



1. Загальна інформація про дисципліну:

Назва дисципліни	ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ
Викладач	Готович Володимир Анатолійович
Профайл викладача	https://tntu.edu.ua
Контактний тел.	Комутатор (0352) 51-97-00, внутрішній 1706 (із зовнішніх телефонів через комутатор, тоді в режимі тонального набору набрати внутрішній номер), 380931688324
E-mail:	gotovych@gmail.com
Сторінка дисципліни в A-Tutor	https://dl.tntu.edu.ua , ID 4773
Консультації	Згідно графіку консультацій у першому семестрі

2. Коротка анотація до дисципліни

Навчальна дисципліна «Об'єктно-орієнтоване програмування» належить до обов'язкових дисциплін освітньої програми «Інформаційні системи та технології» першого (бакалаврського) рівня вищої освіти. Галузь знань 12 «Інформаційні технології», спеціальності 126 «Інформаційні системи та технології». Викладається у 3 семестрі (другий курс) обсягом 5,5 кредитів ECTS. Формою підсумковою контролю є екзамен.

3. Мета та завдання дисципліни

Мета 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#.

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#.

4. Формат дисципліни:

Дисципліна передбачає проведення лекційних, лабораторних занять та консультацій. Для кращого розуміння та засвоєння викладеного матеріалу дисципліна має супровід у вигляді електронного навчального курсу в системі A-Tutor (<https://dl.tntu.edu.ua>). Електронний навчальний курс має лекційний матеріал, лабораторні роботи, питання модульного та підсумкового контролю, систему оцінювання.

5. Результати навчання:

В результаті вивчення дисципліни студент повинен **знати**:

- ✓ основні характеристики та можливості платформи Microsoft .Net Framework;
- ✓ теоретичні основи використання системи контролю версій git та роботи з відповідними консольним та графічним клієнтами;
- ✓ 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-колекції);
- ✓ 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).

В результаті вивчення дисципліни студент повинен **вміти**:

- ✓ використовувати базові засоби мови програмування C# для розробки програм як із консольним, так із

графічним інтерфейсом;

- ✓ використовувати систему контролю версій git;
- ✓ проектувати програмне забезпечення на основі використання принципів об'єктно-орієнтованого підходу;
- ✓ розробляти програмне забезпечення на мові програмування C# із використанням принципів об'єктно-орієнтованого підходу.

Вивчення навчальної дисципліни передбачає формування та розвиток у студентів **загальних компетентностей (КЗ) та спеціальних (фахових) компетентностей (КС)** згідно освітньої програми.

Загальні:

КЗ 1. Здатність до абстрактного мислення, аналізу та синтезу;

КЗ 2. Здатність застосовувати знання у практичних ситуаціях.

Спеціальні (фахові):

КС 1. Здатність аналізувати об'єкт проектування або функціонування та його предметну область;

КС 4. Здатність проектувати, розробляти та використовувати засоби реалізації інформаційних систем, технологій та інфокомунікацій (методичні, інформаційні, алгоритмічні, технічні, програмні та інші);

КС 12. Здатність управляти та користуватися сучасними інформаційно-комунікаційними системами та технологіями (у тому числі такими, що базуються на використанні Інтернет).

Вивчення навчальної дисципліни передбачає формування у студента таких прикладних **результатів навчання (ПР)** згідно освітньої програми:

ПР 3. Використовувати базові знання інформатики й сучасних інформаційних систем та технологій, навички програмування, технології безпечної роботи в комп'ютерних мережах, методи створення баз даних та інтернет-ресурсів, технології розроблення алгоритмів і комп'ютерних програм мовами високого рівня із застосуванням об'єктно-орієнтованого програмування для розв'язання задач проектування і використання інформаційних систем та технологій;

ПР 6. Демонструвати знання сучасного рівня технологій інформаційних систем, практичні навички програмування та використання прикладних і спеціалізованих комп'ютерних систем та середовищ з метою їх запровадження у професійній діяльності;

ПР 7. Обґрунтовувати вибір технічної структури та розробляти відповідне програмне забезпечення, що входить до складу інформаційних систем та технологій;

ПР 13. Використання технології розробки інформаційних систем з застосуванням системи контролю версій (GIT).

