МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
МІЖНАРОДНИЙ ГУМАНІТАРНИЙ УНІВЕРСИТЕТ
Кафедра інформаційних технологій

ПАТЕРНИ ТА ФРЕЙМВОРКИ

Стрелковська І.В., Приходько С.Б., Григор'єва Т.І.

Методичні рекомендації для самостійної роботи здобувачів

Одеса 2025

Затверджено Вченою Радою Міжнародного гуманітарного університету
(протокол № 5 від 20.01.2025 р.)

Стрелковська І.В., Приходько С.Б., Григор'єва Т.І.
Патерни та фреймворки: методичні рекомендації для самостійної роботи здобувачів [Електронне видання]. / Стрелковська І.В., Приходько С.Б., Григор'єва Т.І. Кафедра інформаційних технологій Міжнародного гуманітарного університету. Одеса, 2025. – 32 с.

Методичні рекомендації з курсу «Патерни та фреймворки» розроблено відповідно до навчального плану, вони складаються з навчальної програми курсу, методичних рекомендацій з проведення практичних занять і завдань для самостійної роботи здобувачів, списку рекомендованої літератури. Матеріали призначено для студентів факультету кібербезпеки, програмної інженерії та комп'ютерних наук Міжнародного гуманітарного університету, які в бакалавраті вивчають галузь знань «Інформаційні технології», спеціальність «Електроніка, електронні комунікації, приладобудування та радіотехніка», спеціальність «Середня освіта (Інформатика та програмування)».

ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

<table>
<tr><td>Найменування показників</td><td>Галузь знань, напрям підготовки, освітньо-кваліфікаційний рівень</td><td colspan="2">Характеристика навчальної дисципліни</td></tr>
<tr><td rowspan="10">Кількість кредитів – 12,
загальна кількість
годин – 360</td><td rowspan="10">Галузь – 12 «Інформаційні технології»
Спеціальності:
121 - Інженерія програмного забезпечення
122 - Комп'ютерні науки
125 - Кібербезпека та захист інформації
123 - Комп'ютерна інженерія
Галузь – 17 Інженерія, виробництво та будівництво
Спеціальність: 172 Електроніка, електронні комунікації, приладобудування та радіотехніка
Галузь – 01 «Освіта / Педагогіка»
Спеціальність:
014 - Середня освіта (Інформатика та програмування)</td><td>Денна форма</td><td>Заочна форма</td></tr>
<tr><td colspan="2">обов’язкова</td></tr>
<tr><td colspan="2">Рік підготовки:</td></tr>
<tr><td colspan="2">4</td></tr>
<tr><td colspan="2">Семестр:</td></tr>
<tr><td colspan="2">7, 8</td></tr>
<tr><td colspan="2">Лекційні заняття:</td></tr>
<tr><td>80</td><td>24</td></tr>
<tr><td colspan="2">Практичні заняття:</td></tr>
<tr><td>78</td><td>24</td></tr>
<tr><td rowspan="10">Мова навчання – українська</td><td rowspan="10">Рівень вищої освіти – перший (бакалаврський)</td><td colspan="2">Лабораторні роботи</td></tr>
<tr><td>0</td><td>0</td></tr>
<tr><td colspan="2">Самостійна робота</td></tr>
<tr><td>202</td><td>312</td></tr>
<tr><td colspan="2">Вид контролю:</td></tr>
<tr><td colspan="2">залік, екзамен</td></tr>
</table>

Дисципліна «Патерни та фреймворки» входить до завершального етапу підготовки здобувачів вищої освіти за освітньою програмою «Інженерія програмного забезпечення». Вона спрямована на формування системного уявлення

про сучасні підходи до проєктування та реалізації програмних систем. У межах курсу розглядаються принципи повторного використання програмних рішень, патерни проєктування програмного забезпечення та типові архітектури програмних систем. Значна увага приділяється аналізу породжувальних, структурних і поведінкових патернів, їх призначенню та особливостям застосування під час створення гнучких і модульних програмних систем. Окремий розділ дисципліни присвячений використанню фреймворків під час створення програмного забезпечення. Розглядаються найбільш поширені типи фреймворків, зокрема сучасні фреймворки для розробки застосунків для персональних комп'ютерів, мобільних та вебзастосунків. Особлива увага приділяється принципам їх вибору та практичного використання.

МЕТА ДИСЦИПЛІНИ. Метою дисципліни є формування у здобувачів знань, умінь і навичок використання патернів проєктування та сучасних фреймворків під час створення програмного забезпечення, аналізу архітектурних рішень програмних систем та застосування типових програмних рішень у професійній діяльності.

ПРЕРЕКВІЗИТИ. Передумовами для вивчення дисципліни є знання і вміння, отримані здобувачем при вивченні попередніх навчальних дисциплін бакалаврської підготовки: «Алгоритмізація та програмування», «Алгоритми та структури даних», «Об'єктно-орієнтоване програмування», «Організація баз даних та знань», «Вебтехнології та вебдизайн».

ПОСТРЕКВІЗИТИ. Знання і вміння, отримані здобувачем при вивченні даної навчальної дисципліни, можуть бути використані під час паралельного вивчення дисципліни «Програмування інфокомунікаційних сервісів та систем», а також при виконанні кваліфікаційної роботи.

ЗАПЛАНОВАНІ РЕЗУЛЬТАТИ НАВЧАННЯ

ЗА НАВЧАЛЬНОЮ ДИСЦИПЛІНОЮ

Знання:

– роль патернів і фреймворків на різних етапах життєвого циклу програмного забезпечення;

– класифікацію патернів програмного забезпечення та особливості породжувальних, структурних і поведінкових патернів;

– типові архітектури програмного забезпечення та їх призначення у побудові програмних систем;

– принципи повторного використання програмних рішень у програмній інженерії;

– призначення та особливості використання фреймворків у розробці програмного забезпечення;

– типи фреймворків для створення програмних систем різного призначення;

– архітектурні особливості фреймворків для розробки застосунків для персональних комп’ютерів;

– принципи використання фреймворків для розробки мобільних застосунків;

– особливості використання фреймворків для розробки вебзастосунків.

