Чернівецький національний університет імені Юрія Федьковича

Навчально-науковий інститут фізико-технічних та комп'ютерних наук

Кафедра комп'ютерних систем та мереж

"ЗАТВЕРДЖУЮ"
Директор навчально-наукового інституту
фізико-технічних та комп'ютерних наук
_____ **Петро ШПАТАР**
"31" серпня 2026 року

РОБОЧА ПРОГРАМА
навчальної дисципліни

ОК3 Технології проектування програмних систем
(назва навчальної дисципліни)

обов'язкова
(вказати: обов'язкова / вибіркова)

Освітньо-професійна програма Комп'ютерна інженерія
(назва програми)

Спеціальність F7 Комп'ютерна інженерія
(вказати: код, назва)

Галузь знань F Інформаційні технології
(вказати: шифр, назва)

Рівень вищої освіти другий (магістерський)
(вказати: перший бакалаврський/другий магістерський)

фізико-технічних та комп'ютерних наук
(назва факультету/ навчально-наукового інституту,
на якому здійснюється підготовка фахівців за вказаною освітньо-професійною програмою)

Мова навчання українська

Чернівці 2026 рік

Робоча програма навчальної дисципліни «Технології проектування програмних систем» складена відповідно до освітньо-професійної програми «Комп'ютерна інженерія».

Розробник: Двірничук Костянтин Васильович, доцент кафедри КСМ, PhD (канд. фіз.-мат. наук), доцент
(П.І.Б. авторів, посада, науковий ступінь, вчене звання)

Викладач (чі), що забезпечує читання даної навчальної дисципліни: Двірничук Костянтин Васильович, доцент кафедри КСМ, PhD (канд. фіз.-мат. наук), доцент
(П.І.Б. викладача (ів), посада, науковий ступінь, вчене звання)

Погоджено з гарантом ОП _____ **Сергій БАЛОВСЯК**
(підпис)

Затверджено на засіданні кафедри *комп'ютерних систем та мереж*

Протокол № 2 від "31" серпня 2026 року

Завідувач кафедри _____ **Георгій ВОРОБЕЦЬ**
(підпис)

Схвалено методичною радою навчально-наукового інституту фізико-технічних та комп'ютерних наук

Протокол № 1/2 від "31" серпня 2026 року

Голова методичної ради навчально-наукового інституту фізико-технічних та комп'ютерних наук

_____ **Іван КОЗЯРСЬКИЙ**
(підпис)

1. Мета навчальної дисципліни

Мета: надати студентам систематизовані знання про мету, завдання та принципи технологій проектування програмних систем. Навчити студентів використовувати шаблони проектування (патерни) в своїх програмних розробках. Пояснити студентам суть архітектурних шаблонів та більш детально розібратися з MVC-програмуванням. Надати можливість студентам освоїти деякі сучасні технології та надати їм поняття та підходи тестування та відлагодження програм.

Вивчення даної дисципліни надає студентам ряд переваг, оскільки сучасне програмування не можна представити собі без проектування структури самої програми на основі архітектурних шаблонів, патернів проектування. Зокрема одним із популярних рішень архітектурного шаблону є MVC-шаблон, основи якого і подаються в рамках цієї дисципліни.

Пререквізити

Для коректного розуміння і засвоєння матеріалу курсу здобувачі освіти повинні попередньо опанувати дисципліни: Програмування, Об’єктно-орієнтоване програмування, Інженерія програмного забезпечення.

2. Результати навчання

Вивчення навчальної дисципліни сприяє формуванню у здобувачів вищої освіти таких компетентностей:

ЗК - загальних

ЗК2. Здатність до абстрактного мислення, аналізу і синтезу.

ЗК4. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

ЗК6. Здатність виявляти, ставити та вирішувати проблеми.

ЗК7. Здатність приймати обґрунтовані рішення.

ЗК8. Здатність спілкуватися іноземною мовою.

СК – фахових (спеціальних)

СК1. Здатність до визначення технічних характеристик, конструктивних особливостей, застосування і експлуатації програмних, програмно-технічних засобів, комп’ютерних систем та мереж різного призначення.

