернопіль : Тернопільський національний технічний університет імені Івана Пулюя, 2022. – 76 с.

2. Конспект лекцій з дисципліни “Об'єктно-орієнтоване програмування” для здобувачів освітнього ступеня «бакалавр» спеціальності 122 «Комп'ютерні науки» всіх форм навчання / укладачі : Василенко Я. П., Готович В. А. – Тернопіль : Тернопільський національний технічний університет імені Івана Пулюя, 2022. – 195 с.

6. Рекомендована література

Базова

1. Joseph Albahari, C# 10 in a Nutshell: The Definitive Reference. O'Reilly Media; 1st edition (March 15, 2022). ISBN-10 : 1098121953

2. Дудзяний І. М. Об'єктно-орієнтоване моделювання програмних систем: Навчальний посібник. – Львів: Видавничий центр ЛНУ імені Івана Франка, 2007. – 108 с.

3. C# reference [Електронний ресурс]. – Режим доступу: <https://learn.microsoft.com/en-us/dotnet/csharp/language-reference>

4. Git [Електронний ресурс]. – Режим доступу: <https://docs.gitlab.com/ee/topics/git/#git>

5. Git documentation [Електронний ресурс]. – Режим доступу: <https://git-scm.com/doc>

6. GitHub Training Kit [Електронний ресурс]. – Режим доступу: <https://training.github.com/downloads/ua/github-git-cheat-sheet/>

Допоміжна

1. Алексенко О.В. Технології програмування та створення програмних продуктів. Конспект лекцій. – Суми : Сумський державний університет, 2013. – 133 с.

2. Шаховська Н. Б., Голощук Р. О. Алгоритми і структури даних. Навчальний посібник. Львів : Магнолія, 2018. 216 с.

7. Інформаційні ресурси

1. Сертифікований електронний навчальний курс “Об’єктно-орієнтоване програмування” в системі дистанційного навчання ТНТУ імені І. Пулюя. [Електронний ресурс]. – Режим доступу: <https://dl.tntu.edu.ua/bounce.php?course=4773>.

2. C# Tutorial [Електронний ресурс]. – Режим доступу: <https://www.w3schools.com/cs/index.php>

3. Learn C# Programming [Електронний ресурс]. – Режим доступу: <https://www.tutorialsteacher.com/csharp>

4. C# Tutorial [Електронний ресурс]. – Режим доступу: <https://www.tutorialspoint.com/csharp/index.htm>

8. Зміни та доповнення до робочої програми навчальної дисципліни

№	Зміст внесених змін (доповнень)	Дата і № протоколу засідання кафедри	Примітки