

УДК 37.091.2:004.8-021.111(045)
DOI: <https://doi.org/10.24919/2308-4634.2025.344615>

Сергій Мельник, здобувач третього (освітньо-наукового)
рівня вищої освіти кафедри освітнього менеджменту,
артменеджменту та соціальної роботи
Черкаського національного університету імені Богдана Хмельницького
ORCID: <https://orcid.org/0009-0002-8223-6393>

ПЕДАГОГІЧНІ ЗАСАДИ ПРОЄКТУВАННЯ АДАПТИВНИХ НАВЧАЛЬНИХ МОДУЛІВ ДЛЯ ПІДГОТОВКИ ІТ-ФАХІВЦІВ

У статті представлено концепцію розробки адаптивних навчальних модулів для функціонального програмування з урахуванням індивідуалізації освітнього процесу. Проведено анкетування десяти українських ІТ-компаній, що виявило дефіцит розробників Elixir і розкриває потребу в персоналізованій підготовці. На основі аналізу адаптивного та компетентнісного навчання запропоновано триетапну модель з педагогічними стратегіями: діагностичне оцінювання, метод Use-Modify-Create, проєктна діяльність на основі проблем та використання навчальної аналітики. Запропоновані підходи спрямовані на розвиток самостійності, обчислювального мислення студентів та забезпечення відповідності індустріальним вимогам.

Ключові слова: функціональне програмування; Elixir; компетентнісна освіта; проблемно-орієнтоване навчання; навчальна аналітика; бакалаври ІТ; педагогічне проєктування.

Рис. 2. Табл. 1. Літ. 9.

Serhii Melnyk, Applicant of the Third (Educational and Scientific)
Level of Higher Education of the Educational Management,
Art Management and Social Work Department,
Cherkasy Bohdan Khmelnytskyi National University
ORCID: <https://orcid.org/0009-0002-8223-6393>

PEDAGOGICAL PRINCIPLES OF DESIGNING ADAPTIVE EDUCATIONAL MODULES FOR TRAINING IT SPECIALISTS

This paper outlines pedagogical principles for developing adaptive learning modules designed to equip IT undergraduates with proficiency in Elixir functional programming. Recognizing the increasing industrial relevance of functional paradigms alongside student difficulties transitioning from imperative approaches, a survey of ten Ukrainian IT companies was conducted. Results revealed a significant deficit of Elixir developers and challenges associated with staff retraining, directly informing the design of the proposed educational model.

To accommodate varying prior knowledge, a three-stage adaptive framework is presented, progressing through foundational (pattern matching, recursion, pure functions), intermediate (OTP patterns, concurrency, Phoenix), and advanced (distributed systems, fault tolerance, profiling) concepts. Each stage incorporates diagnostic assessment, project-based learning, and mentorship, with clear articulation of key topics and evaluation criteria. The model integrates competency-based and problem-based methodologies alongside adaptive analytics. Learning analytics platforms monitor student performance to provide targeted resource recommendations while instructors assume a facilitative role.

Key pedagogical strategies include initial diagnostic assessments, the Use-Modify-Create task methodology, problem-oriented project assignments, data-driven feedback via learning analytics, and peer support through mentorship programs. The discussion incorporates considerations of scaffolding and self-regulated learning, emphasizing how adaptive modules foster student autonomy while mitigating cognitive load. Furthermore, ethical implications concerning AI-powered recommendations and data privacy are addressed. The paper concludes that adaptive modules offer a viable solution for aligning academic training with industry requirements; however, successful implementation necessitates investment in both technological infrastructure and faculty professional development.

Keywords: functional programming; Elixir; competence education; problem-oriented learning; educational analytics; IT bachelors; pedagogical design.

Постановка проблеми. Функціональне програмування (ФП) швидко переходить з академічних досліджень у промислові проєкти. Мови на кшталт Elixir забезпечують високу конкурентність, відмовостійкість і масштабованість, що робить їх привабливими для розробки веб-сервісів та розподілених систем. Однак, перехід від імперативних парадигм (на-

приклад, JavaScript, Python, PHP) до ФП представляє значні труднощі для студентів. Традиційні університетські навчальні програми часто характеризуються лінійним форматом, що не враховує індивідуальних рівнів підготовки. Разом з тим, на ринку праці простежується стійка тенденція до дефіциту висококваліфікованих кадрів, що спеціалізуються на мовах функціональної парадигми:

результати нашого опитування показали, що 60 % компаній не мають інженерів Elixir, 10 % взагалі не знають про цю технологію, а ті, хто її використовує, залучають обмежену кількість фахівців (до п'яти). Відсутність систематичної підготовки студентів у сфері ФП є стримуючим фактором для розвитку відповідної індустрії.

