



## ЕФЕКТИВНІ МЕТОДИ ВИВЧЕННЯ ПАТЕРНІВ ПРОЄКТУВАННЯ: АНАЛІЗ СУЧАСНИХ ТЕНДЕНЦІЙ І ОСВІТНІХ ПРАКТИК

**Вступ.** *Визначення ефективних методів вивчення як класичних, так і новітніх патернів проєктування програмного забезпечення, з урахуванням сучасних вимог до його розроблення є актуальним завданням. В основу цього дослідження покладено аналіз новітніх тенденцій щодо практичного застосування різних патернів у процесі розроблення програм, визначення їхньої популярності та пошук методів, що забезпечать ефективніше опанування майбутніми ІТ-фахівцями архітектури застосунків.*

**Методи.** *У процесі дослідження використано методи аналізу й узагальнення інформації, отриманої з відкритих джерел, зокрема, з наукових публікацій, тез, статей, навчальних планів, робочих програм навчальних дисциплін, силябусів та інших нормативних документів. Проаналізовано робочі програми навчальних дисциплін (силябуси) кількох провідних університетів, що дозволило виокремити ефективні методи для вивчення патернів проєктування програмного забезпечення для майбутніх ІТ-фахівців.*

**Результати.** *Аналіз навчальних програм показав, що в університетах України увага зосереджена переважно на вивченні класичних патернів проєктування GoF, що дещо обмежує якість підготовки студентів до роботи з новітніми технологіями. Порівняно з традиційними методами навчання, зокрема, читанням лекцій, проведенням лабораторних занять, використанням UML-діаграм для візуалізації програмних кодів, метод проєктів виявився більш ефективним.*

**Висновки.** *Ефективним методом вивчення патернів проєктування є поєднання класичних лекцій з інтерактивними заняттями та застосуванням методу проєктів. Це сприяє кращому розумінню матеріалу і підготовці студентів до розв'язання реальних задач, які зазвичай є більшими за обсягом і складнішими порівняно з окремою лабораторною роботою, і потребують навичок аналізу і синтезу. Зважаючи на наявні вимоги з боку ринку праці, доцільно оновити навчальні програми, удосконаливши їхній зміст і структуру, зокрема і завдяки впровадженню ефективніших методів навчання.*

**Ключові слова:** *патерни проєктування, методи навчання, метод проєктів.*

### Вступ

В сучасному програмному забезпеченні проєктування є одним із ключових аспектів, що визначає якість, продуктивність і масштабованість розроблюваних програмних систем. Патерни проєктування як узагальнені рішення типових проблем програмування відіграють важливу роль у створенні ефективних, гнучких і підтримуваних програмних рішень (Євланов, 2014). Вивчення патернів проєктування є важливим елементом підготовки фахівців у галузі інформаційних технологій, оскільки дає змогу не лише опанувати кращі практики, а і сформувати навички застосування структурованих підходів до розроблення програмного забезпечення.

Історично основи патернів проєктування були описані у книзі Еріка Гамми, Річарда Гельма, Ральфа Джонсона та Джона Вліссідеса, які 1994 р. у своїй праці "Design Patterns: Elements of Reusable Object-Oriented Software" представили класифікацію 23 класичних патернів проєктування (Gamma et al., 1994). Згодом ці концепції отримали подальший розвиток у зв'язку з еволюцією програмних парадигм і технологій, що призвело до появи нових патернів, зокрема й тих, що стосуються мікросервісної архітектури, хмарних обчислень і розподілених систем.

На сучасному етапі розвитку програмної інженерії вивчення патернів проєктування залишається однією з ключових складових у підготовці фахівців з інженерії програмного забезпечення (Yener, & Theedom, 2014). Зауважимо, що більшість освітніх програм акцентують увагу на класичних патернах GoF, таких як Singleton, Factory, Observer, Proxy та інші, тоді як новітні тенденції, пов'язані з контейнеризацією, подійно-орієнтованою архітектурою (Event-Driven Architecture) та безсерверними (serverless) технологіями, нині меншою мірою інтегровані в навчальні плани (17-695: Design Patterns..., 2022). Це створює певний розрив між академічною підготовкою та реальними вимогами сучасного ринку праці. Інша, не менш важлива невідповідність полягає у переважному застосуванні класичних методів навчання патернам проєктування, що у поєднанні зі складністю власне навчального матеріалу, високим рівнем абстракції не забезпечує їх ефективне опанування (Сілакова, 2017).

Отже, **метою** дослідження є визначення ефективних методів навчання патернам проєктування. Визначення ефективних методів навчання як класичним, так і новітнім патернам дозволить підвищити якість підготовки фахівців у галузі програмної інженерії, сприятиме кращій адаптації випускників до сучасних вимог ІТ-індустрії та забезпечить глибоке розуміння принципів створення гнучких і масштабованих програмних систем.