СК5. Здатність будувати архітектуру та створювати системне і прикладне програмне забезпечення комп’ютерних систем та мереж.

РН - програмних результатів навчання

РН11. Приймати ефективні рішення з питань розроблення, впровадження та експлуатації комп’ютерних систем і мереж, аналізувати альтернативи, оцінювати ризики та імовірні наслідки рішень.

РН12. Вільно спілкуватись усно і письмово українською мовою та однією з іноземних мов (англійською, німецькою, італійською, французькою, іспанською) при обговоренні професійних питань, досліджень та інновацій в галузі інформаційних технологій.

РН13. Зрозуміло і недвозначно доносити власні знання, висновки та аргументацію з питань інформаційних технологій і дотичних міжгалузевих питань до фахівців і нефахівців, зокрема до осіб, які навчаються.

У результаті вивчення навчальної дисципліни здобувач повинен продемонструвати такі результати навчання:

Знати: технології програмування, патерни проектування, будову MVC-моделей програм, методи тестування програмних розробок.

Вміти: будувати програмні додатки, використовувати для цього шаблони проектування та архітектурні шаблони, проводити тестування програмних додатків.

3. Опис навчальної дисципліни

3.1. Загальна інформація

Форма навчання	Рік підготовки	Семестр	Кількість		Кількість годин						Вид підсумкового контролю
			кредитів	всього годин	лекції	практичні	семінарські	лабораторні	самостійна робота	індивідуальні завдання	
Денна	1(5)	1(9)	3	90	14	-	-	16	60	-	Іспит
Заочна	1(5)	1(9)	3	90	4	-	-	4	82	-	Іспит

3.2. Структура змісту навчальної дисципліни

Назви змістових модулів і тем	Кількість годин											
	Денна форма						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. Технології проектування програмних систем												
Тема 1. Поняття архітектури ПЗ	31	3	-	8	-	20	30	1	-	2	-	27
Тема 2. Шаблони проектування програмного забезпечення	31	7	-	4	-	20	30	2	-	1	-	27

Тема 3. Тестування та відлагодження програмних систем	28	4	-	4	-	20	30	1	-	1	-	28
Разом за змістовим модулем 1	90	14	-	16	-	60	90	4	-	4	-	82
Усього годин	90	14	-	16	-	60	90	4	-	4	-	82

Тематика лекційних занять з переліком питань

№	Назва теми з основними питаннями
1	**Поняття архітектури ПЗ.** Визначення архітектури програмного забезпечення, її призначення та роль у життєвому циклі розроблення. Архітектурні рівні, компоненти та зв'язки між ними. Архітектурні стилі та шаблони. Шаблон MVC: модель, подання, контролер. Взаємодія компонентів MVC, маршрутизація запитів. Особливості реалізації MVC-додатків.
2	**Шаблони проектування програмного забезпечення.** Поняття шаблону (патерну) проектування, історія виникнення, класифікація шаблонів. Принципи SOLID. Породжуючі, структурні та поведінкові шаблони. Структурні шаблони: Adapter, Bridge, Composite, Decorator, Facade, Flyweight, Proxy. Поведінкові шаблони: Strategy, Observer, Command, State, Template Method, Iterator, Mediator, Memento, Visitor, Chain of Responsibility. Вибір шаблону для розв'язання типових задач проектування.
3	**Тестування та відлагодження програмних систем.** Мета і принципи тестування. Рівні та види тестування: модульне, інтеграційне, системне, приймальне. Методи «чорної» та «білої» скриньки. Розроблення тестових сценаріїв і наборів тестів. Модульне тестування, імітація залежностей (mock-об'єкти). Розроблення через тестування (TDD). Відлагодження програм: точки зупину, трасування, аналіз журналів. Покриття коду тестами.