Аналіз основних досліджень і публікацій. Персоналізовані адаптивні платформи демонструють значний потенціал у покращенні освітнього процесу. Метааналіз 69 досліджень, проведених у сфері вищої освіти протягом періоду з 2012 по 2024 рік, засвідчує, що адаптивне навчання корелює зі статистично значущим підвищенням академічної успішності в 59 % випадків та збільшенням рівня залученості студентів у 36 % випадків. Водночас автори дослідження підкреслюють наявність технічних обмежень, часових витрат, пов'язаних з реалізацією адаптивних систем, а також необхідність подальшого розвитку методів навчальної аналітики [5].

Згідно з дослідженням Laak та Agu, більшість систем персоналізації, керованих штучним інтелектом, переважно зосереджені на оптимізації відбору навчальних матеріалів, проте недостатньо сприяють формуванню саморегульованого навчання та підвищенню когнітивної залученості студентів. Автори пропонують розробку гібридної моделі, що передбачає інтеграцію автоматизованих рекомендацій з фасилітуючою функцією викладача [6].

Компетентісно-орієнтоване навчання програмуванню асоційоване з позитивним довгостроковим впливом на когнітивні здібності та розвиток обчислювального мислення. 12-річне дослідження, проведене в рамках курсу С-програмування, продемонструвало, що інтеграція шести педагогічних теорій у тривимірну компетентісну модель суттєво покращила когнітивні показники студентів: спостерігалось статистично значуще зростання рівня розумового розвитку за тестом Равена ($t = 8,76$; $p < 0,001$; $d = 0,68$), що корелювало з академічними успіхами та розвитком обчислювального мислення. Дослідження ідентифікувало ключові предиктори покращення когнітивних компетентностей, зокрема активну участь у навчальному процесі, завершення проєктів та залучення до інноваційних практик [9].

Проблемно-орієнтоване навчання (PBL) стимулює розвиток критичного мислення, навичок співпраці та застосування знань у реальних контекстах. Систематичний огляд PBL в комп'ютерних науках О'Грейді підкреслює його відхід від традиційних лекцій та аналізує ініціативи на бакалаврських та магістерських програмах, акцентуючи увагу на методах оцінювання, формулюванні проблемних завдань та тематиці курсів. Ці дані підтверджують ефективність PBL у курсах з інформатики та програмування, обґрунтовуючи його інтеграцію в запропоновану модель [7].

Аналіз публікацій засвідчує зростаючий вплив цифрових технологій на формування професійних компетентностей. Дослідження Ткачук та Медведєвої зосереджено на оцінці рівнів інформаційно-цифрової компетентності студентів педагогічних університетів та визначенні шляхів її розвитку [3]. У статті Р. Гуревича, В. Бойчука, Л. Коношевського, О. Коношевського та Н. Костенко проаналізовано потенціал впровадження інноваційних технологій у навчальний процес, наголошуючи на необхідності модернізації педагогічних практик [1]. Ю. Силенко, Я. Романцова, О. Лисицька та Я. Гарбич досліджують інформаційне освітнє середовище як платформу для інтеграції ІКТ, обґрунтовуючи трансформацію ролі викладача до моделі тьютора [2]. Ці публікації підтверджують актуальність цифрової трансформації освіти та потребу в адаптивних стратегіях навчання.

Мета статті – обґрунтування педагогічних засад проєктування адаптивних навчальних модулів для підготовки бакалаврів ІТ-спеціальностей з акцентом на функціональне програмування. Для досягнення поставленої мети вирішувалися наступні завдання:

1. ідентифікація індустріальних потреб та проблем найму розробників Elixir;
2. аналіз траєкторій переходу студентів від імперативного програмування до функціонального підходу;
3. розробка триетапної моделі адаптивного навчання;
4. визначення педагогічних стратегій і інструментів, спрямованих на персоналізацію навчального процесу та забезпечення високої мотивації.