Уміння:

– систематизувати та аналізувати сучасні інструменти і підходи розробки програмного забезпечення;

– аналізувати архітектуру програмної системи та визначати доцільність використання відповідних патернів;

– застосовувати патерни проєктування під час розробки програмного забезпечення;

– обирати відповідні патерни для вирішення типових задач програмування;

– виконувати проєктування програмних модулів із використанням типових програмних рішень;

– обирати відповідний фреймворк для розробки програмної системи залежно від її призначення;

– використовувати фреймворки для створення застосунків для персональних комп’ютерів;

– використовувати фреймворки для розробки мобільних застосунків;

– використовувати фреймворки для розробки вебзастосунків;

– інтегрувати фреймворки у програмні системи та використовувати їх функціональні можливості під час розробки програмного забезпечення.

Навички:

– застосування патернів проєктування під час створення програмних систем;

– використання сучасних фреймворків у процесі розробки програмного забезпечення;

– проєктування структури програмних систем із використанням типових архітектурних рішень;

– розроблення програмних застосунків із використанням фреймворків різного призначення;

– аналізу та вибору інструментальних засобів для створення програмних систем.

ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Семестр 1. Патерни проєктування програмного забезпечення

Тема 1. Патерни у програмному забезпеченні

Поняття патерну у програмному забезпеченні та його призначення. Історія виникнення підходу до використання патернів. Роль патернів у повторному використанні архітектурних рішень та підвищенні якості програмних систем.

Класифікація патернів програмного забезпечення. Архітектурні патерни, патерни проєктування та ідіоми програмування. Основні категорії патернів: породжувальні, структурні та поведінкові.

Структура опису патернів. Основні елементи опису патерну: назва, призначення, задача, рішення, структура, учасники, взаємодія компонентів.

Переваги застосування патернів у програмному забезпеченні. Обмеження використання патернів. Типові помилки застосування та поняття антипатернів.

Тема 2. Типові архітектури програмного забезпечення

Поняття архітектури програмного забезпечення та її роль у створенні програмних систем. Рівні абстракції програмного забезпечення. Взаємозв’язок архітектури та проєктування.

Архітектура Model–View–Controller. Призначення, структура та взаємодія компонентів. Приклади застосування MVC у програмних системах.

Багаторівнева архітектура програмного забезпечення. Архітектурні підходи Layered Architecture та Client–Server. Особливості організації компонентів у багаторівневих системах.

Архітектура Repository. Централізоване зберігання даних у програмних системах. Порівняння типових архітектур програмного забезпечення та принципи вибору архітектури системи.

Тема 3. Породжувальні патерни

Проблема створення об’єктів у програмних системах. Призначення породжувальних патернів. Загальна характеристика та сфери застосування.

Патерни Factory Method та Abstract Factory. Призначення, структура та приклади використання.

Патерни Builder та Prototype. Поетапне створення складних об’єктів. Використання механізму клонування об’єктів.

Патерн Singleton. Забезпечення існування єдиного екземпляра класу. Переваги та обмеження використання.

Тема 4. Структурні патерни

Призначення структурних патернів. Організація структури програмних систем та взаємодії між класами і компонентами.

Патерни Adapter та Bridge. Узгодження несумісних інтерфейсів. Відокремлення абстракції від реалізації.

Патерни Composite та Decorator. Побудова ієрархічних структур об’єктів. Розширення функціональності об’єктів без зміни базового коду.

Патерни Facade, Flyweight та Proxy. Спрощення взаємодії підсистем. Оптимізація використання ресурсів. Керування доступом до об’єктів.

Тема 5. Поведінкові патерни

Призначення поведінкових патернів. Розподіл відповідальності між компонентами програмної системи. Організація взаємодії об’єктів.

Патерни Observer, Mediator та Chain of Responsibility. Організація обміну повідомленнями між об’єктами.

Патерни Strategy, Command та State. Інкапсуляція алгоритмів. Керування поведінкою об’єктів.

Патерни Iterator, Template Method та Visitor. Організація обходу структур даних. Реалізація операцій над складними структурами об’єктів.

Семестр 2. Фреймворки для розробки програмного забезпечення

Тема 6. Використання фреймворків для створення програмного забезпечення

Поняття фреймворку та його призначення у процесі розробки програмного забезпечення. Відмінність між фреймворками, бібліотеками та програмними платформами.

Причини використання фреймворків під час створення програмних систем. Переваги застосування фреймворків: повторне використання коду, стандартизація архітектурних рішень, прискорення розробки.

Архітектурні особливості фреймворків. Поняття каркасу програмної системи, інверсії керування та розширення функціональності програм за допомогою фреймворків.

Переваги та обмеження використання фреймворків у розробці програмного забезпечення. Типові проблеми інтеграції фреймворків у програмні системи.

Тема 7. Типи фреймворків для розробки програмного забезпечення

Класифікація фреймворків програмного забезпечення. Критерії поділу фреймворків за сферою застосування та функціональним призначенням.

Вебфреймворки, фреймворки для застосунків для персональних комп'ютерів, мобільні фреймворки. Особливості використання кожного типу фреймворків.

Фреймворки для тестування програмного забезпечення та автоматизації розробки. Інструментальні фреймворки та засоби підтримки розробки.

Вибір фреймворку для створення програмної системи. Критерії вибору, оцінювання можливостей фреймворку та сумісність з іншими програмними компонентами.

Тема 8. Фреймворки для розробки застосунків для персональних комп'ютерів

Особливості розробки програмного забезпечення для персональних комп'ютерів. Архітектура застосунків та організація взаємодії з операційною системою.

Фреймворки для створення графічних інтерфейсів користувача. Компонентні моделі інтерфейсів та організація взаємодії елементів інтерфейсу.