**Аналіз останніх досліджень і публікацій.** Вивчення патернів проєктування у провідних університетах світу й України є важливим елементом підготовки майбутніх фахівців у сфері інформаційних технологій. Однак аналіз навчальних програм засвідчує, що зміст курсів із патернів проєктування у більшості університетів зосереджений переважно на класичних підходах, тоді як новітні патерни, що орієнтовані на сучасні архітектурні рішення, представлені недостатньо (Factors Contributing..., 2019).



**НТУ України Київський політехнічний інститут імені Ігоря Сікорського (КПІ).** У межах опанування освітньої компоненти "Патерни об'єктно-орієнтованого проектування" студенти вивчають основи застосування класичних патернів GoF (Порев, 2022). Зокрема, у навчальному плані передбачено такі теми:

- поняття патерну проектування. Патерн Singleton;
- різновиди патернів Factory;
- патерни Facade, Adapter, Dependency Injection, Bridge;
- патерни Decorator, Observer, Visitor.

Загальний обсяг зазначеного курсу становить 18 навчальних годин, що суттєво обмежує можливості детального опрацювання матеріалу та глибокого розуміння структури коду, враховуючи кількість і різноманіття різних патернів проектування. Крім того, курс орієнтований лише на традиційні патерни, тоді як сучасні архітектурні підходи, зокрема й мікросервісні, безсерверні (serverless) і подійно-орієнтовані (Event-Driven Architecture) патерни, залишаються поза межами програми. Частково вони розглядаються в іншій освітній компоненті "Архітектура програмного забезпечення", проте зміст практичних завдань передбачає лише опрацювання цих шаблонів з погляду системного архітектора. Причому залишається поза увагою практична складова, а саме – розроблення програмного застосування відповідної архітектури.

**Національний університет "Києво-Могилянська академія".** Навчальна програма передбачає освітню компоненту, що охоплює повний цикл розроблення програмної системи, включаючи методи аналізу та проектування, уніфікований процес розроблення об'єктно-орієнтованих програм і керування програмним проектом. У курсі практикується розроблення навчальних програмних проєктів студентами в невеликих групах із використанням систем управління версіями, веборієнтованих середовищ і рецензування результатів (Освітня програма..., 2018).

**Київський національний університет імені Тараса Шевченка.** Навчання за спеціальністю "Інженерія програмного забезпечення" проводять на двох факультетах: факультеті комп'ютерних наук і кібернетики (ФКНК) та факультеті інформаційних технологій (ФІТ). Зауважимо, що зміст освітніх компонент, пов'язаних з опануванням об'єктно-орієнтованого програмування, відрізняється суттєво за переліками тем, їх наповненням і розподілом годин, виокремлених на опанування, наприклад, патернів проектування. Вивчення патернів проектування на ФКНК в межах опанування освітньої компоненти "Об'єктно-орієнтоване програмування" концентрується на патернах GoF. Загальний обсяг дисципліни складає 120 год (4 кредити ECTS). На думку авторів дослідження, його не можна вважати достатнім, зважаючи на майбутній фах студентів, адже для вивчення патернів проектування із загального обсягу виокремлено лише три теми, що складає загалом 28 год, з них аудиторних – лише 12 год. Серед методів навчання наявні лише класичні: лекція, лабораторне заняття і самостійна робота (Омельчук, 2022). Проте окремо від основної дисципліни у навчальному плані також присутній "Лабораторний практикум з об'єктно-орієнтованого програмування", що традиційно покращує процес навчання і дозволяє застосовувати той самий метод проєктів, підвищуючи ефективність вивчення матеріалу.

Характерною ознакою організації навчального процесу на ФІТ є запровадження інноваційних технологій навчання, таких як STEM-освіта, в основу якої покладено розвиток критичного мислення, інженерного підходу і вміння застосовувати отримані знання у реальному житті. Щодо освітньої компоненти "Об'єктно-орієнтоване конструювання програм" зауважимо, що вона викладається протягом двох семестрів, є суттєвою за обсягом (становить 330 год, тобто 11 кредитів ECTS), вміщує огляд патернів GoF (загальним обсягом 56 год, з них аудиторних 28 год.) і архітектурних патернів (MVC, MVP, MVVM), подійно-орієнтовані (Event-Driven Architecture) патерни тощо, передбачає застосування крім класичних методів навчання методу проєктів, що, за результатами проведених серед студентів опитувань, позитивно впливає на ефективність вивчення матеріалу (Зубик, 2023).

**University of Vienna.** На відміну від українських вишів, у Віденському університеті патерни проектування вивчають у межах магістерської програми з інформатики в курсі "Internet Computing and Software Technologies". Цей курс включає детальний розгляд методів проектування, архітектур і патернів, що застосовуються у сучасних розподілених системах. Досліджується підтримка патернів у скриптових мовах програмування та сучасних патернів проектування, таких як *Saga Pattern*, *CQRS*, *Sidecar Pattern*, і це дозволяє студентам отримувати знання не лише про класичні, а й про новітні патерни, орієнтовані на сучасні технології (232 Curriculum..., 2022).

