

Секція АЗ «Початкова освіта»	
УДК 378.147:373.3.011.3-051:004.021	
Дата першого надходження статті до видання	2026-06-30
Дата прийняття статті до друку після рецензування	2026-07-30
Дата публікації/оприлюднення	2026-07-30

ПРОЄКТУВАННЯ НАВЧАЛЬНОГО МОДУЛЯ З МЕТОДИКИ АЛГОРИТМІЗАЦІЇ У ПРОФЕСІЙНІЙ ПІДГОТОВЦІ ВЧИТЕЛЯ ПОЧАТКОВИХ КЛАСІВ

Василиків Іван Богданович

кандидат педагогічних наук, доцент,

доцент кафедри фундаментальних дисциплін початкової освіти,
Дрогобицький державний педагогічний університет імені Івана Франка,

Дрогобич, Україна

<https://orcid.org/0000-0002-9220-1736>

Анотація. У статті обґрунтовано зміст і логіку проєктування навчального модуля з методики алгоритмізації для професійної підготовки майбутніх учителів початкових класів. Актуальність дослідження визначається суперечністю між наявністю алгоритмічної змістової лінії в початковій освіті та недостатньою цілісністю університетської підготовки до її методичного забезпечення. Метою статті є розроблення структурно-функціональної моделі навчального модуля, у якій результати навчання, змістові блоки, професійно орієнтовані завдання, навчальні продукти та критерії оцінювання утворюють узгоджену систему. Методологію становлять компетентнісний, діяльнісний, конструктивістський і дизайн-орієнтований підходи. Використано аналіз нормативних документів і наукових джерел, концептуальне моделювання, педагогічне проєктування, структурно-логічний аналіз та експертно-аналітичне зіставлення складників модуля. Результатом дослідження є модуль обсягом три кредити ЄКТС, побудований за циклом «розпізнати алгоритмічну ідею — виконати й пояснити — трансформувати для дитини — спроектувати навчальну ситуацію — апробувати — відрефлексувати». Модуль охоплює п'ять блоків: нормативно-концептуальний, предметно-алгоритмічний, методико-трансформаційний, цифрово-проєктувальний та рефлексивно-оцінювальний. Для кожного блоку визначено очікувані результати, види діяльності та матеріальний освітній продукт. Запропоновано матрицю професійних завдань, що забезпечує перехід студента від виконання готового алгоритму до самостійного конструювання системи завдань для 1–4 класів. Розроблено критеріальну рубрику оцінювання за чотирма параметрами: предметна коректність, вікова й методична доцільність, якість навчальної взаємодії та рефлексивне вдосконалення. Наукова новизна полягає в узгодженні змісту модуля не з переліком цифрових сервісів, а з повним циклом професійної дії вчителя. Практична цінність результатів полягає в можливості використання запропонованої моделі під час оновлення освітніх компонентів спеціальності 013 «Початкова освіта», організації мікрОВикладання та педагогічної практики.

Ключові слова: навчальний модуль, методика алгоритмізації, майбутній учитель початкових класів, професійна підготовка, алгоритмічне мислення, педагогічне проєктування, вправи unplugged, Scratch, мікрОВикладання, критеріальне оцінювання.

DESIGNING A TRAINING MODULE ON ALGORITHMIZATION METHODOLOGY IN THE PROFESSIONAL EDUCATION OF PRIMARY SCHOOL TEACHERS

Ivan Bohdanovych Vasylykiv

Candidate of Pedagogical Sciences, Associate Professor,
Associate Professor at the Department of Fundamental Disciplines of Primary Education,
Drohobych Ivan Franko State Pedagogical University,
Drohobych, Ukraine
<https://orcid.org/0000-0002-9220-1736>