3.3. Тематика лабораторних занять

№	Назва теми (завдання)	Кількість годин
1	Аналіз архітектурного шаблону ПЗ	4
2	Розробка MVC-додатку	4
3	Шаблони (патерни) проектування ПЗ	4
4	Тестування ПЗ	4
	Разом	16

3.4. Самостійна робота студента

№	Назва теми/ кількість балів/ форма контролю	Кількість годин
1	CMF та CMS / результати використовуються при виконанні лабораторних робіт № 1-2 (20 балів)	10
2	Розробка HMVC-, MVVM-, NA-, MVP-, PAC-моделей програм / результати використовуються при виконанні лабораторних робіт № 1-2 (20 балів)	10
3	Антипатерни / результати використовуються при виконанні лабораторної роботи № 3 (15 балів)	10
4	Породжуючі патерни / результати використовуються при виконанні лабораторної роботи № 3 (15 балів)	10
5	Класифікація та проведення тестування / результати використовуються при виконанні лабораторної роботи № 4 (15 балів)	20
	Разом	60

4. Форми і методи навчання

Основними формами організації навчання є лекційні та лабораторні заняття, а також самостійна робота здобувачів вищої освіти. Самостійна робота спрямована на опрацювання матеріалів лекцій та літературних джерел.

У процесі вивчення дисципліни застосовуються такі методи навчання:

- Пояснювально-ілюстративний метод – виклад теоретичного матеріалу з використанням прикладів схем, діаграм, таблиць і мультимедійних презентацій для пояснення основних понять, принципів і методологій.
- Демонстраційний метод – показ прикладів проектування та реалізації програмних систем із застосуванням архітектурних шаблонів і шаблонів проектування.
- Практичний метод – виконання лабораторних завдань, спрямованих на формування навичок розроблення програмних додатків, застосування шаблонів проектування та тестування програмного забезпечення.
- Проблемно-пошуковий метод – постановка практичних проблем і ситуацій, пов’язаних з проектуванням, розробленням і тестуванням програмних систем.
- Аналіз практичних ситуацій – розгляд прикладів побудови програмних систем для вирішення прикладних завдань із застосуванням MVC-архітектури та шаблонів проектування.
- Обговорення та навчальні дискусії – колективний аналіз проблемних питань, обговорення результатів виконання лабораторних робіт, порівняння можливих архітектурних рішень і шаблонів проектування.
- Презентація результатів виконаних завдань – представлення та захист розроблених програмних додатків, обґрунтування прийнятих рішень і формулювання висновків.

Застосовуються інноваційні освітні технології, зокрема, інформаційно-комунікаційні та студентоцентрованого навчання. Освітні технології передбачають використання мультимедійних матеріалів для лекційних занять, методичних вказівок для виконання лабораторних робіт, мов та середовищ розробки комп'ютерних програм, комп'ютерної та телекомунікаційної техніки, відеоконференцій, системи електронного навчання Moodle з метою розвитку навичок проектування, розроблення та тестування програмних систем.

5. Система контролю та оцінювання

Система контролю та оцінювання передбачає перевірку рівня засвоєння теоретичних знань, сформованості практичних умінь і здатності застосовувати цифрові інструменти для виконання навчальних завдань. Оцінювання здійснюється систематично протягом семестру з урахуванням своєчасності, самостійності, правильності та якості виконаних робіт.

Форми та методи контролю

***Поточний контроль*:** *усне опитування* студента на лекційних заняттях (фронтальне, індивідуальне); *письмовий контроль* (звіти із виконання лабораторних робіт); *практичний контроль* (захист лабораторних робіт); *тестовий/машинний контроль* (тести у системі Moodle).

Рубіжний контроль: письмова контрольна робота у кінці змістового модуля №1.

Підсумковий контроль – іспит.

6. Критерії оцінювання поточного та підсумкового контролю

Поточне оцінювання (аудиторна та самостійна робота)				**Підсумковий контроль (іспит)**	**Сумарна кількість балів**
Змістовий модуль 1					
Т1	Т2	Т3	М		
20	15	15	10	40	100

Т1, Т2, Т3 – теми змістового модуля; М – модульна контрольна робота

Залежно від характеру відповіді здобувача освіти зазначена кількість балів може бути скоригована за наступними критеріями:

К-ть балів	**Критерії оцінки**
max	здобувач освіти дає вичерпну відповідь на поставлене запитання;
0,8 · max	здобувач освіти при відповіді на поставлене запитання припустився незначних неточностей, які не впливають на суть відповіді;