Отже, аналіз навчальних програм засвідчує суттєві відмінності у методах вивчення патернів проектування у різних університетах. У той час як європейські університети, зокрема й Віденський, приділяють увагу сучасним архітектурним рішенням, українські заклади вищої освіти поки що зосереджені переважно на вивченні патернів GoF і застосуванні класичних методів навчання, що може обмежувати готовність випускників до роботи у високотехнологічному середовищі.

**Boston University.** У Boston University вивчення патернів проектування відбувається у курсі "MET CS665: Software Design and Patterns", де наголос робиться на практичному застосуванні принципів об'єктно-орієнтованого проектування, включаючи GoF-патерни. В курсі інтегровано кейс-стаді з використанням сучасних мов програмування та приклади з проєктів реального бізнес-середовища (Software Design..., 2025). Проєктний метод навчання забезпечує студентам тісний зв'язок між теорією та практикою на сучасному рівні.

**Carnegie Mellon University (CMU).** Велику увагу CMU приділяє вивченню патернів проектування у контексті системного мислення та гнучких методологій розробки. Курс "17-695: Design Patterns", який викладають у межах програми Master of Software Engineering, включає не лише класичні шаблони, а й аналізує їхню еволюцію, використання в розподілених середовищах і взаємозв'язок із принципами програмної архітектури (17-695: Design Patterns..., 2022). Особливістю підходу в CMU є орієнтація на аналітичне мислення, розв'язання реальних інженерних проблем та адаптацію патернів до складних проєктних сценаріїв. Отже, порівняльний аналіз змісту навчальних програм, виконаний на прикладі шести провідних технічних університетів, свідчить про суттєві відмінності у методах вивчення патернів проектування між вітчизняними та зарубіжними закладами вищої освіти.

У навчальних програмах таких університетів як Carnegie Mellon University (17-695: Design Patterns..., 2022), University of Vienna (232 Curriculum..., 2022) та Boston University (Software Design..., 2025), приділяють більше уваги не лише класичним шаблонам, а й сучасним архітектурним рішенням, що формує здатність студентів швидко адаптуватися до викликів високотехнологічного ринку праці завдяки ефективнішому опануванню шаблонів проектування програмного забезпечення. Вітчизняні програми потребують перегляду їхнього змісту, зокрема й у контексті актуалізації методів навчання, відповідно до глобальних тенденцій у сфері програмного інжинірингу.



## Методи

Шаблони проектування є важливим аспектом об'єктно-орієнтованого програмування, що дозволяє створювати гнучкі, масштабовані та підтримувані програмні рішення. Проте навчання цих концепцій часто викликає труднощі серед студентів, оскільки вони мають високий рівень абстракції та потребують глибокого розуміння архітектурних принципів. У цій роботі розглянуто різні методи вивчення шаблонів проектування і запропоновано кращий метод навчання у контексті його ефективності.

Дослідження навчального процесу показує, що студенти, які мають базові знання об'єктно-орієнтованого програмування, часто стикаються з труднощами під час вивчення шаблонів проектування. Основні виклики, з якими вони зустрічаються, містять високу абстрактність шаблонів, що ускладнює розуміння їхніх переваг над традиційними методами розробки, відсутність зв'язку між шаблонами та реальними практичними задачами, а також труднощі у використанні UML-діаграм, які є важливим інструментом для розуміння архітектури програмних систем.

Під час навчального процесу у згаданих вище університетах акцент зазвичай робиться на найпоширеніших шаблонах проектування, які поділяють на три категорії: породжувальні, структурні та поведінкові. До породжувальних шаблонів належать Singleton, який гарантує існування лише одного екземпляра класу, та Factory Method, що забезпечує гнучке створення об'єктів без прив'язки до конкретного класу. Структурні шаблони включають Decorator, який дає змогу динамічно розширювати функціональність класів, не змінюючи їхнього коду. Серед поведінкових шаблонів особливо актуальні такі: Observer, який реалізує механізм спостереження за змінами стану об'єкта, та Strategy, що дозволяє динамічно змінювати алгоритми під час виконання програми.

У процесі навчання зазвичай застосовують різні методи, які можна умовно розділити на традиційні (класичні) й інтерактивні.

**Традиційні (класичні)** методи, такі як читання лекційних і самостійне опрацювання навчальних матеріалів, за результатами опитувань студентів виявилися менш ефективними (Shatz, 2024), оскільки студенти часто не могли пов'язати отримані знання з практичними завданнями, наближеними до реальних. Лекції, що базувалися на теоретичних поясненнях і UML-діаграмах, демонстрували низьку ефективність через відсутність безпосередньої взаємодії студентів із кодом.