6. Обсяг дисципліни

Вид заняття	Загальна кількість годин
лекції	32
лабораторні заняття	32
самостійна робота	101
Всього за дисципліну	165

7. Ознаки дисципліни:

Рік викладання	Семестр	Курс	Спеціальність	Нормативна/вибіркова
2020	3	2	126 «Інформаційні системи та технології»	Нормативна

8. Пререквізити

Студенти повинні володіти базовими знаннями з теорії алгоритмів та основ програмування. Освоєння дисципліни є однією із передумов для підготовки та захисту студентами кваліфікаційної роботи.

9. Технічне й програмне забезпечення /обладнання:

Студент повинен мати рівень впевненого користувача ОС Windows, середовища проектування Microsoft Visual Studio 2019 Community або Microsoft Visual Studio Code.

10. Політика дисципліни

Усі процедури навчального процесу під час викладання дисципліни відповідають положенню про академічну доброчесність учасників освітнього процесу та недопущення академічного плагіату в Тернопільському національному технічному університеті імені Івана Пулюя.

11. Схема дисципліни

Тиж./ дата/ год.	Тема, підтеми	Форма заняття, формат	Матеріали	Література, ресурси в інтернеті	Завдання, год	Засоби діагностики	Вага оцінки
1	2	3	4	5	6	7	8
Тиж. 1/ 2 акад. год.	Тема 1. Основи технологічної платформи MS .Net Framework. Причини появи та ключові характеристики платформи MS .Net Framework. Архітектура MS .Net Framework. Динамічна і статична складові. Структура бібліотеки класів FCL. Простір імен. Загальноомовне середовище виконання – динамічна складова MS. NET Framework. Універсальна система типів.	Лекція, F2F Лаб. роб. № 1	Лекційний матеріал, Лаб. роб. № 1	Дистанційний курс	Опрацювання лекцій, 3 лаб. роб., 3	Захист звіту з лаб. роб.	6
Тиж. 2-3/ 4 акад. год.	Тема 2. Базові засоби мови програмування C# (типи даних, вирази). Основні поняття C#. Система типів C#. Змінні, операції, вирази. Перетворення вбудованих арифметичних типів-значень. Перетворення з рядкового типу в арифметичний. Стиль програмування. Узгодження іменування.	Лекція, F2F	Лекційний матеріал	Дистанційний курс	Опрацювання лекцій, 5		
Тиж. 4/ 2 акад. год.	Тема 3. Базові засоби мови програмування C# (оператори). Умовні оператори. Оператори циклу. Оператори переходу. Базові конструкції структурного програмування.	Лекція, F2F Лаб. роб. № 2	Лекційний матеріал, Лаб. роб. № 2	Дистанційний курс	Опрацювання лекцій, 3 лаб. роб., 3	Захист звіту з лаб. роб.	6
Тиж. 5/ 2 акад. год.	Тема 4. Масиви і стандартні класи мови C#. Робота з масивами. Оператор циклу foreach. Клас Random і його методи. Клас Match і його методи.	Лекція, F2F Лаб. роб. № 3	Лекційний матеріал, Лаб. роб. № 3	Дистанційний курс	Опрацювання лекцій, 3 лаб. роб., 3	Захист звіту з лаб. роб.	6
Тиж. 6/ 2 акад. год.	Тема 5. Робота з рядками. Обробка помилок. Тип char. Масиви типу char. Робота з рядками. Клас String і його методи. Клас StringBuilder і його методи. Обробка помилок. Клас Exception. Комплексний приклад. Вдосконалений консольний калькулятор.	Лекція, F2F Лаб. роб. № 4	Лекційний матеріал, Лаб. роб. № 4	Дистанційний курс	Опрацювання лекцій, 3 лаб. роб., 3	Захист звіту з лаб. роб.	6