Поширені фреймворки для розробки застосунків для персональних комп'ютерів: Qt, .NET, Electron. Архітектурні особливості та функціональні можливості цих фреймворків.

Інтеграція застосунків для персональних комп'ютерів із зовнішніми сервісами та мережевими ресурсами. Особливості створення клієнтських програмних систем.

Тема 9. Фреймворки для розробки мобільних застосунків

Особливості розробки мобільних застосунків. Архітектура мобільних програмних систем та взаємодія з мобільними платформами.

Фреймворки для створення мобільних застосунків. Підходи до розробки нативних та кросплатформних програм.

Поширені фреймворки для розробки мобільних застосунків: Flutter, React Native, Xamarin. Архітектурні особливості та сфери застосування.

Компонентна структура мобільних застосунків. Інтерфейс користувача, обробка подій та взаємодія з апаратними можливостями мобільних пристроїв.

Тема 10. Фреймворки для розробки вебзастосунків

Особливості архітектури вебзастосунків. Клієнтська та серверна частини вебсистем, взаємодія між компонентами.

Вебфреймворки для серверної частини програмних систем: Django, Flask, Spring. Організація обробки запитів, маршрутизація та взаємодія з базами даних.

Фреймворки клієнтської частини вебзастосунків: Angular, React, Vue. Організація інтерфейсу користувача та керування станом вебдодатків.

Порівняння сучасних вебфреймворків та підходів до побудови вебзастосунків.

СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Семестр 1. Патерни проєктування програмного забезпечення

Назви змістових модулів і тем	Кількість годин									
	Денна форма					Заочна форма				
	Заг.	у тому числі				Заг.	у тому числі			
		Лек.	Прак.	Лаб.	Сам. роб.		Лек.	Прак.	Лаб.	Сам. роб.
Тема 1. Патерни у програмному забезпеченні	36	8	8	0	20	36	2	2	0	32
Тема 2. Типові архітектури програмного забезпечення	36	8	8	0	20	36	4	4	0	28
Тема 3. Породжувальні патерни	36	8	8	0	20	36	2	2	0	32
Тема 4. Структурні патерни	36	8	8	0	20	36	2	2	0	32
Тема 5. Поведінкові патерни	36	8	6	0	22	36	2	2	0	32
Загалом	**180**	**40**	**38**	**0**	**102**	**180**	**12**	**12**	**0**	**156**
Підсумковий контроль – залік										

Семестр 2. Фреймворки для розробки програмного забезпечення

Назви змістових модулів і тем	Кількість годин									
	Денна форма					Заочна форма				
	Заг.	у тому числі				Заг.	у тому числі			
		Лек.	Прак.	Лаб.	Сам. роб.		Лек.	Прак.	Лаб.	Сам. роб.
Тема 6. Використання фреймворків для створення програмного забезпечення	36	8	8	0	20	36	2	2	0	32
Тема 7. Типи фреймворків для розробки програмного забезпечення	36	8	8	0	20	36	4	4	0	28
Тема 8. Фреймворки для розробки застосунків для персональних комп'ютерів	36	8	8	0	20	36	2	2	0	32
Тема 9. Фреймворки для розробки мобільних застосунків	36	8	8	0	20	36	2	2	0	32
Тема 10. Фреймворки для розробки вебзастосунків	36	8	8	0	20	36	2	2	0	32
Загалом	**180**	**40**	**40**	**0**	**100**	**180**	**12**	**12**	**0**	**156**
Підсумковий контроль – екзамен										

ПИТАННЯ ДО ПРАКТИЧНИХ ЗАНЯТЬ

Семестр 1. Патерни проєктування програмного забезпечення

Тема 1. Патерни у програмному забезпеченні

1. Аналіз поняття патерну у програмному забезпеченні.
2. Класифікація патернів програмного забезпечення.
3. Аналіз структури опису патерну.
4. Аналіз прикладів застосування патернів у програмних системах.

Тема 2. Типові архітектури програмного забезпечення

1. Аналіз архітектури Model–View–Controller.
2. Побудова структурної схеми програмної системи з використанням архітектури MVC.
3. Аналіз багаторівневої архітектури програмного забезпечення.

4. Порівняння типових архітектур програмного забезпечення.
Тема 3. Породжувальні патерни 1. Розробка застосунку на основі патерну Factory Method. 2. Розробка застосунку на основі патерну Singleton.
Тема 4. Структурні патерни 1. Розробка застосунку на основі патерну Adapter. 2. Розробка застосунку на основі патерну Decorator.
Тема 5. Поведінкові патерни 1. Розробка застосунку на основі патерну Observer. 2. Розробка застосунку на основі патерну Strategy.

Семестр 2. Фреймворки для розробки програмного забезпечення

Тема 6. Використання фреймворків для створення програмного забезпечення 1. Аналіз поняття фреймворку та його ролі у процесі розробки програмного забезпечення. 2. Порівняння фреймворків, бібліотек та програмних платформ. 3. Аналіз архітектурних особливостей сучасних фреймворків. 4. Аналіз прикладів використання фреймворків у програмних системах.
Тема 7. Типи фреймворків для розробки програмного забезпечення 1. Класифікація фреймворків за сферою застосування. 2. Аналіз фреймворків для розробки застосунків для персональних комп'ютерів. 3. Аналіз фреймворків для розробки мобільних застосунків. 4. Аналіз фреймворків для розробки вебзастосунків.
Тема 8. Фреймворки для розробки застосунків для персональних комп'ютерів 1. Розробка застосунку за допомогою фреймворку Qt. 2. Розробка застосунку за допомогою фреймворку .NET.
Тема 9. Фреймворки для розробки мобільних застосунків 1. Розробка застосунку за допомогою фреймворку Flutter. 2. Розробка застосунку за допомогою фреймворку React Native.
Тема 10. Фреймворки для розробки вебзастосунків 1. Розробка застосунку за допомогою фреймворку Flask. 2. Розробка застосунку за допомогою фреймворку Django.