Натомість **інтерактивні методи** виявилися значно ефективнішими (Добровольська, 2018). Живе кодування, під час якого викладач демонструє реалізацію шаблонів у реальному часі, дозволяє студентам краще засвоювати матеріал, проте воно не завжди передбачає активність студентів. Парне програмування, в якому один студент пише код, а інший пояснює логіку дій, сприяє кращому розумінню концепцій, але також має недоліки, адже важко сепарувати компетентність окремого студента, коли вони працюють у парі, і виокремити незрозумілі саме для нього моменти. Найбільш ефективним виявився **метод проєктів** (Петровський, Пасічник, & Скрипник, 2021), у межах застосування якого студенти розробляють власні програмні рішення з використанням шаблонів. Це сприяє насамперед їх самостійності у прийнятті рішень, й ефективному формуванню й закріпленню знань, оскільки вони мають можливість застосовувати шаблони у реальних сценаріях розробки. Також ефективними виявляються вікторини та короткі тести, які виконуються студентами після опанування кожної окремої теми, і які допомагають оцінити індивідуальний рівень засвоєння матеріалу, виявити прогалини у знаннях. Одним із найменш ефективних методів виявляється надмірне використання UML-діаграм без практичних прикладів їхнього застосування, оскільки це ускладнює сприйняття матеріалу студентами, які не мали достатнього досвіду роботи з такими схемами раніше.

Статистику успішності студентів за умови впровадження проєктного методу навчання наведено на рис. 1.

Для оцінювання ефективності вивчення матеріалу за традиційними та сучасними інноваційними методами доцільно використовувати метод порівняння, заснований на аналізі оцінок студентів. Одним із способів кількісного оцінювання є нормалізоване порівняння середніх значень, що дозволяє визначити рівень впливу інноваційного підходу на результати навчання.

Формально ефективність навчання  $E$  можна обчислити за такою формулою:

$$E = \frac{S_{new} - S_{old}}{\sigma_{old}}, \quad (1)$$

де  $S_{new}$  – середня оцінка студентів, які навчалися за сучасною методикою;  $S_{old}$  – середня оцінка студентів, які навчалися за традиційною методикою;  $\sigma_{old}$  – стандартне відхилення оцінок у групі традиційного навчання.

Стандартне відхилення оцінок у групі традиційного навчання можна визначити за такою формулою:

$$\sigma_{old} = \sqrt{\frac{1}{N} \sum_{i=1}^N (S_i - S_{old})^2}, \quad (2)$$

де  $N$  – кількість студентів у групі традиційного навчання;  $S_i$  – оцінка  $i$ -го студента у цій групі;  $S_{old}$  – середня оцінка у групі традиційного навчання.

Наведені формули дають змогу оцінити ступінь покращення навчальних результатів шляхом нормалізації різниці між середніми значеннями відносно варіативності оцінок у контрольній групі. Якщо отримане значення  $E$  є значущим ( $E > 1$ ), це може свідчити про ефективність проєктного методу засвоєння навчального матеріалу.

У магістерському дослідженні Р. Голубека (Holubek, 2022) здійснено кількісне та якісне оцінювання ефективності вивчення патернів проектування з використанням методу проєктів. Результати дослідження свідчать про підвищення результатів навчання студентів у межах запровадженого методу порівняно з традиційними методами навчання.

Середній бал, отриманий студентами після впровадження методу проєктів, склав 7,8 із 10, при цьому медіанне значення дорівнювало 8, що свідчить про стабільно високі результати більшості учасників. У той самий час, середній бал за результатами іспиту за традиційною методикою становив 6,5, а стандартне відхилення – 1,5. Застосування формули нормалізованого приросту ефективності дозволило кількісно оцінити рівень покращення:

$$E = \frac{S_{new} - S_{old}}{\sigma_{old}} = \frac{7,8 - 6,5}{1,5} \approx 0,87.$$