1	2	3	4	5	6	7	8
Тиж. 7/ 2 акад. год.	Тема 6. Робота з файлами. Введення-виведення на консоль. Файлове введення-виведення. Потoki. Абстрактний клас Stream і його нащадки. Класи StreamReader, StreamWriter, StringReader, StringWriter.	Лекція, F2F Лаб. роб. № 5	Лекційний матеріал, Лаб. роб. № 5	Дистанційний курс	Опрацювання лекцій, 3 лаб. роб., 3	Тестові завдання модуля 1, 13	Модульний контроль 1
Тиж. 8-9/ 4 акад. год.	Тема 7. Класи і об'єкти. Основні принципи ООП і їх реалізація в C#. Клас як модуль і клас як тип даних. Основні принципи ООП і їх реалізація в C#. Методи класу. Використання вказівника this в тілі класу. Конструктори. Типи конструкторів. Властивості get і set.	Лекція, F2F Лаб. роб. № 6	Лекційний матеріал, Лаб. роб. № 6	Дистанційний курс	Опрацювання лекцій, 5 лаб. роб., 3	Захист звіту з лаб. роб.	6
Тиж. 10-11/ 4 акад. год.	Тема 8. Реалізація принципу успадкування. Організація ієрархії класу. Використання в похідному класі конструктора базового класу з параметрами. Віртуальні методи. Пізніше і раннє зв'язування об'єктів класу. Абстрактні класи і методи. Приховані класи. Види відношень між класами. Клас object – базовий клас ієрархії класів C#.	Лекція, F2F Лаб. роб. № 7	Лекційний матеріал, Лаб. роб. № 7	Дистанційний курс	Опрацювання лекцій, 5 лаб. роб., 3	Захист звіту з лаб. роб.	6
Тиж. 12-13/ 4 акад. год.	Тема 9. Реалізація принципу поліморфізму. Перевантаження методів. Методи зі змінною кількістю аргументів. Індексатори. Перевантаження операцій. Приклад класу з перевантаженими методами і операціями.	Лекція, F2F Лаб. роб. № 8	Лекційний матеріал, Лаб. роб. № 8	Дистанційний курс	Опрацювання лекцій, 5 лаб. роб., 3	Захист звіту з лаб. роб.	6
Тиж. 14/ 2 акад. год.	Тема 10. Інтерфейси. Стратегії реалізації інтерфейсу. Інтерфейс як окремий випадок абстрактного класу. Стратегії реалізації інтерфейсів у C#. Інтерфейси і поля. Інтерфейси і успадкування.	Лекція, F2F Лаб. роб. № 9	Лекційний матеріал, Лаб. роб. № 9	Дистанційний курс	Опрацювання лекцій, 3 лаб. роб., 3	Захист звіту з лаб. роб.	6
Тиж. 15/ 2 акад. год.	Тема 11. Вбудовані інтерфейси C#. Впорядкованість об'єктів і інтерфейс IComparable. Впорядкування об'єктів за кількома критеріями. Інтерфейс IComparer. Перелічуваність об'єктів і інтерфейси. Інтерфейс IEnumerable. Ітератори і інтерфейс IEnumerable.	Лекція, F2F Лаб. роб. № 10	Лекційний матеріал, Лаб. роб. № 10	Дистанційний курс	Опрацювання лекцій, 3 лаб. роб., 3	Захист звіту з лаб. роб.	6
Тиж. 16/ 2 акад. год.	Тема 12. Класи-колекції (Stack, Queue, ArrayList). Класи-прототипи. Абстрактні структури даних. Колекції. Простір імен System.Collections. Клас Stack (Стек). Клас Queue (Черга). Клас ArrayList (Динамічний масив). Класи-прототипи (класи з родовими параметрами).	Лекція, F2F	Лекційний матеріал	Дистанційний курс	Опрацювання лекцій, 3 Тестові завдання модуля 2, 14	Модульний контроль 2	8

12. Система оцінювання та вимоги

Форма підсумкового семестрового контролю – **екзамен**.

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

За виконання 10-ти лабораторних робіт студент отримує максимально 60 балів (10 * 6).

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).

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	Лаб. роб. №10	6		
Лекція 6	Лаб. роб. №5	6	Лекція 12				

До підсумкового семестрового контролю (складання екзамену) допускаються студенти, які протягом семестру виконали всі види навчальної роботи, успішно пройшли проміжні (модульні) контролю і набрали не менше 45 балів.

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

13. Навчально-методичне забезпечення

1. Електронний навчальний курс «Об'єктно-орієнтоване програмування» для студентів денної форми навчання спеціальності 126 «Інформаційні системи та технології».

2. Методичні вказівки щодо самостійної роботи студентів та модульного контролю знань з дисципліни «Об'єктно-орієнтоване програмування» / Укл.: Шмигер Г.П., Василенко Я.П. — Тернопіль : ТНТУ, 2016 — 21 с.