0,6 · max	здобувач освіти при відповіді на поставлене запитання припустився помилок, які виправляє за допомогою викладача; в середньому може дати правильні відповіді на 50% питань теми;
0,4 · max	здобувач освіти при відповіді на поставлене запитання припустився суттєвих помилок, які все ж таки виправляє за допомогою викладача; дає правильні відповіді на 30% питань теми;
0,2 · max	отримує здобувач освіти, який за допомогою викладача фрагментарно відповідає на запитання, проте не в повній мірі володіє мінімальним рівнем знань з даного питання;
0	якщо характер відповідей дає підставу стверджувати, що здобувач освіти неправильно зрозумів суть питання чи не знав правильної відповіді, а тому відповідав, припускаючись грубих помилок.

Примітка: за max прийнято максимальну оцінку для даного виду діяльності. Заокруглення проводиться до одиниць балу.

Згідно шкали ECTS загальна кількість балів, яку здобувач освіти може отримати у процесі вивчення дисципліни, становить 100 балів, з яких 60 балів здобувач освіти набирає при поточних видах контролю і 40 балів – у процесі підсумкового контролю (іспит).

Шкала оцінювання: національна та ЄКТС

Оцінка за національною шкалою	**Оцінка за шкалою ECTS**	
	Оцінка (бали)	**Пояснення за розширеною шкалою**
Відмінно	A (90-100)	відмінно
Добре	B (80-89)	дуже добре
	C (70-79)	добре
Задовільно	D (60-69)	задовільно
	E (50-59)	достатньо
Незадовільно	FX (35-49)	(незадовільно) з можливістю повторного складання
	F (1-34)	(незадовільно) з обов'язковим повторним курсом

7. Перелік питань для самоконтролю та підсумкового контролю навчальних досягнень здобувачів освіти

Змістовий модуль 1. Технології проектування програмних систем

1. Дати визначення поняття архітектури програмного забезпечення.
2. Описати завдання та роль архітектури у процесі розроблення програмних систем.
3. Що таке архітектурний шаблон? Навести приклади архітектурних шаблонів.
4. Описати будову шаблону MVC: призначення моделі, подання та контролера.

5. Як відбувається взаємодія компонентів у MVC-додатку? Описати потік обробки запиту.
6. Які переваги та недоліки використання шаблону MVC?
7. Описати особливості моделей HMVC, MVVM, MVP та PAC.
8. У чому полягає відмінність між шаблонами MVC та MVVM?
9. Що таке CMS і CMF? Чим вони відрізняються?
10. За якими критеріями обирається архітектура програмного додатку?
11. Що таке шаблон (патерн) проектування? Як класифікуються шаблони проектування?
12. Описати принципи SOLID.
13. Які породжуючі шаблони проектування існують? Описати їх призначення.
14. Описати шаблон Singleton та його недоліки.
15. У чому полягає різниця між шаблонами Factory Method та Abstract Factory?
16. Описати структурні шаблони Adapter, Decorator, Facade, Proxy, Composite.
17. У чому полягає відмінність між шаблонами Adapter та Decorator?
18. Описати поведінкові шаблони Strategy, Observer, Command, State, Template Method.
19. Описати шаблон Observer та навести приклад його використання.
20. Що таке антипатерни проектування? Навести приклади антипатернів.
21. Яким чином шаблони проектування сприяють повторному використанню коду та супроводженню програм?
22. Які мета та принципи тестування програмного забезпечення?
23. Описати рівні тестування: модульне, інтеграційне, системне, приймальне.
24. У чому полягає різниця між методами тестування «чорної» та «білої» скриньки?
25. Як розробляються тестові сценарії та набори тестів?
26. Що таке модульне тестування? Які засоби модульного тестування використовуються?
27. Що таке імітація залежностей (mock-об’єкти) і для чого вона використовується?
28. Описати методологію розроблення через тестування (TDD).
29. Що таке покриття коду тестами? Які метрики покриття існують?
30. Описати засоби відлагодження програм: точки зупину, покрокове виконання, журналювання.
31. Які існують види нефункціонального тестування (продуктивності, навантажувальне, безпеки)?

8. Рекомендована література

Фахова (основна)

1. Двірничук К.В. Конспект лекцій з навчальної дисципліни «Технології проектування програмних систем» (електронне видання). Чернівці : ЧНУ, 2022. 100 с.