	I	I	I	I	UML	AE	AE	AD	UML	AD	AE	AD	AE	AD	AE		
	Introduction				Strategy				Decorator				Visitor				
	After break																
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	%	
Student 1	C	D	B	A	A	A	B	B	A	A	A	C	C	C	C	0,53	
Student 2	A	C	B	C	D	A	B	B	B	B	A	A	A	C	A	0,67	
Student 3	-	-	-	-	-	C	B	B	A	A	A	C	A	C	A	0,8	
Student 4	A	C	B	C	A	A	B	A	C	A	A	C	A	A	A	0,73	
Student 5	A	C	B	C	B	A	-	B	A	C	-	B	A	A	-	0,67	
Student 6	-	-	-	-	-	-	-	-	A	A	A	C	-	-	-	0,75	
Student 7	A	C	B	C	D	C	B	A	B	-	-	-	-	-	-	0,67	
Student 8	A	C	B	C	A	C	B	B	A	C	A	C	A	C	C	0,87	
Student 9	-	-	-	-	D	-	-	-	-	-	-	-	-	-	-	0	
Student 10	-	-	-	-	-	A	B	B	A	A	A	A	A	C	A	0,7	
Student 11	A	C	B	A	-	A	B	-	-	A	A	C	-	-	-	0,78	
Student 12	-	-	-	C	D	A	B	C	-	-	-	-	-	-	-	0,8	
Student 13	A	C	B	C	A	A	B	C	A	C	-	-	-	-	-	1	
Student 14	A	C	B	C	A	C	B	A	A	A	A	C	A	C	A	0,87	
Student 15	A	C	B	-	-	-	B	A	-	-	-	-	-	-	-	0,8	
Student 16	A	C	A	D	A	-	C	A	A	A	A	C	A	C	A	0,64	
Student 17	A	C	B	A	A	A	B	B	A	C	A	C	A	A	-	0,79	
Student 18	A	C	B	C	A	A	B	B	A	B	A	C	B	A	A	0,73	
Student 19	-	-	-	-	A	A	B	B	-	C	A	C	-	-	-	0,86	
Student 20	A	C	B	C	A	A	B	B	A	A	A	C	A	C	C	0,8	
Student 21	A	C	B	B	A	A	B	A	A	C	A	C	A	A	A	0,8	
Student 22	A	C	B	A	D	A	B	B	-	-	-	-	-	-	-	0,63	
Student 23	A	C	C	C	D	C	B	A	A	C	A	C	A	C	A	0,8	
Student 24	-	-	-	C	-	-	-	-	-	-	-	-	-	-	-	1	
Student 25	-	-	-	C	-	-	B	B	-	-	-	-	-	-	-	0,67	
Student 26	-	-	B	C	-	-	-	-	-	-	-	-	-	-	-	1	
Student 27	A	C	B	C	C	A	B	A	B	A	-	A	-	-	-	0,55	
Student 28	A	C	B	C	A	A	B	B	A	C	A	C	A	A	A	0,87	
Student 29	A	D	B	C	-	-	-	-	-	-	-	-	-	-	-	0,75	
Student 30	B	-	C	C	-	-	-	-	-	-	-	-	-	-	-	0,33	
Student 31	-	-	-	-	-	-	-	-	A	C	A	C	-	-	-	1	
Student 32	-	-	-	-	-	-	-	-	-	-	-	-	A	A	B	0,33	
Student 33	B	C	B	C	-	-	-	-	-	-	-	-	-	-	-	0,75	
Student 34	A	C	B	C	D	A	B	C	-	A	A	-	A	-	A	0,83	
Student 35	A	C	C	A	A	C	B	B	A	A	A	C	A	A	A	0,67	
Student 36	A	C	D	C	A	C	C	B	-	C	A	-	-	-	-	0,7	
Student 37	A	D	B	C	D	A	B	C	B	C	A	-	-	-	-	0,73	
Student 38	-	C	B	C	C	A	B	B	B	B	A	C	A	C	B	0,64	
Student 39	A	D	B	C	A	A	C	B	A	A	A	C	A	C	A	0,73	
Student 40	A	C	B	C	-	-	-	-	-	-	-	-	-	-	-	1	
Student 41	A	D	B	A	C	A	B	B	C	C	A	C	A	C	A	0,67	
Student 42	-	-	-	-	-	-	B	A	A	B	A	C	B	-	A	0,63	
Student 43	-	-	-	-	-	-	-	-	A	C	A	C	A	-	-	1	
Student 44	A	C	B	C	A	C	-	-	-	-	-	-	-	-	-	1	
Student 45	A	C	B	C	-	-	-	-	-	-	-	-	-	-	-	1	
Student 46	A	C	B	D	A	A	C	B	A	C	A	C	A	A	A	0,73	
Student 47	C	C	D	A	A	C	B	C	-	-	-	-	-	-	-	0,63	
Student 48	-	-	-	-	B	A	B	B	D	A	A	A	A	C	A	0,55	
Student 49	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	
Student 50	-	-	-	-	A	A	B	C	-	-	-	-	-	C	A	1	
Student 51	A	C	A	A	A	C	B	B	A	C	A	C	-	-	-	0,75	
Student 52	D	C	D	C	D	A	C	A	A	C	A	C	A	A	A	0,6	
Student 53	-	-	-	-	A	A	B	B	A	C	A	C	A	C	A	0,91	
Student 54	A	C	B	C	B	C	B	A	B	A	A	C	B	C	-	0,64	
Student 55	-	-	-	-	-	A	B	C	A	A	A	C	A	A	A	0,8	
Percentage correct	0,9	0,9	0,8	0,7	0,6	1	0,9	0,2	0,7	0,5	1	0,8	0,9	0,6	0,8		
Total answers	36	36	38	40	36	38	40	40	34	37	34	33	29	27	26		
Correct answers	31	31	30	29	21	38	35	7	25	20	34	28	25	16	21		

Рис. 1. Дані успішності студентів у результаті запровадження у навчання методу проєктів



Отримане значення  $E \approx 0,87$ , хоч і не перевищує умовного порогу значущості ( $E > 1$ ), однак засвідчує помірне покращення результатів навчання. Позитивну динаміку також підтверджують суб'єктивні оцінки студентів, зафіксовані в анкетуванні після проходження курсу: 85 % респондентів схвально оцінили власне лекцію. У межах аналізу практичної діяльності встановлено, що 70 % студентів змогли коректно застосувати патерни у межах контрольного завдання, а 80 % продемонстрували високе розуміння UML-діаграм. Утім, 60 % студентів зазначили труднощі в реалізації функціонального наповнення патернів, що свідчить про потребу у посиленій роботі на етапі практичної імплементації.