3. Методичні вказівки до виконання лабораторних робіт з дисципліни «Об'єктно-орієнтоване програмування» / Укл.: Шмигер Г.П., Василенко Я.П. — Тернопіль : ТНТУ, 2016 — 26 с.

4. Конспект лекцій з дисципліни «Об'єктно-орієнтоване програмування» / Укладачі: Шмигер Г.П., Василенко Я.П. — Тернопіль : ТНТУ, 2016 — 72 с.

14. Рекомендована література

Базова

1. Албахари Дж., Албахари Б. С# 6.0. Справочник. Полное описание языка. – М.: ООО "И.Д. Вильямс", 2018.

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

3. Леоненков А. Самоучитель UML. Эффективный инструмент моделирования информационных систем. - СПб.: БХВ-Петербург, 2001. - 304 с.

4. Павловская Т.А. С#. Программирование на языке высокого уровня. Учебник для вузов. – СПб.: Питер Пресс, 2007. – 432 с.

5. Павловская Т.А., Щупак Ю.А. С++. Объектно-ориентированное программирование: Практикум. – СПб.: Питер, 2006. – 265 с.: ил.
6. Пышкин Е.В. Основные концепции и механизмы объектно-ориентированного программирования. – СПб.: БХВ-Петербург, 2005. – 640 с.: ил.
7. Скит Дж. С# для профессионалов. Тонкости программирования. – М.: ООО "И.Д. Вильямс", 2017.
8. Троелсен Э. Язык программирования С# 2010 и платформа .NET 4. – М.: ООО "И.Д. Вильямс", 2011. – 856 с.: ил.
9. Фленов М.Е. Библия С#. 2-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2011. – 560 с.: ил. + CD-ROM
10. Шилдт, Г. С# 4.0: полное руководство.: Пер. с англ. – М.: ООО "И.Д. Вильямс", 2011. – 1056 с.: ил.

Допоміжна

1. Алексенко О.В. Технології програмування та створення програмних продуктів. Конспект лекцій. – Суми : Сумський державний університет, 2013. – 133 с.
2. Англо-український тлумачний словник з обчислювальної техніки, Інтернету, програмування. – К.: СофтПрес, 2006. – 8237с.
3. Ахо Альфред, В., Хопкрофт. Структуры данных и алгоритмы.: Пер.с англ.: М.: Издательский дом «Вильямс», 2003. – 384 с.: ил.
4. Брукс Ф. Как проектируются и создаются программные комплексы. Мифический человек-месяц. Очерки по системному программированию. – СПб.: Изд-во «Символ-Плюс», 2000. – 304 с.
5. Буч Г., Рамбо Д., Джекобсон А. Язык UML. Руководство пользователя. – М.: ДМК, 1998.
6. Вирт Н. Алгоритмы и структуры данных. – СПб: Невский диалект, 2001.
7. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. СПб.: Изд-во «Питер», 2007. – 366 с.
8. Дональд Кнут. Программирование как искусство // Лекции лауреатов премии Тьюринга. 1966—1985. — М.: Мир, 1993.
9. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. — М.: МЦНМО, 1999.
10. Косовский Н. К. Основы теории элементарных алгоритмов. — Л.: Изд-во Ленингр. ун-та, 1987.
11. Лавров С. С. Программирование. Математические основы, средства, теория. — СПб.: БХВ-Петербург, 2001.
12. Лаптев В. В. С++. Объектно-ориентированное программирование: Учебное пособие / В. В. Лаптев. – СПб.: Питер, 2008. – 464 с. – ISBN 978-5-91180-200-4.
13. Пол А. Объектно-ориентированное программирование на С++. –М.: Бином, 2006. – 443с.

15. Інформаційні ресурси

1. <http://dl.tntu.edu.ua> Електронні навчальні курси ТНТУ імені І. Пулюя.
2. Tutorialspoint. [Електронний ресурс]. – Режим доступу: <https://www.tutorialspoint.com/csharp/index.htm>.
3. METANIT.COM. Сайт о программировании. [Електронний ресурс]. – Режим доступу: <https://metanit.com/sharp/tutorial/1.1.php>
4. ProfessorWeb .NET & Web programing [Електронний ресурс]. – Режим доступу: <https://professorweb.ru/>
5. Алгоритми і методи. [Електронний ресурс]. Режим доступу: <http://algorist.manual.ru/>.