2. Трофименко О. Г., Дика А. І. Тестування та забезпечення якості програмних систем : навчальний посібник. Одеса : Фенікс, 2024. 195 с. URL: https://hdl.handle.net/11300/27717.
3. Marcotte C.-H. Architecting ASP.NET Core Applications : An atypical design patterns guide for .NET 8, C# 12, and beyond. 3rd ed. Birmingham : Packt Publishing, 2024. 806 p.
4. Baptista G., Abbruzzese F. Software Architecture with C# 12 and .NET 8 : Build enterprise applications using microservices, DevOps, EF Core, and design patterns for Azure. 4th ed. Birmingham : Packt Publishing, 2024. 756 p.
5. Richards M., Ford N. Fundamentals of Software Architecture : A Modern Engineering Approach. 2nd ed. Sebastopol : O'Reilly Media, 2025. 550 p.
6. Esposito D. Clean Architecture with .NET. Hoboken : Microsoft Press, 2024. 336 p.
7. Патерни проєктування. URL: https://abitap.com/category/paterny-proektuvannya/ (дата звернення: 16.11.2026).
8. Мова програмування C#. URL: https://abitap.com/category/c/ (дата звернення: 16.11.2026).
9. ASP.NET Core (MVC, Web API). URL: https://abitap.com/category/asp-net-core/ (дата звернення: 16.11.2026)

Допоміжна

10. Улічев О. С., Папіж Л. М., Ревнюк О. А. Досягнення в тестуванні програмного забезпечення. Наукова перспектива. Центральноукраїнський науковий вісник. Технічні науки. 2024. Вип. 10 (41), ч. І. С. 3–16. URL: https://centralukraine-tech.kntu.kr.ua/tech/article/view/337 (дата звернення: 16.11.2026).
11. Улічев О. С., Доренський О. П., Кулагін В. П. Інноваційні рішення та переваги мікросервісної архітектури програмних продуктів. Центральноукраїнський науковий вісник. Технічні науки. 2024. Вип. 10 (41). С. 16–29. DOI: https://doi.org/10.32515/2664-262X.2024.10(41).16-29.
12. Соколова Н. О., Лисун Ю. Р., Гаркуша І. М., Балалаєва О. Ю. Розробка адаптивного вебдизайну на основі патернів проєктування. Комп'ютерно-інтегровані технології: освіта, наука, виробництво. 2025. Вип. 60. С. 300–311. DOI: https://doi.org/10.36910/6775-2524-0560-2025-60-32.
13. Overview of ASP.NET Core MVC. Microsoft Learn. URL: https://learn.microsoft.com/aspnet/core/mvc/overview (дата звернення: 16.11.2026).
14. Unit testing best practices for .NET. Microsoft Learn. URL: https://learn.microsoft.com/dotnet/core/testing/unit-testing-best-practices (дата звернення: 16.11.2026).

9. Інформаційні ресурси

1. Навчальна платформа Moodle. URL: https://moodle.chnu.edu.ua/course/view.php?id=3145.
2. Освітні програми кафедри КСМ. URL: https://csn.chnu.edu.ua/about-us/ok-rivni/
3. ОПП "Комп'ютерна інженерія" (магістратура). URL: https://csn.chnu.edu.ua/spetsialnist-123-komp-yuterna-inzheneriya-opp-komp-yuterna-inzheneriya-magistratura-1-5-r/

10. Політика академічної доброчесності

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

- «Етичний кодекс Чернівецького національного університету імені Юрія Федьковича» https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/etychnyi-kodeks-chernivetskoho-natsionalnoho-universytetu-imeni-yuriia-fedkovycha/
- «Положенням про виявлення та запобігання академічного плагіату у Чернівецькому національному університеті імені Юрія Федьковича» https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/polozhennia-pro-vyiavlennia-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/

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);
- не допускається використання ШІ як заміни самостійного виконання завдань або з метою порушення вимог академічної доброчесності;
- додаткові обмеження чи вимоги щодо використання ШІ можуть визначатися викладачем в освітньому процесі для конкретних видів навчальної діяльності відповідно до особливостей дисципліни.