САМОСТІЙНА РОБОТА

До самостійної роботи студентів включаються:

1. Знайомство з науковою та навчальною літературою відповідно зазначених у програмі тем.
2. Опрацювання лекційного матеріалу.
3. Підготовка до практичних занять.
4. Консультації з викладачем протягом семестру.
5. Самостійне опрацювання окремих питань навчальної дисципліни.
6. Підготовка до підсумкового контролю.

Тематика та питання до самостійної підготовки та індивідуальних завдань

Семестр 1. Патерни проєктування програмного забезпечення

Тема 1. Патерни у програмному забезпеченні Історія виникнення патернів у програмній інженерії. Роль патернів у повторному використанні архітектурних рішень. Класифікація патернів програмного забезпечення. Структура опису патернів проєктування. Поняття антипатернів у програмному забезпеченні.
Тема 2. Типові архітектури програмного забезпечення Архітектура Model–View–Controller. Багаторівнева архітектура програмного забезпечення. Архітектура Client–Server. Архітектура Repository. Порівняння типових архітектур програмних систем.
Тема 3. Породжувальні патерни Призначення породжувальних патернів у програмному забезпеченні. Патерни Factory Method, Abstract Factory, Builder, Prototype, Singleton.
Тема 4. Структурні патерни Призначення структурних патернів у програмному забезпеченні. Патерни Adapter, Bridge, Composite, Decorator, Facade, Flyweight, Proxy.
Тема 5. Поведінкові патерни Призначення поведінкових патернів у програмному забезпеченні. Патерни Observer, Mediator, Strategy, Command, State, Iterator, Visitor.

Семестр 2. Фреймворки для розробки програмного забезпечення

Тема 6. Використання фреймворків для створення програмного забезпечення Поняття фреймворку у програмному забезпеченні. Відмінність між фреймворками, бібліотеками та програмними платформами. Переваги використання фреймворків під час розробки програмних систем. Архітектурні особливості сучасних фреймворків.
Тема 7. Типи фреймворків для розробки програмного забезпечення Класифікація фреймворків за сферою застосування. Вебфреймворки. Фреймворки для застосунків для персональних комп'ютерів. Мобільні фреймворки. Ігрові фреймворки. Фреймворки для тестування програмного забезпечення.
Тема 8. Фреймворки для розробки застосунків для персональних комп'ютерів Архітектура застосунків для персональних комп'ютерів. Фреймворки для створення графічних інтерфейсів користувача. Компонентні моделі графічних інтерфейсів. Порівняння сучасних фреймворків для створення застосунків для персональних комп'ютерів.
Тема 9. Фреймворки для розробки мобільних застосунків Архітектура мобільних застосунків. Нативні та кросплатформні мобільні фреймворки. Особливості створення інтерфейсів мобільних застосунків. Інтеграція мобільних застосунків із вебсервісами.
Тема 10. Фреймворки для розробки вебзастосунків Архітектура сучасних вебзастосунків. Фреймворки серверної частини вебзастосунків. Фреймворки клієнтської частини вебзастосунків. Порівняння сучасних вебфреймворків.

Завдання для підготовки до практичних занять

Семестр 1. Патерни проєктування програмного забезпечення

Тема 1. Патерни у програмному забезпеченні

Підготувати приклад опису патерну за структурою: назва, призначення, задача, рішення, структура, учасники. Навести приклади антипатернів у реальних проєктах та пояснити їхні наслідки. Порівняти три категорії патернів (породжувальні, структурні, поведінкові) на прикладах.

Тема 2. Типові архітектури програмного забезпечення

Реалізувати просту програму з архітектурою **MVC** (наприклад, калькулятор або список завдань). Побудувати схему багаторівневої архітектури (Layered Architecture) для умовної системи управління бібліотекою. Порівняти архітектури Client–Server та Repository на прикладі системи управління базою даних.

Тема 3. Породжувальні патерни

Реалізувати патерн Factory Method для створення різних типів документів (Word, PDF). Використати Abstract Factory для створення графічних елементів інтерфейсу (кнопки, поля вводу) у різних стилях. Реалізувати патерн Builder для побудови складного об’єкта (наприклад, комп’ютер із різними конфігураціями). Продемонструвати патерн Prototype через клонування об’єкта «Користувач». Реалізувати патерн Singleton для класу «Налаштування програми».

Тема 4. Структурні патерни

Реалізувати патерн Adapter для узгодження інтерфейсу старого класу з новим API. Використати патерн Bridge для відокремлення абстракції «Фігура» від реалізації «Рисування». Реалізувати патерн Composite для представлення ієрархії файлів і папок. Продемонструвати патерн Decorator для додавання нових функцій до класу «Повідомлення» без зміни його коду. Реалізувати патерн Facade для спрощення роботи з підсистемою «Мультимедіа». Використати патерн Flyweight для оптимізації збереження великої кількості об’єктів «Символ». Реалізувати патерн Proxy для контролю доступу до ресурсу (наприклад, база даних).

Тема 5. Поведінкові патерни

Реалізувати патерн Observer для системи сповіщень (наприклад, підписка на новини). Використати патерн Mediator для організації взаємодії між кількома компонентами чату. Реалізувати патерн Chain of Responsibility для обробки запитів у системі логування. Продемонструвати патерн Strategy для вибору алгоритму сортування залежно від умов. Реалізувати патерн Command для реалізації операцій «Скасувати» та «Повторити» у текстовому редакторі. Використати патерн State для моделювання поведінки об’єкта «Документ» (чернетка, опублікований, архівований). Реалізувати патерн Iterator для обходу елементів колекції. Використати патерн Template Method для визначення загальної структури алгоритму з можливістю перевизначення окремих кроків. Реалізувати патерн Visitor для виконання операцій над елементами складної структури (наприклад, дерево виразів).