### Результати

У сукупності зазначені дані вказують на те, що застосування проектного методу до вивчення патернів проектування ефективне і веде до поліпшення успішності студентів, покращення засвоєння ними навчального матеріалу. Незважаючи на те, що отримане числове значення ефективності не є високим у межах статистичної моделі, воно демонструє обґрунтованість застосування методу проектів як доповнення до класичних методів навчання, зокрема в аспекті емоційного залучення студентів до навчального процесу з метою забезпечення кращого розуміння ними навчального матеріалу через призму його майбутнього практичного застосування.

Аналіз методів вивчення шаблонів проектування показав, що метод проектів є ефективним (Тадеуш, 2017). Інтерактивні методи покращують сприйняття навчального матеріалу та дозволяють краще його розуміти. Таким чином, для успішного вивчення шаблонів проектування доцільно поєднувати теоретичні лекції із завданнями, наближеними до практичних, які виконуються поетапно протягом тривалого проміжку часу з поступовим нарощуванням програмного коду, активною взаємодією студентів у межах уявних команд і з викладачем, що дає їм змогу ефективно засвоювати знання та застосовувати їх у реальних сценаріях розробки.

### Дискусія і висновки

Підсумовуючи результати дослідження методів вивчення патернів проектування, можна зробити такі висновки. На основі проведеного аналізу навчальних програм і методів навчання в окремих провідних університетах України та деяких зарубіжних закладах вищої освіти виявлено деякі важливі аспекти, що потребують уваги для покращення процесу навчання. Вивчення патернів проектування з використанням традиційних методів навчання, таких як лекції, лабораторні заняття і самостійна робота студентів, виявилось менш ефективним унаслідок обмеженої можливості студентів у застосуванні теоретичних знань у реальних сценаріях розробки. Натомість проектний метод навчання продемонстрував вищу ефективність. Застосування цього методу дозволяє студентам не лише механічно запам'ятовувати теоретичні відомості й відтворювати їх, розв'язуючи задачі, відірвані від реальних проектів, але й підтвердити набуті знання практикою, рухаючись до реалізації задуманого проекту покроково, бути для цього достатньо вмотивованим і орієнтованим на результат. Належна вмотивованість полегшує студентам опанування складних абстракцій.

З огляду на результати дослідження, вважаємо за доцільне зосередити зусилля на оновленні навчальних програм університетів, зокрема і за рахунок коригування їхнього змісту шляхом включення до їхнього змісту сучасних патернів проектування, що відповідають вимогам сьогодення. Інноваційні методи навчання забезпечать студентам кращу підготовку до роботи у сучасному інформаційному середовищі. Метод проектів дозволить студентам досягнути кращих результатів навчання, а також допоможе підготуватися до реальних викликів у їх майбутній діяльності за фахом. Зважаючи на важливість використання патернів проектування у розробці сучасних програмних систем, ефективність їх вивчення є необхідною умовою підготовки висококваліфікованих фахівців у галузі інформаційних технологій.

**Внесок авторів.** Людмила Зубик – концептуалізація дослідження, розроблення методів дослідження та загальне керівництво; Даниїл Фуркало – дослідження літературних джерел, математичні розрахунки та підготовка графічних матеріалів.

**Джерела фінансування.** Це дослідження не отримало жодного гранта від фінансової установи в державному, комерційному або некомерційному секторах.

