

Чернівецький національний університет імені Юрія Федьковича

(повне найменування закладу вищої освіти)

Факультет математики та інформатики

(назва факультету/навчально-наукового інституту)

Кафедра математичного моделювання

(назва кафедри, що забезпечує викладання)

“ЗАТВЕРДЖУЮ”

**Декан факультету
математики та інформатики**

Ольга МАРТИНЮК

“28” серпня 2026 року

РОБОЧА ПРОГРАМА

навчальної дисципліни

Основи об'єктно-орієнтованого програмування

(назва навчальної дисципліни)

обов'язкова

(вказати: обов'язкова)

Освітньо-професійна програма **Комп'ютерні науки та проектування програмних систем**

(назва програми)

Спеціальність **Ф3 Комп'ютерні науки**

(вказати: код, назва)

Галузь знань **Ф Інформаційні технології**

(вказати: шифр, назва)

Рівень вищої освіти **перший (бакалаврський)**

(вказати: перший (бакалаврський) / другий (магістерський) / третій (освітньо-науковий))

Факультет **математики та інформатики**

(назва факультету/ навчально-наукового інституту, на якому здійснюється підготовка фахівців за вказаною освітньою програмою)

Мова навчання **українська**

(вказати: на якій мові читається дисципліна)

Чернівці 2026 рік

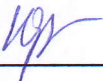
Робоча програма навчальної дисципліни «*Основи об'єктно-орієнтованого програмування*» складена відповідно до освітньо-професійної програми «Комп'ютерні науки та проектування програмних систем»

Розробник:

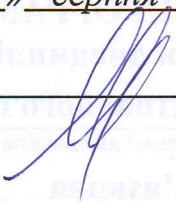
Дорош Андрій Богданович, асистент кафедри математичного моделювання, кандидат фіз.-мат. наук, доцент

Викладач, що забезпечує читання даної навчальної дисципліни:

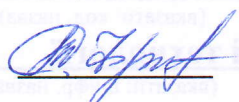
Дорош Андрій Богданович, асистент кафедри математичного моделювання, кандидат фіз.-мат. наук, доцент

Погоджено з гарантом ОП  **Ігор ЮРЧЕНКО**

Затверджено на засіданні кафедри математичного моделювання
Протокол № 1 від «27» серпня 2026 року

Завідувач кафедри  **Ігор ЧЕРЕВКО**

Схвалено методичною радою факультету математики та інформатики
Протокол № 1 від «28» серпня 2026 року

Голова методичної ради  **Тоня ФРАТАВЧАН**

Мета навчальної дисципліни

Мета вивчення навчальної дисципліни – оволодіти основними принципами об'єктно-орієнтованого підходу до програмування.

Завдання вивчення навчальної дисципліни

Після вивчення даної дисципліни студент повинен знати:

- основні принципи ООП;
- особливості ООП в мові програмування C++;
- особливості створення UML-діаграм;
- принципи SOLID.

Після вивчення даної дисципліни студент повинен вміти:

- створювати програмний код із використанням класів;
- реалізовувати принципи ООП мовою програмування C++;
- продумувати та реалізовувати ієрархію класів у проєктах;
- створювати UML-діаграми;
- використовувати принципи SOLID на практиці.

Пререквізити

Для вивчення даної освітньої компоненти будуть базою такі дисципліни: Програмування.

Результати навчання:

Під час вивчення дисципліни, відповідно до освітньо-професійної програми, формуються наступні

загальні компетентності:

ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК2. Здатність застосовувати знання у практичних ситуаціях.

ЗК3. Знання та розуміння предметної області та розуміння професійної діяльності.

ЗК6. Здатність вчитися й оволодівати сучасними знаннями.

ЗК7. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

ЗК8. Здатність генерувати нові ідеї (креативність).

ЗК9. Здатність працювати в команді.

ЗК11. Здатність приймати обґрунтовані рішення.

ЗК12. Здатність оцінювати та забезпечувати якість виконуваних робіт.

фахові компетентності:

ФК3. Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проєктування, 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.

ФК8. Здатність проєктувати 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.

програмні результати навчання:

ПРН1. Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук.

ПРН5. Проєктувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій.

ПРН9. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.

ПРН15. Застосовувати знання методології та CASE-засобів проєктування складних систем, методів структурного аналізу систем, об'єктно-орієнтованої методології проєктування при розробці і дослідженні функціональних моделей організаційно-економічних і виробничо-технічних систем.