Теми доповідей

1. Історія виникнення патернів у програмуванні
2. Класифікація патернів у програмному забезпеченні
3. Основні категорії патернів та їх застосування

4. Структура опису патерну та її елементи
5. Переваги і обмеження використання патернів
6. Антипатерни та типові помилки застосування
7. Архітектура програмного забезпечення і її роль у створенні систем
8. Архітектура Model View Controller та приклади застосування
9. Багаторівнева архітектура Layered Architecture і Client Server
10. Архітектура Repository та централізоване зберігання даних
11. Порівняння типових архітектур і принципи вибору
12. Factory Method призначення та приклади використання
13. Abstract Factory створення сімейств об’єктів
14. Builder поетапне створення складних об’єктів
15. Prototype використання механізму клонування об’єктів
16. Singleton забезпечення єдиного екземпляра класу
17. Adapter узгодження інтерфейсів
18. Bridge відокремлення абстракції від реалізації
19. Composite побудова ієрархічних структур
20. Decorator розширення функціональності об’єктів
21. Facade спрощення взаємодії підсистем
22. Flyweight оптимізація використання ресурсів
23. Proxy керування доступом до об’єктів
24. Observer організація сповіщень
25. Mediator взаємодія між компонентами
26. Chain of Responsibility обробка запитів
27. Strategy інкапсуляція алгоритмів
28. Command реалізація операцій Скасувати та Повторити
29. State керування станами об’єкта
30. Iterator організація обходу структур даних
31. Template Method визначення структури алгоритму
32. Visitor операції над складними структурами
33. Поняття фреймворку та його відмінність від бібліотек і платформ

34.Переваги використання фреймворків у розробці програмних систем

35.Архітектурні особливості фреймворків та інверсія керування

36.Типові проблеми інтеграції фреймворків у програмні системи

37.Класифікація фреймворків за сферою застосування

38.Вебфреймворки особливості та приклади

39.Фреймворки для застосунків для персональних комп’ютерів

40.Мобільні фреймворки нативні та кросплатформні

41.Фреймворки для тестування та автоматизації розробки

42.Критерії вибору фреймворку для створення програмної системи

43.Qt архітектура та функціональні можливості

44..NET особливості та приклади застосування

45.Electron створення кросплатформних застосунків

46.Інтеграція застосунків для ПК із зовнішніми сервісами

47.Flutter архітектура та сфери застосування

48.React Native особливості та приклади використання

49.Xamarin інтеграція з мобільними платформами

50.Компонентна структура мобільних застосунків

51.Django і Flask серверна частина вебсистем

52.Spring архітектура та можливості

53.Angular організація інтерфейсу користувача

54.React керування станом вебдодатків

55.Vue простота та гнучкість у веброзробці

56.Порівняння сучасних вебфреймворків

ВИДИ ТА МЕТОДИ КОНТРОЛЮ

Семестр 1. Патерни проєктування програмного забезпечення

Види контролю		Складові оцінювання
Поточний контроль, який здійснюється під час проведення практичних занять, виконання самостійної роботи, індивідуальних завдань, дослідної роботи здобувача тощо.		100%
Підсумковий контроль – залік		
Методи діагностики знань (контролю)	Фронтальне опитування; наукова доповідь, реферат, усне повідомлення, індивідуальне опитування; робота у групах; ділова гра, розв’язання ситуаційних завдань, кейсів, виконання практичних завдань, залік	

Питання для підсумкового контролю

1. Поняття патерну у програмному забезпеченні, його призначення та роль у розробці програмних систем.
2. Історія виникнення патернів у програмній інженерії та розвиток підходів до повторного використання проєктних рішень.
3. Переваги використання патернів у програмному забезпеченні.
4. Обмеження використання патернів та типові помилки їх застосування.
5. Класифікація патернів програмного забезпечення.
6. Архітектурні патерни, патерни проєктування та ідіоми програмування: призначення та відмінності.
7. Структура опису патерну: призначення, задача, рішення, структура, учасники, наслідки застосування.
8. Поняття антипатерну у програмному забезпеченні та приклади антипатернів.
9. Поняття архітектури програмного забезпечення та її роль у створенні програмних систем.

10. Архітектура Model–View–Controller: структура, призначення та сфери застосування.

11. Багаторівнева архітектура програмного забезпечення: особливості, переваги та обмеження.

12. Архітектура Client–Server: структура, призначення та особливості використання.

13. Архітектура Repository: призначення, особливості та сфери застосування.

14. Порівняння типових архітектур програмного забезпечення та принципи вибору архітектури системи.

15. Породжувальні патерни: призначення, особливості та сфери застосування.

16. Патерн Factory Method: призначення, структура, особливості та сфера застосування.

17. Патерн Abstract Factory: призначення, структура, особливості та сфера застосування.

18. Порівняння патернів Factory Method та Abstract Factory.

19. Патерн Builder: призначення, структура, особливості та сфера застосування.

20. Патерн Prototype: призначення, структура, особливості та сфера застосування.

21. Патерн Singleton: призначення, особливості реалізації, переваги, недоліки та сфера застосування.

22. Структурні патерни: призначення, особливості та сфери застосування.

23. Патерн Adapter: призначення, особливості та сфера застосування.

24. Патерн Bridge: призначення, особливості та сфера застосування.

25. Патерн Composite: призначення, особливості та сфера застосування.

26. Патерн Decorator: призначення, особливості та сфера застосування.

27. Патерн Facade: призначення, особливості та сфера застосування.

28. Патерн Flyweight: призначення, особливості та сфера застосування.

29. Патерн Proxy: призначення, особливості та сфера застосування.

30. Порівняння патернів Adapter, Decorator, Facade і Proxy.

31. Поведінкові патерни: призначення, особливості та сфери застосування.

32. Патерн Observer: призначення, особливості та сфера застосування.

33. Патерн Mediator: призначення, особливості та сфера застосування.