Виклад основного матеріалу. Матеріали дослідження включали онлайн-анкетування десяти ІТ-компаній, спрямоване на з'ясування потреб у спеціалістах Elixir, їхньої кількості, причин вибору/відмови від мови та доступності на ринку. Опитувальник було сформовано на основі роботи команди галузевих експертів, до якої увійшли представники ІТ-компаній та викладачі фахових дисциплін. На підготовчому етапі було проведено кілька робочих зустрічей, під час яких експерти узгодили перелік ключових тематичних блоків та сформулювали запитання, спрямовані на виявлення як поточних, так і перспективних потреб ринку праці. Опрацьовані дані за допомогою описової статистики та візуалізовані у вигляді діаграм (рис. 1–2). Якісний аналіз відповідей респондентів, проведений методом відкритого кодування, дозволив виділити ключові теми (переваги мови, складнощі впровадження, кадрові питання) та сформулювати категорії освітніх потреб: “дефіцит кваліфікованих розробників”, “необхідність практичного досвіду проєктної діяльності” та “значення функціональної парадигми для розподілених систем”. Надійність

результатів анкетування підтверджено методом тріангуляції з експертними інтерв'ю трьох провідних інженерів. Педагогічні засади формування базувалися на аналізі міжнародної та вітчизняної літератури щодо адаптивного навчання, компетентнісного підходу, проблемно-орієнтованого навчання (PBL) та викладання функціонального програмування.

Діаграма на рис. 1 демонструє, що 6 із 10 компаній не мають у штаті розробників Elixir, тоді як 3 компанії мають відповідні команди, а одна не надала відповідь. Діаграма на рис. 2 відображає

розподіл за чисельністю команд: дві організації мають 3-5 інженерів, одна – понад десять, ще одна – 1-2, а шість компаній взагалі не мають таких спеціалістів. Компанії, які використовують Elixir, відзначили його конкурентність, стабільність та позитивний досвід розробки. Основною перешкодою виявилася обмежена доступність кваліфікованих фахівців на ринку та складність перекваліфікації існуючих розробників. Організації, які не використовують Elixir, пояснили це дефіцитом кадрів та відсутністю внутрішньої потреби.

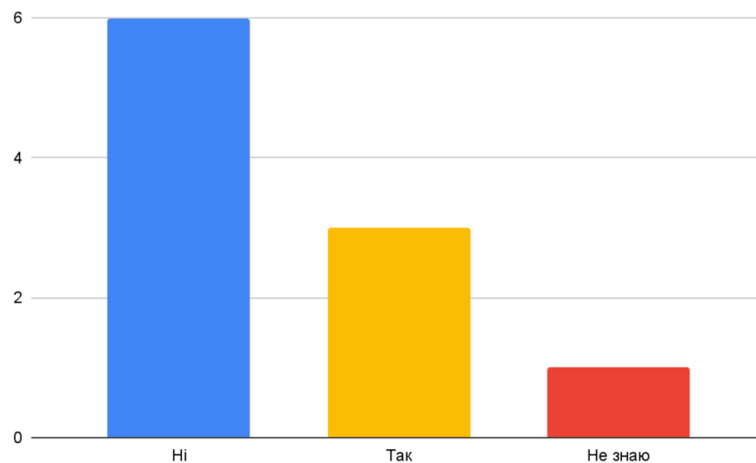


Рис. 1. Чи працюють у вашій компанії інженери Elixir? (n=10)