Abstract. The article substantiates the content and design logic of a training module on algorithmization methodology for the professional education of future primary school teachers. The relevance of the study is determined by the contradiction between the algorithmic content line in primary education and the insufficient coherence of university preparation for its methodological implementation. The aim of the article is to develop a structural and functional model of a training module in which learning outcomes, content units, profession-oriented tasks, learning artefacts, and assessment criteria form an aligned system. The methodology combines competence-based, activity-based, constructivist, and design-oriented approaches. The methods used include analysis of regulatory documents and research publications, conceptual modelling, pedagogical design, structural-logical analysis, and expert-analytical comparison of the module components. The result of the study is a three-ECTS-credit module organised around the cycle 'recognise an algorithmic idea — execute and explain — transform it for a child — design a learning situation — test it — reflect and improve'. The module comprises five units: regulatory-conceptual, subject-algorithmic, methodological-transformation, digital-design, and reflective-assessment. For each unit, expected outcomes, learning activities, and a tangible educational artefact are defined. A matrix of professional tasks is proposed to guide the student from executing a ready-made algorithm to independently constructing a coherent set of tasks for grades 1–4. An analytic assessment rubric is developed according to four parameters: subject accuracy, age and methodological appropriateness, quality of learning interaction, and reflective improvement. The scientific novelty lies in aligning the module content not with a list of digital services but with the complete cycle of a teacher's professional action. The practical value of the results consists in the possibility of applying the proposed model to update educational components of specialty 013 Primary Education and to organise microteaching and teaching practice.

Keywords: training module, algorithmization methodology, future primary school teacher, professional education, algorithmic thinking, pedagogical design, unplugged activities, Scratch, microteaching, criteria-based assessment.

Вступ

Актуальність проблеми. Цифрова трансформація початкової освіти актуалізувала не стільки потребу раннього навчання програмування, скільки необхідність формувати в дітей уміння впорядковувати дії, бачити зв'язок між умовою та результатом, знаходити повторювані фрагменти, перевіряти створений план і виправляти помилки. У Державному стандарті початкової освіти ці вміння представлені через роботу з командами, виконавцями, послідовностями, простими моделями та цифровими середовищами [5; 8]. Вони мають міжпредметний характер, оскільки виявляються під час розв'язування математичної задачі, виконання технологічної інструкції, планування досліду, побудови маршруту або створення цифрової історії. Тому майбутній учитель повинен розуміти алгоритмізацію одночасно

як зміст інформатичної галузі, спосіб організації пізнавальної діяльності й ресурс розвитку самоконтролю молодшого школяра.

Університетська підготовка нерідко розподіляє необхідні знання між кількома дисциплінами: педагогіка пояснює закономірності навчання, психологія — вікові можливості дитини, методика інформатики — предметний зміст, а цифрові практикуми — роботу з програмними засобами. За відсутності спеціально спроектованого інтегровального модуля студент може успішно виконати завдання у Scratch, але мати труднощі з поясненням алгоритмічної ідеї семирічній дитині, добором пропедевтичної вправи без комп'ютера, передбаченням типової помилки або формулюванням критеріїв успіху. Отже, проблема полягає не в кількості вивчених інструментів, а в розриві між предметною дією студента та її методичною трансформацією у навчальну діяльність учня.

Аналіз останніх досліджень і публікацій. Дослідження цифрової компетентності майбутніх педагогів наголошують на інтеграції технологічного, методичного й рефлексивного складників [1–3; 7]. Рамка DigCompEdu пов'язує цифрову компетентність учителя з добором ресурсів, організацією навчання, оцінюванням та підтримкою активності здобувачів освіти [12]. Зарубіжні роботи з обчислювального мислення доводять, що підготовка вчителя має включати не лише поняття декомпозиції, абстрагування, алгоритму й налагодження, а й досвід проєктування педагогічних сценаріїв [13–16]. Систематичні огляди Scratch і вправ unplugged виявляють їхню результативність за умови цілеспрямованого методичного супроводу, поступового ускладнення та рефлексії [11; 17; 18].

Виділення невирішеної частини проблеми. Попередня авторська публікація була зосереджена на особливостях формування алгоритмічного мислення молодших школярів на уроках інформатики [19]. У цій статті предмет дослідження принципово інший: не процес навчання учнів, а архітектура окремого університетського модуля, який готує студента до такого навчання. У наукових джерелах достатньо описані цифрові компетентності та окремі засоби, однак бракує конструкції, що пов'язує очікуваний професійний результат, послідовність навчальних дій, обов'язковий продукт студента й прозорий спосіб оцінювання. Невирішеним залишається також питання, як забезпечити перехід від засвоєння алгоритмічних понять до їх дидактичного перетворення відповідно до віку, класу та навчальної ситуації.