### Список використаних джерел

- Євланов, М. В. (2014). Патерни проектування вимог до інформаційної системи. *Вісник Національного університету "Львівська політехніка": Інформаційні системи та мережі*, 783, 429–434. <https://openarchive.nure.ua/server/api/core/bitstreams/ccca27a1e-4532-40bc-8a3b-248f662b3d66/content>
- Зубик, Л. В. (2023). Робоча програма навчальної дисципліни "Об'єктно-орієнтоване конструювання програм". КНУ імені Тараса Шевченка. <https://surl.li/djzhuo>
- Омельчук, Л. (2022). Робоча програма навчальної дисципліни "Об'єктно-орієнтоване програмування". КНУ імені Тараса Шевченка. [https://drive.google.com/file/d/1\\_IHCutPL\\_Om\\_SeEeXpO1sQn1WmAhAx0/\\_view](https://drive.google.com/file/d/1_IHCutPL_Om_SeEeXpO1sQn1WmAhAx0/_view)
- Освітня програма "Інженерія програмного забезпечення". (2018). *НУ Києво-могилянська академія*. <https://www.ukma.edu.ua/ects/index.php/2011-04-18-08-31-28/163-bakalavr/bpprogin/248-2018-11-01-10-37-52>
- Петровський, С. С., Пасічник, О. А., & Скрипник, Т. К. (2021). Особливості використання методу проектів при викладанні освітніх компонент спеціальності комп'ютерні науки. *Вісник Хмельницького національного університету. Технічні науки*, 4, 30–34. <https://www.doi.org/10.31891/2307-5732-2021-299-4-30-34>
- Порев, В. М. (2022). *Об'єктно-орієнтоване програмування*. НТТУ "КПІ імені Ігоря Сікорського". <https://ela.kpi.ua/server/api/core/bitstreams/854443cb-3375-4e76-961a-1f9a74a15b07/content>
- Сілакова, Т. Т. (2017). Проектні технології підготовки студентів. *Вісник Національного авіаційного університету. Серія: Педагогіка, Психологія*, 11, 153–158. <https://doi.org/10.18372/2411-264X.11.12571>
- Тадеуш, О. М. (2017). Метод проектів як форма продуктивного навчання студентів. *Науковий часопис НПУ імені М. П. Драгоманова. Серія 16: Теорія особистість учителя: проблеми теорії і практики*, 29, 142–146. [http://nbuv.gov.ua/UJRN/Nchnpu\\_016\\_2017\\_29\\_33](http://nbuv.gov.ua/UJRN/Nchnpu_016_2017_29_33)
- 17-695: Design Patterns. Syllabus (2022). *Carnegie Mellon University*. [https://mse.s3d.cmu.edu/0\\_documents/syllabi/fa2022/17695-design-patterns.pdf](https://mse.s3d.cmu.edu/0_documents/syllabi/fa2022/17695-design-patterns.pdf)
- 232 Curriculum for the master's program in Computer Science (2022). *University of Vienna*. [https://informatik.univie.ac.at/fileadmin/user\\_upload/f\\_informatik/Studium/Master\\_Informatik/curricula/MA\\_Inf\\_2022\\_EN.pdf](https://informatik.univie.ac.at/fileadmin/user_upload/f_informatik/Studium/Master_Informatik/curricula/MA_Inf_2022_EN.pdf)
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.
- Holubek, R. (2022). *Designing the Teaching Pattern: a structured approach to teach design patterns*. Radboud University. [https://www.cs.ru.nl/masters-theses/2022/R\\_Holubek\\_Designing\\_the\\_teaching\\_pattern\\_a\\_structured\\_approach\\_to\\_teach\\_design\\_patterns.pdf](https://www.cs.ru.nl/masters-theses/2022/R_Holubek_Designing_the_teaching_pattern_a_structured_approach_to_teach_design_patterns.pdf)
- Shatz, I. (2024). Assumption-checking rather than (just) testing: The importance of visualization and effect size in statistical diagnostics. *Behavior Research Methods*, 56, 826–845. <https://doi.org/10.3758/s13428-023-02072-x>
- Software Design and Patterns (MET CS-665). Syllabus. (2025). *Boston University Metropolitan College*. <https://www.bu.edu/csmet/files/2025/01/665-syllabus-2025-spring.docx.pdf>
- Yener, M., & Theedom, A. (2014). *Professional Java EE Design Patterns*. John Wiley & Sons.