Опис навчальної дисципліни

Загальна інформація

Форма навчання	Рік підготовки	Семестр	Кількість		Кількість годин						Вид підсумкового контролю
			кредитів	годин	лекції	практичні	семінарські	лабораторні	самостійна робота	індивідуальні завдання	
Денна	1	2	4	120	16	—	—	30	74	—	залік

Структура змісту навчальної дисципліни

Назви змістових модулів і тем	Кількість кредитів — 4 / годин — 120 год.											
	денна форма — 120 год.						заочна форма					
	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
Змістовий модуль 1												
Тема 1. Процедурне та об'єктно- орієнтоване програмування	16	2		5		9						
Тема 2. Поля і методи класів.	22	3		5		14						

Область видимості. Інкапсуляція												
Тема 3. Успадкування класів. UML-діаграми класів	22	3		5		14						
Разом за ЗМ 1	60	8		15		37						
Змістовий модуль 2												
Тема 4. Поліморфізм	17	3		5		9						
Тема 5. Абстрактні класи	21	2		5		14						
Тема 6. Принципи SOLID	22	3		5		14						
Разом за ЗМ 2	60	8		15		37						
Усього годин	120	16		30		74						
Підсумкова форма контролю	<i>залік</i>											

Теми лекційних занять

№ з/п	Назва теми	Кількість годин
1	Процедурне та об'єктно-орієнтоване програмування. Основні поняття ООП.	2
2	Поля і методи класів. Область видимості. Інкапсуляція.	3
3	Успадкування класів. UML-діаграми класів.	3
4	Поліморфізм.	3
5	Абстрактні класи.	2
6	Принципи SOLID.	2
	Разом	16

Теми лабораторних занять

№ з/п	Назва теми (завдання)	Кількість годин
1	Тема: Процедурне та об'єктно-орієнтоване програмування. Завдання: Написати невелику програму, яка описує об'єкти та процеси певної предметної області, згідно варіанту завдань, без і з використанням класів.	5

2	Тема: Поля і методи класів. Область видимості. Інкапсуляція. Завдання: Створити публічні та приватні поля й методи в класах, а також публічні функції доступу до них (зчитування та запису даних).	5
3	Тема: Успадкування класів. UML-діаграми класів. Завдання: Створити кілька дочірніх класів, які успадковують деякі поля та методи від батьківських. Побудувати UML-діаграму класів.	5
4	Тема: Поліморфізм. Завдання: Створити в дочірніх класах методи, які демонструють принцип роботи поліморфізму в ООП.	5
5	Тема: Абстрактні класи. Завдання: Створити абстрактний клас, із кількома класами-нащадками, які реалізують усі абстрактні (чисті віртуальні) методи.	5
6	Тема: Принципи SOLID. Завдання: Реалізувати ієрархію класів у проєкті таким чином, щоб виконувались принципи SOLID.	5
	Разом	30

Самостійна робота

Самостійна робота складає 74 години. Самостійна робота включає наступні види навчальних робіт:

- підготовка до лекційних занять;
- виконання лабораторних робіт;
- підготовка до заліку.

№ з/п	Назва теми	Кількість годин
1	Процедурне та об'єктно-орієнтоване програмування.	9
2	Поля і методи класів. Область видимості. Інкапсуляція.	14
3	Успадкування класів. UML-діаграми.	14
4	Поліморфізм.	9
5	Абстрактні класи.	14
6	Принципи SOLID.	14
	Разом	74

Контроль та оцінювання навчальних досягнень студентів у процесі вивчення навчальної дисципліни

Під час вивчення навчальної дисципліни використовуються такі види та методи контролю: *лабораторні роботи та тести.*

Перелік питань для самоконтролю й контролю навчальних досягнень студентів з навчальної дисципліни:

- 1) Чим відрізняються процедурний та об'єктно-орієнтований підходи програмування?
- 2) У чому суть інкапсуляції в ООП?
- 3) Що таке геттери і сеттери?
- 4) У чому суть успадкування в ООП?
- 5) Які бувають області видимості полів та методів класу?
- 6) У чому суть поліморфізму в ООП?
- 7) Які бувають види поліморфізму?
- 8) Для чого потрібні UML-діаграми?
- 9) У чому суть абстрактних класів?
- 10) Які є принципи SOLID?

Критерії оцінювання навчальних досягнень студентів за результатами поточного контролю

Система оцінювання рівня навчальних досягнень ґрунтується на принципах ECTS та є накопичувальною.

Протягом семестру студенти виконують 6 лабораторних робіт та 1 контрольне тестування. Лабораторні роботи (ЛР) оцінюються різною кількістю балів, контрольне тестування 10 балами, усього протягом семестру можна набрати 60 балів (див. таблицю нижче).

Поточне оцінювання (аудиторна та самостійна робота)							Разом за семестр	Залік	Сумарна к-ть балів
ЛР1	ЛР2	ЛР3	ЛР4	ЛР5	ЛР6	Підеумкове тестування		40	100
10	10	10	5	10	5	10	60		

Індивідуальна траєкторія здобувача вищої освіти у процесі вивченні навчальної дисципліни – участь у неформальній/інформальній освіті:

Якщо здобувач освіти отримав знання окрім формальної освіти у неформальній/інформальній освіті, зарахування результатів навчання здійснюється згідно положення «Про визнання результатів навчання, отриманих у формальній, нефо-

рмальній та/або інформальній освіті у Чернівецькому національному університеті імені Юрія Федьковича» та рішення кафедри у випадку, якщо тематика семінарів, вебінарів, воркшопів, тренінгів, у яких взяв участь студент підтверджується сертифікатом, програмою, а також відповідає змісту навчальної дисципліни (окремій темі або змістовому модулю).