Мета статті. Метою статті є теоретичне обґрунтування й проєктування навчального модуля з методики алгоритмізації у професійній підготовці вчителя початкових класів. Для досягнення мети необхідно: визначити проєктні принципи модуля; сформулювати його інтегрований результат навчання; побудувати змістову й діяльнісну послідовність; установити навчальні продукти, що засвідчують сформованість професійної дії; розробити матрицю завдань та критеріальну рубрику оцінювання; окреслити способи інтеграції модуля з педагогічною практикою.

Наукова новизна. Уперше для підготовки майбутнього вчителя початкових класів запропоновано проєктну модель навчального модуля, у якій одиницею структурування виступає не тема або цифровий сервіс, а завершена професійна дія. Новим є узгодження п'яти змістових блоків із циклом методичної трансформації алгоритмічного змісту, обов'язковими продуктами студента та критеріями їх оцінювання. Матриця ускладнення завдань відображає перехід від розпізнавання алгоритмічної ідеї до апробації й доказового редизайну навчальної ситуації.

Практичне значення. Запропоновану структуру можна використати під час оновлення робочої програми освітнього компонента з методики навчання інформатичної галузі, розроблення вибіркового курсу або практикуму перед педагогічною практикою. Таблиці модуля придатні як основа календарно-тематичного

планування, завдань для самостійної роботи, мікрОВикладання, портфоліо та критеріального оцінювання студентських проєктів.

Методологія

Методи дослідження. Дослідження виконано як теоретичне педагогічне проєктування. Компетентнісний підхід задає кінцевий орієнтир — здатність майбутнього вчителя самостійно створити, провести й удосконалити навчальну ситуацію з алгоритмізації. Діяльнісний підхід визначає спосіб досягнення результату через послідовність професійно наближених дій. Конструктивістський підхід передбачає, що поняття алгоритму, виконавця, розгалуження чи циклу не повідомляються у завершеному вигляді, а реконструюються студентами під час аналізу задач і помилок. Дизайн-орієнтований підхід використано для узгодження цілей, навчальної активності, продуктів і оцінювання.

Джерела даних. Джерельний корпус охопив нормативні документи початкової та вищої освіти України [5; 6; 8], дослідження цифрової компетентності педагогів [1–4; 7; 12], праці з підготовки вчителів до розвитку обчислювального мислення [13–16], а також систематичні огляди використання Scratch і безкомп'ютерних вправ [11; 17; 18]. Окремо враховано попереднє дослідження алгоритмічного мислення молодших школярів [19], щоб чітко розмежувати вже опублікований учнівський аспект і новий аспект професійного проєктування модуля.

Процедура проєктування складалася з п'яти кроків. Спочатку нормативні результати початкової освіти було переведено у перелік професійних дій учителя. Далі для кожної дії визначено знання, способи виконання й типові труднощі студента. На третьому кроці дії згруповано в логічні змістові блоки та впорядковано від розпізнавання алгоритмічної ідеї до створення власного навчального фрагмента. На четвертому кроці для кожного блоку встановлено обов'язковий продукт, який можна спостерігати й оцінити. Завершальний крок полягав у створенні аналітичної рубрики та перевірці внутрішньої узгодженості моделі за ланцюгом «результат — діяльність — продукт — критерій».

Інструменти аналізу. Для аналізу нормативних результатів застосовано матрицю «дія учня — об'єкт дії — умови виконання — доказ досягнення». Внутрішню узгодженість модуля перевірено за картою конструктивного узгодження «результат навчання — діяльність студента — освітній продукт — критерій оцінювання». Професійні завдання зіставлено за рівнем автономії, складністю методичної трансформації та кількістю репрезентацій алгоритмічної ідеї. Для підсумкового проєкту створено аналітичну рубрику з описами початкового, достатнього й високого рівнів за чотирма критеріями.

Обмеження дослідження. Запропонована модель не подається як емпірично доведена технологія. Її статус — теоретично обґрунтований проєкт, придатний до подальшого експериментального випробування. Це обмеження враховано під час формулювання висновків: стаття описує структуру, механізм і критерії реалізації, але не приписує модулю статистично підтвердженого впливу на результати студентів.

Результати

Інтегрований результат модуля сформульовано як здатність студента спроектувати, обґрунтувати, провести та вдосконалити фрагмент навчання алгоритмізації для учнів 1–4 класів, поєднавши предметну коректність, вікову доступність, діяльнісний спосіб навчання, доцільний цифровий або безкомп'ютерний засіб і формувальне оцінювання. Таке формулювання принципово відрізняється від результатів на кшталт «знати види алгоритмів» або «вміти працювати у Scratch», оскільки містить повний цикл професійної дії та передбачає створення продукту, придатного до аналізу.