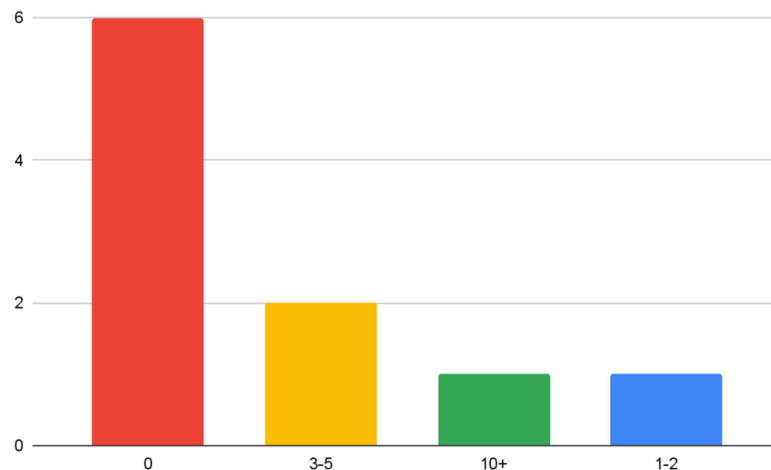


Рис. 2. Чисельність Elixir-команд у компаніях (n=10)

Аналіз опитування та огляд літератури дозволили ідентифікувати типові траєкторії переходу до функціонального програмування. Найбільша частка фахівців (близько третини) приходить з JavaScript, стільки ж – із Ruby, близько чверті – із Python, решта – з Java або PHP. Для кожної групи характерні специфічні труднощі: розробники JavaScript стикаються з відмовою від мутабельності та циклів;

програмісти Ruby адаптуються до використання функцій і модулів замість класів; Python-розробникам необхідно прийняти імутабельність та рекурсію, а Java-фахівцям – переосмислити моделі конкурентності. Ці спостереження підкреслюють необхідність розробки адаптивного навчального контенту, що враховує попередній досвід студентів.

Триетапна модель адаптивного навчання передбачає поступове засвоєння матеріалу з урахуванням індивідуальних потреб студентів:

1. Початковий етап (0–1 рік): Метою є формування базового розуміння функціонального програмування. Навчальний матеріал структуровано в короткі блоки, що супроводжуються інтерактивними практичними завданнями на pattern matching, рекурсію та композицію функцій. Діагностичне тестування на початку курсу визначає необхідність індивідуалізації навчальної траєкторії. Візуалізація переваг Elixir у конкурентності та простоті створення процесів сприяє підтримці мотивації.

2. Проміжний етап (1–3 роки): Опанування OTP-паттернів (GenServer, supervisor trees), архітектурних рішень та фреймворку Phoenix відбувається через проєктну діяльність у парах або малих групах над міні-застосунками. Регулярні контрольні точки (кожні 3–4 тижні) забезпечують корекцію траєкторії навчання, а викладач виконує роль ментора та

фасилітатора, надаючи необхідний зворотний зв'язок.

3. Просунутий етап (понад 3 роки): Акцент робиться на оптимізації, профілюванні та розробці розподілених систем. Студенти реалізують складні проєкти, що імітують реальні сценарії, вивчають методи масштабування та fault tolerance. Peer review та менторство сприяють формуванню професійної спільноти.

Перехід між етапами забезпечується гнучкою системою оцінювання та аналітики успішності, що дозволяє адаптувати складність завдань відповідно до індивідуального прогресу студентів.

Для наочності узагальнені характеристики кожного етапу представлено в табл. 1, яка містить ключові теми, основні навчальні дії та засоби контролю, що дозволяє викладачам оперативно визначати відповідний контент і методи для кожної групи студентів. Використання лаконічних формулювань сприяє швидкому порівнянню етапів навчання.

Таблиця 1

Основні характеристики етапів адаптивної моделі

Етап	Ключові теми	Методи та контроль
Початковий	pattern matching; рекурсія; композиція; pure functions	діагностичні тести; короткі відео; мікрівправи; автоматичний аналіз коду
Проміжний	OTP патерни; Concurrency model; Phoenix framework; TDD	групові проєкти; peer programming; ретроспективи; рев'ю коду
Просунутий	розподілені системи; fault tolerance; profiling; Nerves / LiveView	capstone project; майстер-класи; публічні презентації; рубіжні звіти

Педагогічні стратегії та інструменти включають:

– Діагностичне тестування й персоналізація: Початкове опитування та тест для визначення попереднього досвіду та формування індивідуальних траєкторій навчання.

– Use-Modify-Create: Структурування завдань за принципом використання готового коду, його модифікації та створення власних рішень для поступового освоєння функціонального програмування.

– Проблемно-орієнтовані проєкти: Завершення кожного етапу проєктами, що імітують реальні задачі (від чатів до розподілених сервісів).