Критерії оцінювання результатів навчання (з навчальної дисципліни) на підсумковому контролі

Критерієм підсумкового оцінювання є досягнення студентом мінімальних порогових рівнів оцінок (балів) за кожним запланованим результатом навчання навчальної дисципліни.

Підсумкова оцінка, як показник результатів вивчення навчальної дисципліни, складається із сумарної кількості балів за поточне оцінювання – **60 балів** та підсумкового модуль-контролю (залік) – **40 балів**, за **100-бальною університетською шкалою**.

Шкала оцінювання: національна та ЄКТС

Оцінка за національною шкалою	Оцінка за шкалою ECTS	
	Оцінка (бали)	Пояснення за розширеною шкалою
Зараховано	A (90-100)	відмінно
	B (80-89)	дуже добре
	C (70-79)	добре
	D (60-69)	задовільно
	E (50-59)	достатньо
Незараховано	FX (35-49)	(незараховано) з можливістю повторного складання
	F (1-34)	(незадовільно) з обов'язковим самостійним опрацюванням освітнього компоненту до перескладання

Методи та освітні технології навчання

Лекції, лабораторні роботи, тестування, аудиторне та онлайн-навчання з використанням системи Moodle та платформ для проведення онлайн-зустрічей.

Методи навчання:

- вербальні методи (лекція, бесіда, диспут, пояснення, розповідь тощо);
- практичні методи (лабораторні роботи);
- наочні методи (демонстрація, ілюстрація);
- робота з інформаційними ресурсами: з навчально-методичною, науковою, нормативною літературою та інтернет-ресурсами;
- самостійна робота над індивідуальним завданням або за програмою навчальної дисципліни;
- навчання з використанням відповідних онлайн-платформ.

Рекомендована література

Основна

1. Мартін Р. Чиста архітектура: Мистецтво створення програмного забезпечення. Видання друге / пер. з англ. І. Боднар-Терещенко. — Харків : Вид-во «Ранок» : Фабула, 2023. — 368 с.
2. Ярошко Сергій, Ярошко Оксана. Методи розробки алгоритмів. Програмування мовою C++: навч. посібник / С.А.Ярошко, О.С.Ярошко // Львів: ЛНУ імені Івана Франка. — 2022. — 248с.
3. Бублик В.В. Об'єктно-орієнтоване програмування: [Підручник] / В.В. Бублик. // К.: ІТ-книга, 2015. — 624 с.

Додаткова

1. Stroustrup B. A Tour of C++. Addison-Wesley. (3rd edition) Softcover and electronic versions, 2022. 190 p.
2. Грицюк Ю.І. Об'єктно-орієнтоване програмування мовою C++ : навчальний посібник. / Ю.І. Грицюк, Т.Є. Рак. — Львів : Вид-во Львівського ДУ БЖД, 2011. — 404 с.

Інформаційні ресурси

1. Онлайн-довідник C++ [Електронний ресурс]. — Режим доступу: <https://en.cppreference.com>
2. Документація C++ від Microsoft [Електронний ресурс]. — Режим доступу: <https://learn.microsoft.com/uk-ua/cpp/?view=msvc-170&viewFallbackFrom=vs-2019>
3. Уроки C++ [Електронний ресурс]. — Режим доступу: <https://acode.com.ua/uroki-po-cpp>
4. Електронний курс у системі Moodle: <https://moodle.chnu.edu.ua/course/view.php?id=9488>

Політика академічної доброчесності

Дотримання політики щодо академічної доброчесності учасниками освітнього процесу при вивченні освітнього компонента регламентовано чинними документами ЧНУ:

- "Кодекс академічної доброчесності та етики Чернівецького національного університету імені Юрія Федьковича" <https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/kodeks-akademichnoi-dobrochesnosti-ta-etyky-chernivetskoho-natsionalnoho-universytetu-imeni-yurii-fedkovycha/>

- "Положення про виявлення та запобігання академічного плагіату у Чернівецькому національному університеті імені Юрія Федьковича"
<https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/polozhennia-pro-vyivlennia-ta-zapobihannia-akademichnomu-plahiatu-u-chernivetskomu-natsionalnomu-universyteti-imeni-yuriia-fedkovycha/>
- "Політика використання штучного інтелекту в Чернівецькому національному університеті імені Юрія Федьковича"
<https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/polityka-vykorystannia-shtuchnoho-intelektu-v-chernivetskomu-natsionalnomu-universyteti-imeni-yuriia-fedkovycha-zi-zminamy-zatverdzhenymy-vchenoiu-radoiu-chnu-imeni-yuriia-fedkovycha/>

Використання здобувачами освіти інструментів штучного інтелекту (ШІ) під час виконання навчальних завдань:

допускається за умови дотримання принципів академічної доброчесності (здобувач освіти несе відповідальність за зміст, достовірність і результати поданої роботи, а у разі використання ШІ зобов'язаний зазначити факт і спосіб його застосування).