Модуль розраховано на три кредити ЄКТС — 90 годин, із яких 30 годин становить контактна робота, а 60 годин — керована самостійна діяльність. Обсяг не є випадковим: він дає змогу пройти повний цикл від концептуального орієнтування до мікрОВикладання й повторного проєктування. Модуль може функціонувати як змістова частина дисципліни «Методика навчання інформатичної освітньої галузі», як вибіркового компонента або як міждисциплінарний практикум перед педагогічною практикою.

Таблиця 1. Структура навчального модуля з методики алгоритмізації

Блок і орієнтовний обсяг	Провідне питання	Діяльність студента	Підсумковий продукт
1. Нормативно-концептуальний, 12 год	Які результати алгоритмічної змістової лінії має забезпечити вчитель?	Аналіз стандарту, програм і підручників; побудова карти результатів для 1–4 класів.	Карта наступності результатів навчання.
2. Предметно-алгоритмічний, 18 год	Які способи мислення стоять за алгоритмічними поняттями?	Розв'язування, пояснення й налагодження лінійних, розгалужених і циклічних алгоритмів.	Пояснювальний словник понять із прикладами й контрприкладми.
3. Методико-трансформаційний, 24 год	Як перетворити предметну дію на доступну діяльність молодшого школяра?	Проєктування вправ unplugged, ігор, інструкцій, запитань і диференційованої підтримки.	Система завдань для обраного класу.
4. Цифрово-проєктувальний, 20 год	Коли цифрове середовище справді покращує навчання?	Створення моделі або історії у Scratch; зіставлення цифрового й безкомп'ютерного сценаріїв.	Цифровий мініпроєкт і методичний паспорт до нього.
5. Рефлексивно-оцінювальний, 16 год	Як побачити процес мислення учня та вдосконалити власне рішення?	МікрОВикладання, спостереження, взаємооцінювання, аналіз відеофрагмента, редизайн.	Портфоліо: план, матеріали, запис апробації, самоаналіз і виправлена версія.

Джерело: розроблено автором.

Перший блок створює нормативну рамку модуля. Студенти не переписують вимоги стандарту, а встановлюють, як змінюється складність алгоритмічної діяльності від 1 до 4 класу. Для цього вони позначають дієслова результатів навчання, визначають об'єкт дії, умови виконання та можливе свідчення досягнення. Наприклад, результат «складає послідовність дій» конкретизується через тип ситуації, кількість кроків, наявність чи відсутність готового набору команд, спосіб пояснення та перевірки. Такий аналіз запобігає випадковому добору ефектних, але програмово невмотивованих завдань.

У предметно-алгоритмічному блоці акцент переноситься з термінів на операції мислення. Студент має вміти розкласти складну дію на кроки, відокремити істотні умови, визначити виконавця і допустимі команди, побачити повторення, сформулювати умову вибору та перевірити результат. Кожне поняття опрацьовується

через триаду «приклад — контрприклад — помилка». Так, під час вивчення циклу студент не лише складає повторювану послідовність, а й пояснює, коли повторення недоцільне, чому цикл може не завершитися та як виявити зайву або пропущену дію.

Методико-трансформаційний блок є центральним, оскільки саме тут предметне знання набуває педагогічної форми. Проектування починається з прогнозованої учнівської дії, а не з назви цифрового сервісу. Студент визначає, що дитина має помітити, виконати, пояснити й перевірити; після цього добирає сюжет, матеріали, форму взаємодії та спосіб підтримки. Одна алгоритмічна ідея подається щонайменше у трьох репрезентаціях: тілесно-руховій, предметно-графічній та цифровій. Перехід між репрезентаціями допомагає відокремити зміст поняття від конкретного інтерфейсу.

Вправи unplugged використовуються не як розважальна пауза, а як повноцінна модель алгоритмічної дії. Їхня якість визначається наявністю однозначного виконавця, системи команд, очікуваного результату, можливості перевірки й педагогічно продуктивної помилки. Наприклад, маршрут клітинним полем може бути засобом вивчення послідовності, але стає вправою на налагодження лише тоді, коли учень отримує некоректний маршрут, локалізує помилкову команду, обґрунтовує зміну та повторно перевіряє виконання.