34. Патерн Chain of Responsibility: призначення, особливості та сфера застосування.

35. Патерн Strategy: призначення, особливості та сфера застосування.

36. Патерн Command: призначення, особливості та сфера застосування.

37. Патерн State: призначення, особливості та сфера застосування.

38. Патерн Iterator: призначення, особливості та сфера застосування.

39. Патерн Template Method: призначення, особливості та сфера застосування.

40. Патерн Visitor: призначення, особливості та сфера застосування.

Семестр 2. Фреймворки для розробки програмного забезпечення

Види контролю		**Складові оцінювання**
Поточний контроль, який здійснюється під час проведення практичних (лабораторних) занять, виконання самостійної роботи, індивідуальних завдань, дослідної роботи здобувача тощо		50%
Підсумковий контроль, який здійснюється у ході проведення екзамену		50%
Методи діагностики знань (контролю)	Фронтальне опитування, розв`язання практичних завдань, тестування здобувачів, наукові доповіді, реферати, усні повідомлення, екзамен	

Питання для підсумкового контролю

1. Поняття фреймворку у програмному забезпеченні, його призначення та роль у розробці програмних систем.
2. Відмінність між фреймворком, бібліотекою та програмною платформою.
3. Переваги використання фреймворків під час розробки програмного забезпечення.

4. Недоліки та обмеження використання фреймворків у програмних системах.
5. Архітектурні особливості сучасних фреймворків.
6. Поняття каркасу програмної системи та його роль у розробці програмного забезпечення.
7. Інверсія керування (Inversion of Control) у фреймворках.
8. Розширення функціональності програмних систем за допомогою фреймворків.
9. Класифікація фреймворків програмного забезпечення.
10. Типи фреймворків за сферою застосування.
11. Вебфреймворки: призначення, особливості та сфери застосування.
12. Фреймворки для розробки застосунків для персональних комп'ютерів: призначення, особливості та сфери застосування.
13. Фреймворки для розробки мобільних застосунків: призначення, особливості та сфери застосування.
14. Фреймворки для тестування програмного забезпечення: призначення, особливості та сфери застосування.
15. Критерії вибору фреймворку для створення програмної системи.
16. Архітектура застосунків для персональних комп'ютерів.
17. Фреймворки для створення графічних інтерфейсів користувача.
18. Фреймворк Qt: призначення, архітектурні особливості та сфера застосування.
19. Фреймворк .NET: призначення, архітектурні особливості та сфера застосування.
20. Фреймворк Electron: призначення, архітектурні особливості та сфера застосування.
21. Архітектура мобільних застосунків.
22. Нативні та кросплатформні мобільні фреймворки.
23. Фреймворк Flutter: призначення, архітектурні особливості та сфера застосування.
24. Фреймворк React Native: призначення, архітектурні особливості та сфера застосування.

25. Фреймворк Xamarin: призначення, архітектурні особливості та сфера застосування.
26. Компонентна структура мобільних застосунків.
27. Архітектура вебзастосунків.
28. Клієнтська та серверна частини вебзастосунків.
29. Фреймворк Django: призначення, архітектурні особливості та сфера застосування.
30. Фреймворк Flask: призначення, архітектурні особливості та сфера застосування.
31. Фреймворк Spring: призначення, архітектурні особливості та сфера застосування.
32. Фреймворк Angular: призначення, архітектурні особливості та сфера застосування.
33. Фреймворк React: призначення, архітектурні особливості та сфера застосування.
34. Фреймворк Vue: призначення, архітектурні особливості та сфера застосування.
35. Маршрутизація у вебфреймворках.
36. Взаємодія вебфреймворків із базами даних.
37. Шаблонізація у вебфреймворках.
38. Організація клієнтського інтерфейсу у сучасних вебзастосунках.
39. Порівняння сучасних вебфреймворків.
40. Перспективи розвитку фреймворків в інженерії програмного забезпечення.

ОЦІНЮВАННЯ ПОТОЧНОЇ, САМОСТІЙНОЇ ТА ІНДИВІДУАЛЬНОЇ РОБОТИ ЗДОБУВАЧІВ

Семестр 1. Патерни проєктування програмного забезпечення

Види роботи	**Планові терміни виконання**	**Форми контролю та звітності**	**Максимальний відсоток оцінювання**
Систематичність і активність роботи на практичних заняттях			
1.1. Підготовка до практичних занять	Відповідно до робочої програми та розкладу занять	Перевірка обсягу та якості засвоєного матеріалу під час практичних занять	**60**
Виконання завдань для самостійного опрацювання			
1.2. Індивідуальне тестування	Відповідно до робочої програми та розкладу занять	Перевірка результатів індивідуального тестування, перевірка конспектів лекцій, перевірка рефератів	**20**
Виконання індивідуальних завдань (дослідна робота здобувача)			
1.3. Інші види індивідуальних завдань, в т.ч. підготовка наукових публікацій, участь у роботі круглих столів, конференцій тощо.	Відповідно до робочої програми та розкладу занять	Обговорення результатів проведеної роботи під час аудиторних занять, наукових конференцій та круглих столів.	**20**
Разом балів за поточний контроль			**100**
Загальний результат оцінювання			**100**

Поточний контроль знань здобувачів освіти при формі контролю залік (максимум 100 балів) оцінюється за шкалою («2», «3», «4», «5»), з обов'язковим переведенням балів за кожний вид роботи таким чином:

– за підготовку до аудиторних занять (максимум 60 балів) здобувачу освіти необхідно мати не менше 1/2 оцінок від загальної кількості практичних (лабораторних) занять, для переводу балів за підготовку до аудиторних занять середньоарифметична оцінка множиться на коефіцієнт 12;

– за виконання завдань для самостійного опрацювання (тестування, завдання, підготовка реферату (есе) за заданою тематикою) (максимум 20 балів) здобувачу освіти необхідно мати не менше 1/2 оцінок від загальної кількості програмного матеріалу, що виноситься на самостійне вивчення, для переводу зазначених балів середньоарифметична оцінка множиться на коефіцієнт 4;

– за інші види індивідуальних завдань, в т.ч. підготовку наукових публікацій, участь у роботі круглих столів, конференцій тощо (максимум 20 балів) здобувачу освіти необхідно мати не менше 1/2 оцінок від загальної кількості інших видів індивідуальних завдань, для переводу балів за підготовку до аудиторних занять середньоарифметична оцінка множиться на коефіцієнт 4.