– Навчальна аналітика та зворотний зв'язок: Збір даних про успішність, час виконання та типові помилки для генерації персоналізованого зворотного зв'язку та рекомендацій.

– Розвиток спільноти й менторство: Залучення до open-source проєктів, хакатонів та зустрічей з практиками, а також організація менторства для підтримки навчання та розвитку кар'єрних навичок.

– Ефективність запропонованих стратегій підтверджується результатами зовнішніх досліджень. Зокрема, “Charman у ранньому оцінюванні навчання функціональному програмуванню застосовував

метод Use-Modify-Create, що призвело до збільшення балів студентів на етапі Create” приблизно на 9 %, демонструючи переваги поступового переходу від використання готового коду до створення власних рішень [4]. Крім того, систематичний огляд навчальної аналітики у курсах програмування підкреслив необхідність збору детальних когнітивних показників та ідентифікації ефективних і неефективних моделей поведінки для своєчасних корекційних заходів [8].

– Запропонована модель інтегрує компетентнісний та проблемно-орієнтований підходи з адаптивними технологіями, враховуючи індивідуальні траєкторії навчання студентів. Результати опитування ІТ-компаній підтверджують дефіцит фахівців Elixir, що акцентує необхідність університетських курсів, орієнтованих на практичні компетентності. Навчальна аналітика відіграє ключову роль у швидкій ідентифікації та корекції навчальних труднощів. Реалізація адаптивних систем вимагає технічної інфраструктури, підготовки педагогів та забезпечення академічної доброчесності при використанні ІІІ-інструментів, зокрема через прозорість рекомендацій та дотримання конфіденційності даних.

Подальші дослідження передбачають експериментальне впровадження моделі, аналіз її впливу на результати навчання та вивчення потенціалу генеративних ШІ для підтримки студентів.

Розвиток саморегуляції студентів відбувається за допомогою постановки цілей, планування роботи та оцінки прогресу, а викладачі виконують роль фасилітаторів, сприяючи рефлексії та корекції траєкторій навчання. Цифрові компетентності педагогів є критичним фактором успіху, що потребує вміння інтерпретувати дані аналітичних систем та адаптувати контент.

Висновки. Опитування підтверджує наявність дефіциту фахівців, що стимулює необхідність спеціалізованої підготовки. Навчання має враховувати різний попередній досвід програмування (JavaScript, Ruby, Python, Java) та сприяти переходу від імперативних парадигм. Структурована модель з чіткими компетенціями та інтервенціями для початкового, проміжного та просунутого етапів. Діагностичне тестування, метод Use-Modify-Create, проблемно-орієнтоване навчання (PBL), аналітика та менторство формують цілісну систему підтримки студентів.

Перспективу подальших досліджень вбачаємо в експериментальному впровадженні моделі, оцінці її впливу на результати навчання та дослідження можливостей генеративних ШІ для персоналізованого зворотного зв'язку.

ЛІТЕРАТУРА

1. Гуревич Р., Бойчук В., Коношевський Л., Коношевський О., Костенко Н. Використання інноваційних технологій у навчальному процесі. *Молодь і ринок*, 2023. № 5 (213), С. 18–23. DOI: <https://doi.org/10.24919/2308-4634.2023.282829>.
2. Силенко Ю., Романцова Я., Лисицька О., Гарбич Я. Інформаційне освітнє середовище як платформа реалізації інформаційно-комунікаційних технологій у ЗВО. *Молодь і ринок*, 2023. № 6 (214), С. 60–68. DOI: <https://doi.org/10.24919/2308-4634.2023.287944>.
3. Ткачук Г., Медведєва М. ІКТ як засіб формування інформаційно-цифрової компетентності студентів педагогічних університетів. *Молодь і ринок*, 2023. № 1 (209), С. 74–80. DOI: <https://doi.org/10.24919/2308-4634.2023.272479>.
4. Chapman P. Experiences of early assessment to teach functional programming. *Journal of Functional Programming*, 2025. 35. DOI: <https://doi.org/10.1017/s0956796824000182>.
5. du Plooy E., Casteleijn D. & Franzsen D. Personalized adaptive learning in higher education: a scoping review of key characteristics and impact on academic performance and engagement. *Heliyon*, Стаття e39630. 2024. DOI: <https://doi.org/10.1016/j.heliyon.2024.e39630>.
6. Laak K.-J. & Aru J. (2024, 3 квітня). AI and personalized learning: bridging the gap with modern educational goals. *ArXiv*. DOI: <https://doi.org/10.48550/arXiv.2404.02798>.
7. O'Grady M.J. Practical Problem-Based Learning in Computing Education. *ACM Transactions on Computing Education*, 2012. № 12(3), С. 1–16. DOI: <https://doi.org/10.1145/2275597.2275599>.
8. Omer U., Tehseen R., Farooq M.S. & Abid A. Learning analytics in programming courses: Review and impli-