Цифрово-проектувальний блок побудовано за принципом функціональної доцільності. Scratch використовується тоді, коли візуальні блоки дають змогу швидко змінювати послідовність, спостерігати виконання, бачити наслідок умови чи повторення та створювати відкритий творчий продукт [10; 17]. Якщо мету краще досягти за допомогою карток, руху або предметної моделі, цифрове середовище не є обов'язковим. Студент обґрунтовує вибір засобу в методичному паспорті, указуючи навчальну функцію кожного використаного елемента.

Проектною одиницею модуля є професійне завдання, а не інформаційна тема. Воно містить педагогічну ситуацію, обмеження, ресурс і продукт. Наприклад: «Учні третього класу виконують готові скрипти, але не можуть пояснити роль умови. Розробіть дві взаємопов'язані вправи — без комп'ютера і у Scratch — та критерії, за якими буде видно розуміння розгалуження». Така конструкція змушує студента одночасно застосувати предметне знання, урахувати вік, створити навчальну послідовність і передбачити доказ результату.

Таблиця 2. Матриця ускладнення професійних завдань у межах модуля

Рівень професійної дії	Що виконує студент	Типове завдання	Ознака переходу на наступний рівень
1. Розпізнавання	Виявляє алгоритмічну ідею та нормативний результат.	Визначити, які вміння формує запропонована вправа.	Аргументовано пов'язує дію учня з результатом програми.
2. Пояснення	Виконує алгоритм, вербалізує кроки й знаходить помилки.	Пояснити помилковий скрипт мовою, доступною дитині.	Пояснення є точним, коротким і не підмінює дію учня.
3. Трансформація	Адаптує одну ідею для різного віку та репрезентації.	Створити варіанти вправи для 2 і 4 класів.	Змінює не лише складність тексту, а й характер учнівської самостійності.
4. Проектування	Створює послідовність дій, підтримку й оцінювання.	Розробити фрагмент уроку з переходом unplugged → Scratch.	Усі елементи підпорядковані одному результату навчання.
5. Апробація та редизайн	Проводить мікрОВикладання, збирає докази й удосконалює продукт.	Після взаємооцінювання переробити завдання та пояснити зміни.	Редизайн спирається на спостережувані труднощі, а не на загальне враження.

Джерело: авторська розробка.

МікрОВикладання організовується як короткий цикл апробації тривалістю 10–15 хвилин. Один студент виконує роль учителя, частина групи — роль учнів із заданими навчальними профілями, решта спостерігає за одним визначеним аспектом: якістю інструкції, характером запитань, реакцією на помилку або використанням критеріїв успіху. Після проведення обговорюється не особистість студента, а зв'язок між проєктним рішенням і спостережуваною діяльністю «учнів». Обов'язковою є друга версія фрагмента, оскільки без редизайну рефлексія залишається декларативною.

Формувальне оцінювання в модулі має дві спрямованості. По-перше, студент навчається бачити прогрес молодшого школяра: чи може дитина відтворити послідовність, пояснити зв'язок між кроками, передбачити результат, знайти помилку та перенести спосіб дії в іншу ситуацію. По-друге, оцінюється професійне рішення самого студента. Для цього застосовуються чек-листи, короткі аналітичні коментарі, взаємооцінювання за одним критерієм, порівняння першої та другої версій продукту й підсумкова аналітична рубрика.

Таблиця 3. Аналітична рубрика оцінювання підсумкового проєкту

Критерій	Початковий рівень	Достатній рівень	Високий рівень
Предметна коректність	Є неточності в командах, умовах, повтореннях або очікуваному результаті.	Алгоритм коректний; окремі пояснення потребують уточнення.	Коректність поєднана з передбаченням контрприкладів і типових помилок.
Вікова та методична доцільність	Завдання перевантажене, репродуктивне або не відповідає результату.	Завдання доступне й узгоджене з метою, але варіативність підтримки обмежена.	Передбачено поступове ускладнення, диференціацію та кілька репрезентацій.
Організація навчальної взаємодії	Інструкція домінує над самостійною дією; критерії неясні.	Є запитання, взаємодія та зрозумілі критерії успіху.	Запитання стимулюють пояснення, перевірку, дискусію й самокорекцію.
Рефлексія та редизайн	Зміни косметичні або не обґрунтовані доказами.	Виправлено помічені труднощі й пояснено основні зміни.	Редизайн системний; використано дані спостереження, зворотний зв'язок і самоаналіз.