Критерії оцінювання поточного контролю

– **«2»** – виставляється за неправильну відповідь (письмову роботу), яка не відповідає змісту теми та свідчить про нерозуміння її основних положень;

– **«3»** – виставляється за відповідь (письмову роботу), яка відповідає змісту теми, але є неповною і неточною у визначенні понять, свідчить про неповне розуміння основних положень навчального матеріалу, свідчить про те, що знання здобувача мають фрагментарний, поверховий характер;

– «**4**» – оцінюється правильна і обґрунтована відповідь (письмова робота), з якої видно, що здобувач знає й розуміє теоретичний матеріал в обсязі навчальної теми, володіє необхідними навичками, не допускає суттєвих помилок.

– «**5**» – оцінюється повна, правильна, ґрунтовна відповідь (письмова робота) на запитання теми, що передбачає логічність і послідовність викладу матеріалу, володіння термінологією, демонструє вміння повно й глибоко використовувати теоретичні знання у практичній площині.

Семестр 2. Фреймворки для розробки програмного забезпечення

Види роботи	Планові терміни виконання	Форми контролю та звітності	Максимальний відсоток оцінювання
Систематичність і активність роботи на практичних заняттях			
1.1. Підготовка до практичних занять	Відповідно до робочої програми та розкладу занять	Перевірка обсягу та якості засвоєного матеріалу під час практичних занять	**30**
Виконання завдань для самостійного опрацювання			
1.2. Індивідуальне тестування	Відповідно до робочої програми та розкладу занять	Перевірка результатів індивідуального тестування, перевірка конспектів лекцій, перевірка рефератів	**10**
Виконання індивідуальних завдань (дослідна робота здобувача)			
1.3. Інші види індивідуальних завдань, в т.ч. підготовка наукових публікацій, участь у роботі круглих столів, конференцій тощо.	Відповідно до робочої програми та розкладу занять	Обговорення результатів проведеної роботи під час аудиторних занять, наукових конференцій та круглих столів.	**10**
Разом балів за поточний контроль			**50**
Підсумковий контроль – екзамен			**50**
Загалом балів			**100**

Поточний контроль знань здобувачів при підсумковому контролі у формі екзамену (максимум 50 балів) оцінюється за шкалою («2», «3», «4», «5»), з обов'язковим переведенням балів за кожний вид роботи таким чином:

– за підготовку до аудиторних занять (максимум 30 балів) здобувачу освіти необхідно мати не менше 1/2 оцінок від загальної кількості практичних (лабораторних) занять, для переводу балів за підготовку до аудиторних занять середньоарифметична оцінка множиться на коефіцієнт 6;

– за виконання завдань для самостійного опрацювання (тестування, завдання, підготовка реферату за заданою тематикою) (максимум 10 балів) здобувачу освіти необхідно мати не менше 1/2 оцінок від загальної кількості програмного матеріалу, що виноситься на самостійне вивчення, для переводу зазначених балів середньоарифметична оцінка множиться на коефіцієнт 2;

– за інші види індивідуальних завдань, в т.ч. підготовку наукових публікацій, участь у роботі круглих столів, конференцій тощо (максимум 10 балів) здобувачу освіти необхідно мати не менше 1/2 оцінок від загальної кількості інших видів індивідуальних завдань, для переводу балів за підготовку до аудиторних занять середньоарифметична оцінка множиться на коефіцієнт 2.

Критерії оцінювання поточного контролю

– **«2»** – виставляється за неправильну відповідь (письмову роботу), яка не відповідає змісту теми та свідчить про нерозуміння її основних положень;

– **«3»** – виставляється за відповідь (письмову роботу), яка відповідає змісту теми, але є неповною і неточною у визначенні понять, свідчить про неповне розуміння основних положень навчального матеріалу, свідчить про те, що знання здобувача мають фрагментарний, поверховий характер;

– **«4»** – оцінюється правильна і обґрунтована відповідь (письмова робота), з якої видно, що здобувач знає й розуміє теоретичний матеріал в обсязі навчальної теми, володіє необхідними навичками, не допускає суттєвих помилок.

– **«5»** – оцінюється повна, правильна, ґрунтовна відповідь (письмова робота) на запитання теми, що передбачає логічність і послідовність викладу матеріалу, володіння термінологією, показує вміння повно й глибоко використовувати теоретичні знання в практичній площині.

Критерії оцінювання підсумкового контролю у формі екзамену

Максимальна оцінка підсумкового контролю у формі екзамену не може перевищувати 50 балів. При цьому рівень знань оцінюється:

– **від 40 до 50 балів** – здобувач виявляє особливі творчі здібності, вміє самостійно знаходити та опрацьовувати необхідну інформацію, демонструє знання матеріалу, проводить узагальнення і висновки;

– **від 30 до 40 балів** – здобувач володіє знаннями матеріалу, але допускає незначні помилки у формуванні термінів, категорій, проте за допомогою викладача швидко орієнтується і знаходить правильні відповіді;

– **від 20 до 30 балів** – здобувач відтворює значну частину теоретичного матеріалу, виявляє знання і розуміння основних положень, з допомогою викладача може аналізувати навчальний матеріал, але дає недостатньо обґрунтовані, невичерпні відповіді, допускає помилки.