## References

- 17-695: Design Patterns. Syllabus (2022). *Carnegie Mellon University*. [https://mse.s3d.cmu.edu/0\\_documents/syllabi/fa2022/17695-design-patterns.pdf](https://mse.s3d.cmu.edu/0_documents/syllabi/fa2022/17695-design-patterns.pdf)
- 232 Curriculum for the master's program in Computer Science (2022). *University of Vienna*. [https://informatik.univie.ac.at/fileadmin/user\\_upload/f\\_informatik/Studium/Master\\_Informatik/curricula/MA\\_Inf\\_2022\\_EN.pdf](https://informatik.univie.ac.at/fileadmin/user_upload/f_informatik/Studium/Master_Informatik/curricula/MA_Inf_2022_EN.pdf)
- Educational program "Software Engineering". (2018). *Kyiv-Mohyla National University* [in Ukrainian]. <https://www.ukma.edu.ua/ects/index.php/2011-04-18-08-31-28/163-bakalavr/bpprogin/248-2018-11-01-10-37-52>
- Evlanov, M. V. (2014). Patterns of Designing Requirements for an Information System. *Bulletin of the National University "Lviv Polytechnic": Information Systems and Networks*, 783, 429–434 [in Ukrainian]. <https://openarchive.nure.ua/server/api/core/bitstreams/cca27a1e-4532-40bc-8a3b-248f662b3d66/content>
- Gamma, E., Helm, R., Johnson, R. & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.
- Holubek, R. (2022). *Designing the Teaching Pattern: a structured approach to teach design patterns*. Radboud University. [https://www.cs.ru.nl/masters-theses/2022/R\\_Holubek\\_Designing\\_the\\_teaching\\_pattern\\_a\\_structured\\_approach\\_to\\_teach\\_design\\_patterns.pdf](https://www.cs.ru.nl/masters-theses/2022/R_Holubek_Designing_the_teaching_pattern_a_structured_approach_to_teach_design_patterns.pdf)
- Omelchuk, L. (2022). *Work program of the academic discipline "Object-Oriented Programming"*. Taras Shevchenko National University of Kyiv [in Ukrainian]. [https://drive.google.com/file/d/1\\_IHCutPL\\_0m\\_SeEeXpO1sQn1WmAhAx0/view](https://drive.google.com/file/d/1_IHCutPL_0m_SeEeXpO1sQn1WmAhAx0/view)
- Petrovsky, S. S., Pasichnyk, O. A., & Skrypnyk, T. K. (2021). Peculiarities of using the project method in teaching educational components of the computer science specialty. *Bulletin of Khmelnytsky National University. Technical Sciences*, 4, 30–34 [in Ukrainian]. <https://www.doi.org/10.31891/2307-5732-2021-299-4-30-34>
- Porev, V. M. (2022). Object-oriented programming. Igor Sikorsky Kyiv Polytechnic Institute [in Ukrainian]. <https://ela.kpi.ua/server/api/core/bitstreams/854443cb-3375-4e76-961a-1f9a74a15b07/content>
- Shatz, I. (2024). Assumption-checking rather than (just) testing: The importance of visualization and effect size in statistical diagnostics. *Behavior Research Methods*, 56, 826–845. <https://doi.org/10.3758/s13428-023-02072-x>
- Silakova, T. T. (2017). Project technologies for student training. *Bulletin of the National Aviation University. Series: Pedagogy, Psychology*, 11, 153–158 [in Ukrainian]. <https://doi.org/10.18372/2411-264X.11.12571>
- Software Design and Patterns (MET CS-665). Syllabus (2025). *Boston University Metropolitan College*. <https://www.bu.edu/csnet/files/2025/01/665-syllabus-/2025-spring.docx.pdf>
- Tadeush, O. M. (2017). The project method as a form of productive student learning. *Scientific Journal of the NPU named after M.P. Dragomanov. Series 16: Creative personality of a teacher: problems of theory and practice*, 29, 142–146 [in Ukrainian]. [http://nbuv.gov.ua/UJRN/Nchnpu\\_016\\_2017\\_29\\_33](http://nbuv.gov.ua/UJRN/Nchnpu_016_2017_29_33)
- Yener, M., & Theedom, A. (2014). *Professional Java EE Design Patterns*. John Wiley & Sons.
- Zubyk, L. V. (2023). *Work program of the academic discipline "Object-Oriented Program Design"*. Taras Shevchenko National University of Kyiv [in Ukrainian]. <https://surl.li/djzhuo>

Отримано редакцією журналу / Received: 03.04.25  
Прорецензовано / Revised: 16.04.25  
Схвалено до друку / Accepted: 01.05.25

Daniil FURKALO, Student  
ORCID ID: 0009-0005-8584-7167  
e-mail: daniilf077@knu.ua  
Taras Shevchenko National University of Kyiv, Ukraine

Liudmyla ZUBYK, PhD (Ped.), Assoc. Prof.  
ORCID ID: 0000-0002-2087-5379  
e-mail: zubyk.liudmyla@knu.ua  
Taras Shevchenko National University of Kyiv, Ukraine

## EFFECTIVE APPROACHED TO TEACHING DESIGN PATTERNS: ANALYSIS OF CURRENT TRENDS AND EDUCATIONAL PRACTICES

**Background.** *Determining effective methods for studying both classical and modern software design patterns, taking into account modern requirements for its development, is an urgent task. The basis of this study was the analysis of the latest trends in the practical application of various patterns in the process of program development, determining their popularity and searching for methods that will ensure more effective mastering of application architecture by future IT specialists.*

**Methods.** *The research used methods of analyzing and summarizing information obtained from open sources, in particular, scientific publications, theses, articles, curricula, work programs of educational disciplines, syllabi and other regulatory documents. The work programs of educational disciplines (syllabuses) of several leading universities were analyzed, which allowed us to identify effective methods for studying software design patterns for future IT specialists.*

**Results.** *The analysis of curricula showed that in Ukrainian universities, attention is focused mainly on the study of classical GoF design patterns, which somewhat limits the quality of students' preparation for working with the latest technologies. Compared to traditional teaching methods, in particular, lectures, laboratory classes, and the use of UML diagrams for visualization of program codes, the project method turned out to be more effective.*

**Conclusions.** *An effective method of studying design patterns is to combine classical lectures with interactive classes and the use of the project method. This contributes to a better understanding of the material and prepares students to solve real problems, which are usually larger in volume and more complex than separate laboratory work, and require analysis and synthesis skills. Given the existing requirements of the labor market, it is advisable to update curricula by improving their content and structure, including by introducing more effective teaching methods.*

**Keywords:** *design patterns, learning methods, project method.*

Автори заявляють про відсутність конфлікту інтересів. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; в рішенні про публікацію результатів.

The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; in the decision to publish the results.