Джерело: розроблено автором на основі логіки критеріально орієнтованого оцінювання.

Підсумкове портфоліо складається з карти програмового результату, системи щонайменше трьох взаємопов'язаних завдань, матеріалів для учнів, цифрового або безкомп'ютерного прототипу, критеріїв успіху, запису чи протоколу мікрОВикладання, отриманого зворотного зв'язку та виправленої версії. Така сукупність демонструє не одноразове виконання, а траєкторію професійного рішення. Портфоліо також полегшує інтеграцію з педагогічною практикою: окремі матеріали можуть бути апробовані в реальному класі за погодженням із учителем-наставником.

Узгодженість модуля забезпечується тим, що кожний результат має спостережуваний продукт, а кожний критерій застосовується не лише наприкінці, а впродовж роботи. Якщо студент має навчитися адаптувати завдання до віку, він створює дві версії одного задуму; якщо має навчитися реагувати на помилку, аналізує

помилковий алгоритм і проводить відповідний епізод мікрОВикладання; якщо має опанувати рефлексію, подає не опис вражень, а зіставлення первинного рішення, даних спостереження й виконаної зміни. Таким чином зменшується розрив між заявленими компетентностями та реальним змістом навчання.

Обговорення

Інтерпретація результатів. Розроблена модель показує, що підготовка до навчання алгоритмізації стає цілісною лише тоді, коли предметна дія послідовно перетворюється на педагогічне рішення. П'ять блоків модуля не є незалежними тематичними частинами: кожний наступний використовує продукт попереднього. Карта нормативних результатів визначає зміст предметних вправ; пояснювальний словник допомагає створити віково доступні інструкції; система завдань переходить у цифровий або безкомп'ютерний прототип; апробація дає дані для редизайну. Отже, результатом є не накопичення матеріалів, а документована траєкторія професійного рішення. Значення матриці завдань полягає в керованому зростанні автономії студента: від аналізу готового прикладу до відповідальності за мету, спосіб навчання, оцінювання та виправлення власного проєкту.

Порівняння з іншими дослідженнями. Запропонована конструкція підтримує висновки про необхідність інтегрувати обчислювальне мислення в педагогічну освіту [13–15], але конкретизує механізм такої інтеграції через професійні продукти й мікрОВикладання. Вона узгоджується з DigCompEdu [12], оскільки поєднує створення ресурсів, організацію навчання, оцінювання та розвиток цифрової компетентності учнів. Результати систематичних оглядів підтверджують потенціал вправ unplugged і Scratch [11; 17], тоді як застереження щодо кодоцентричного трактування обчислювального мислення [18] обґрунтовує відмову від побудови модуля навколо одного програмного середовища. На відміну від попередньої публікації, присвяченої формуванню алгоритмічного мислення молодших школярів [19], ця робота досліджує архітектуру університетської підготовки й критерії якості професійного рішення майбутнього вчителя.

Наукова новизна. Наукова новизна полягає у створенні структурно-функціональної моделі, що поєднує нормативний, предметний, методико-трансформаційний, цифрово-проектувальний і рефлексивно-оцінювальний блоки єдиним циклом професійної дії. На відміну від тематичного komponування курсу, запропоновано діяльнісний принцип: кожний блок завершується спостережуваним продуктом, а його якість визначається наперед заданими критеріями. Уперше для цього модуля розроблено матрицю п'яти рівнів професійної автономії — розпізнавання, пояснення, трансформація, проєктування, апробація та редизайн. Новим є також одночасне оцінювання предметної коректності, вікової доцільності, організації взаємодії та доказовості внесених змін. Це дає змогу операціоналізувати поняття методичної готовності та підготувати його до подальшої емпіричної перевірки.

Практичне значення. Модель можна безпосередньо адаптувати до освітніх програм спеціальності 013 «Початкова освіта» як модуль обсягом три кредити ЄКТС, частину обов'язкової дисципліни або вибіркового практикум. Таблиця структури визначає розподіл годин і підсумкові продукти; матриця професійних завдань допомагає будувати послідовність аудиторної та самостійної роботи; аналітична рубрика забезпечує прозоре оцінювання підсумкового проєкту. Портфоліо може бути інтегроване з педагогічною практикою: студент апробує окремих фрагмент у класі, фіксує труднощі учнів, отримує зворотний зв'язок учителя-наставника та подає виправлену версію. Для викладача модель створює основу моніторингу не лише кінцевого продукту, а й розвитку професійного мислення студента.