– **від 10 до 20 балів** – здобувач володіє навчальним матеріалом на середньому рівні, допускає помилки, серед яких є значна кількість суттєвих;

– від 0 до 10 балів – здобувач володіє навчальним матеріалом на рівні, вищому за початковий, значну частину його відтворює на репродуктивному рівні, на всі запитання дає необґрунтовані, невичерпні відповіді, допускає помилки.

ОЦІНЮВАННЯ ЗАГАЛЬНИХ РЕЗУЛЬТАТІВ ЗНАНЬ ЗДОБУВАЧІВ

Загальні результати знань здобувачів оцінюються наступним чином:

– **«Відмінно»/«зараховано» (A)** – від 90 до 100 балів. Здобувач виявляє особливі творчі здібності, вміє самостійно знаходити та опрацьовувати необхідну інформацію, демонструє знання матеріалу, проводить узагальнення і висновки. Був присутній на лекціях та практичних заняттях, під час яких давав вичерпні, обґрунтовані, теоретично і практично правильні відповіді, має конспект з виконаними завданнями до самостійної роботи, презентував реферат за заданою тематикою, проявляє активність і творчість у науково-дослідній роботі;

– **«Добре»/«зараховано» (B)** – від 82 до 89 балів. Здобувач володіє знаннями матеріалу, але допускає незначні помилки у формуванні термінів, категорій, проте за допомогою викладача швидко орієнтується і знаходить правильні відповіді. Був присутній на лекціях та практичних заняттях, має конспект з виконаними завданнями до самостійної роботи, презентував реферат за заданою тематикою, проявляє активність і творчість у науково-дослідній роботі;

– **«Добре»/«зараховано» (C)** – від 74 до 81 балів. Здобувач відтворює значну частину теоретичного матеріалу, виявляє знання і розуміння основних положень, з допомогою викладача може аналізувати навчальний матеріал, але дає недостатньо обґрунтовані, невичерпні відповіді, допускає помилки. При цьому враховується

наявність конспекту з виконаними завданнями до самостійної роботи, реферату та активність у науково-дослідній роботі;

– **«Задовільно»/«зараховано» (D)** – від 64 до 73 балів. здобувач був присутній не на всіх лекціях та практичних заняттях, володіє навчальним матеріалом на середньому рівні, допускає помилки, серед яких є значна кількість суттєвих. При цьому враховується наявність конспекту з виконаними завданнями до самостійної роботи та рефератів;

– **«Задовільно»/«зараховано»** (E) – від 60 до 63 балів. Здобувач був присутній не на всіх лекціях та практичних заняттях, володіє навчальним матеріалом на рівні, вищому за початковий, значну частину його відтворює на репродуктивному рівні, на всі запитання дає необґрунтовані, невичерпні відповіді, допускає помилки, має неповний конспект з завданнями до самостійної роботи.

– **«Незадовільно з можливістю повторного складання»/«не зараховано» (FX)** – від 35 до 59 балів. Здобувач володіє матеріалом на рівні окремих фрагментів, що становлять незначну частину навчального матеріалу.

– **«Незадовільно з обов'язковим повторним вивченням дисципліни»/«не зараховано» (F)** – від 0 до 34 балів. Здобувач не володіє навчальним матеріалом.

Таблиця відповідності результатів контролю знань за різними шкалами

100-бальна шкала	Шкала за ECTS	Національна шкала	
		Екзамен	Залік
90-100	A	Відмінно	Зараховано
82-89	B	Добре	Зараховано
74-81	C		
64-73	D	Задовільно	Зараховано
60-63	E		
35-59	FX	Незадовільно	Не зараховано
0-34	F		

10. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна

1. Freeman E., Robson E. Head First Design Patterns: A Brain-Friendly Guide. 2nd ed. O'Reilly Media, 2020. 672 p. ISBN: 9781492077954, 149207795X.
2. Richards M., Ford N. Fundamentals of Software Architecture: An Engineering Approach. O'Reilly Media, 2020. 432 p. ISBN: 9781492043423, 1492043427.
3. Newman, S. Building Microservices. O'Reilly Media. 2021. 616 p. ISBN: 9781492033998, 1492033995
4. Martin R. C. Clean Architecture: A Craftsman's Guide to Software Structure and Design. Prentice Hall, 2017. 432 p. ISBN: 9780134494326, 0134494326.
5. Kleppmann M. Designing Data-Intensive Applications. O'Reilly Media, 2017. 616 p. ISBN: 9781491903100, 1491903104.

Допоміжна

6. Fowler M. Refactoring: Improving the Design of Existing Code. 2nd ed. Addison-Wesley Professional, 2018. 448 p. ISBN: 9780134757704, 013475770X.
7. Hunt A., Thomas D. The Pragmatic Programmer: Your Journey to Mastery. 2nd ed. Addison-Wesley Professional, 2019. 352 p. ISBN: 9780135956915, 0135956919.
8. Vernon V. Domain-Driven Design. MTM, 2023. ISBN: 9789130090075, 9130090075.
9. Ford N., Richards M., Sadalage P., Dehghani Z. Software Architecture: The Hard Parts. O'Reilly Media, 2021. 462 p. ISBN: 9781492086864, 149208686X.
10. Burns B., Beda J., Hightower K. Kubernetes: Up and Running. 3rd ed. O'Reilly Media, 2022. 328 p. ISBN: 9781098110178, 109811017X.

Інформаційні ресурси

11. Gamma E. et al. Design Patterns Catalog. URL: https://refactoring.guru/design-patterns
12. Microsoft Developer Documentation. URL: https://learn.microsoft.com
13. Mozilla Developer Network (MDN Web Docs). URL: https://developer.mozilla.org
14. Qt Documentation. URL: https://doc.qt.io
15. Flutter Documentation. URL: https://docs.flutter.dev

ЗМІСТ

Навчальне видання

Стрелковська Ірина Вікторівна

Приходько Сергій Борисович

Григор'єва Тетяна Ігорівна

ПАТЕРНИ ТА ФРЕЙМВОРКИ

Методичні рекомендації для самостійної роботи здобувачів