cations. *Education and Information Technologies*. 2023. DOI: <https://doi.org/10.1007/s10639-023-11611-0>.

9. Yao D. & Lin J. Cognitive enhancement through competency-based programming education: a 12-year longitudinal study. *Education and Information Technologies*. 2025. DOI: <https://doi.org/10.1007/s10639-025-13582-w>.

REFERENCES

1. Gurevych, R., Boichuk, V., Konoshevskiy, L., Konoshevskiy, O. & Kostenko, N. (2023). Vykorystannia innovatsiynykh tekhnolohii u navchalnomu protsesi [Use of innovative technologies in the educational process]. *Youth & market*, No. 5 (213), pp. 18–23. DOI: <https://doi.org/10.24919/2308-4634.2023.282829> (Accessed 16 Nov. 2025). [in Ukrainian].
2. Sylenko, Yu., Romantsova, Ya., Lysytska, O. & Harbych, Ya. (2023). Informatsiine osvittne seredovyshche yak platforma realizatsii informatsiino-komunikatsiynykh tekhnolohii u ZVO [Information educational environment as a platform for the implementation of information and communication technologies in an institution of higher education]. *Youth & market*, No. 6(214), pp. 60–68. DOI: <https://doi.org/10.24919/2308-4634.2023.287944> (Accessed 16 Nov. 2025). [in Ukrainian].
3. Tkachuk, H. & Medvedieva, M. (2023). IKT yak zasib formuvannia informatsiino-tyfrovoi kompetentnosti studentiv pedahohichnykh universytetiv [ICT as a tool for forming the information and digital students' competence in pedagogical universities]. *Youth & market*, No. 1 (209), pp. 74–80. DOI: <https://doi.org/10.24919/2308-4634.2023.272479> (Accessed 16 Nov. 2025). [in Ukrainian].
4. Chapman, P. (2025). Experiences of early assessment to teach functional programming. *Journal of Functional Programming*, 35. DOI: <https://doi.org/10.1017/s0956796824000182> (Accessed 16 Nov. 2025). [in English].
5. du Plooy, E., Casteleijn, D. & Franzsen, D. (2024). Personalized adaptive learning in higher education: a scoping review of key characteristics and impact on academic performance and engagement. *Heliyon*, Article e39630. DOI: <https://doi.org/10.1016/j.heliyon.2024.e39630> (Accessed 16 Nov. 2025). [in English].
6. Laak, J., & Aru, J. (2024, April 3). AI and personalized learning: bridging the gap with modern educational goals. *ArXiv*. DOI: <https://doi.org/10.48550/arXiv.2404.02798> (Accessed 16 Nov. 2025). [in English].
7. O'Grady, M. J. (2012). Practical Problem-Based Learning in Computing Education. *ACM Transactions on Computing Education*, No. 12(3), pp. 1–16. DOI: <https://doi.org/10.1145/2275597.2275599> (Accessed 16 Nov. 2025). [in English].
8. Omer, U., Tehseen, R., Farooq, M.S. & Abid, A. (2023). Learning analytics in programming courses: Review and implications. *Education and Information Technologies*. DOI: <https://doi.org/10.1007/s10639-023-11611-0> (Accessed 16 Nov. 2025). [in English].
9. Yao, D. & Lin, J. (2025). Cognitive enhancement through competency-based programming education: a 12-year longitudinal study. *Education and Information Technologies*. DOI: <https://doi.org/10.1007/s10639-025-13582-w> (Accessed 16 Nov. 2025). [in English].

Стаття надійшла до редакції 26.11.2025