Висновки

У результаті дослідження розроблено структурно-функціональну модель навчального модуля з методики алгоритмізації для професійної підготовки майбутнього вчителя початкових класів. Її системотвірним елементом є повний цикл професійної дії: розпізнавання алгоритмічної ідеї, предметне виконання й пояснення, адаптація для молодшого школяра, проєктування навчальної ситуації, апробація, аналіз доказів та редизайн. Модуль обсягом три кредити ЄКТС охоплює п'ять взаємопов'язаних блоків і завершується портфоліо, що документує не лише кінцевий результат, а й процес удосконалення.

Наукова новизна запропонованого рішення полягає в узгодженні результатів навчання, професійних завдань, матеріальних продуктів і критеріїв оцінювання навколо методичної трансформації алгоритмічного змісту. Розроблена матриця демонструє поступове зростання автономії студента, а аналітична рубрика робить оцінювання прозорим і спрямованим на якість педагогічного рішення. Модель відрізняється від попереднього дослідження формування алгоритмічного мислення учнів [19], оскільки її предметом є проєктування університетського освітнього компонента.

Практичне значення полягає в можливості включити модуль до обов'язкових або вибіркових компонентів спеціальності 013 «Початкова освіта», використати його завдання під час методичних практикумів і поєднати портфоліо з педагогічною практикою. Подальша робота має бути спрямована на емпіричне випробування моделі, уточнення індикаторів рубрики та визначення того, які поєднання безкомп'ютерних і цифрових форматів найкраще підтримують здатність студентів створювати віково доцільні й методично цілісні завдання.

Список використаних джерел

1. Бахмат Н. В., Сторчова Т. В., Моцик Р. В., Мелекесцева Н. В., Братиця Г. Г. Сучасні тенденції розвитку цифрової компетентності майбутніх учителів: європейський досвід. Академічні візії. 2023. Вип. 15. <https://doi.org/10.5281/zenodo.7593637>.
2. Бойчук Ю. Д., Боярська-Хоменко А. В., Доценко С. О. Формування цифрової компетентності майбутніх учителів: досвід Харківського національного педагогічного університету імені Г. С. Сковороди. Дидактика. 2021. Вип. 1. С. 7–13.
3. Жерновникова О. А., Перетяга Л. Є., Ковтун А. В. та ін. Технологія формування цифрової компетентності майбутніх учителів засобами гейміфікації. Інформаційні технології і засоби навчання. 2020. Т. 75, № 1. С. 170–185. <https://doi.org/10.33407/itlt.v75i1.3036>.
4. Овчарук О. В., Гриценчук О. О., Іванюк І. В. та ін. Розвиток інформаційно-комунікаційної компетентності вчителів в умовах євроінтеграційних процесів в освіті: політики та практики. Київ : Педагогічна думка, 2019. 203 с.
5. Про затвердження Державного стандарту початкової освіти : Постанова Кабінету Міністрів України від 21.02.2018 № 87. URL: <https://zakon.rada.gov.ua/laws/show/87-2018-%D0%BF#Text>.
6. Про затвердження стандарту вищої освіти за спеціальністю 013 «Початкова освіта» для першого (бакалаврського) рівня вищої освіти : Наказ Міністерства освіти і науки України від 23.03.2021 № 357.
7. Стойка О. Я. Формування інформаційно-цифрової компетентності майбутніх учителів в умовах дистанційного навчання. Педагогічні науки: теорія та практика. 2023. № 2 (46). С. 66–72. <https://doi.org/10.26661/2786-5622-2023-2-09>.
8. Типові освітні програми для 1–4 класів закладів загальної середньої освіти. Міністерство освіти і науки України. URL: <https://mon.gov.ua/osvita-2/zagalna-serednya-osvita/osvitni-programi/tipovi-osvitni-programi-dlya-1-4-klasiv>.

9. Шкуренко О. В., Смик А. О. Методичні особливості вивчення теми «Алгоритми і виконавці» у початковій школі. Молодий вчений. 2024. № 7 (131). С. 8–11. <https://doi.org/10.32839/2304-5809/2024-7-131-2>.
10. Яценко О. І., Чумак Л. Ф. Критерії добору середовища навчання програмування для формування ІКТ-компетентності майбутніх учителів початкової школи. Інформаційні технології і засоби навчання. 2020. Т. 78, № 4. С. 225–239. <https://doi.org/10.33407/itlt.v78i4.3469>.
11. Chen P., Yang D., Metwally A. H. S., Lavonen J., Wang X. Fostering computational thinking through unplugged activities: A systematic literature review and meta-analysis. International Journal of STEM Education. 2023. Vol. 10. Art. 47. <https://doi.org/10.1186/s40594-023-00434-7>.
12. Redecker C. European Framework for the Digital Competence of Educators: DigCompEdu. Luxembourg : Publications Office of the European Union, 2017. <https://doi.org/10.2760/159770>.
13. Rodrigues R. N., Andrade W. L., Sampaio L. M. R., Campos L. M. R. S. Integration of computational thinking in initial teacher education: A systematic review. Frontiers in Education. 2024. Vol. 9. Art. 1330065. <https://doi.org/10.3389/feduc.2024.1330065>.
14. Yadav A., Mayfield C., Zhou N., Hambrusch S., Korb J. T. Computational thinking in elementary and secondary teacher education. ACM Transactions on Computing Education. 2014. Vol. 14, No. 1. Art. 5. <https://doi.org/10.1145/2576872>.
15. Yadav A., Stephenson C., Hong H. Computational thinking for teacher education. Communications of the ACM. 2017. Vol. 60, No. 4. P. 55–62. <https://doi.org/10.1145/2994591>.
16. Shute V. J., Sun C., Asbell-Clarke J. Demystifying computational thinking. Educational Research Review. 2017. Vol. 22. P. 142–158. <https://doi.org/10.1016/j.edurev.2017.09.003>.
17. Fagerlund J., Häkkinen P., Vesisenaho M., Viiri J. Computational thinking in programming with Scratch in primary schools: A systematic review. Computer Applications in Engineering Education. 2021. Vol. 29, No. 1. P. 12–28. <https://doi.org/10.1002/cae.22255>.
18. Kite V., Park S., Wiebe E. The code-centric nature of computational thinking education: A review of trends and issues in computational thinking education research. SAGE Open. 2021. Vol. 11, No. 2. <https://doi.org/10.1177/21582440211016418>.
19. Василиків І. Б., Романчук Р. М. Особливості формування алгоритмічного мислення молодших школярів на уроках інформатики. Молодь і ринок. 2021. № 2 (188). С. 90–95. <https://doi.org/10.24919/2308-4634.2021.230493>.

References

- Bakhmat, N. V., Storchova, T. V., Motsyk, R. V., Mieliekiesceva, N. V., & Bratytsia, H. H. (2023). Suchasni tendentsii rozvytku tsyfrovoi kompetentnosti maibutnikh uchyteliv: Yevropeyskyi dosvid [Current trends in the development of future teachers' digital competence: European experience]. *Akademichni vizii*, 15. <https://doi.org/10.5281/zenodo.7593637>
- Boichuk, Y. D., Boiarska-Khomenko, A. V., & Dotsenko, S. O. (2021). Formuvannia tsyfrovoi kompetentnosti maibutnikh uchyteliv [Formation of future teachers' digital competence]. *Dydaktyka*, 1, 7–13.
- Zhernovnykova, O. A., Peretiaha, L. Y., Kovtun, A. V., et al. (2020). Tekhnolohiia formuvannia tsyfrovoi kompetentnosti maibutnikh uchyteliv zasobamy heimifikatsii [Technology for developing future teachers' digital competence through gamification]. *Informatsiini tekhnolohii i zasoby navchannia*, 75(1), 170–185. <https://doi.org/10.33407/itlt.v75i1.3036>
- Ovcharuk, O. V., Hrytsenchuk, O. O., Ivaniuk, I. V., et al. (2019). Rozvytok informatsiino-komunikatsiinoi kompetentnosti vchyteliv v umovakh yevrointehratsiinykh protsesiv v osviti [Development of teachers' information and communication competence in European integration processes]. *Pedahohichna dumka*.

- Cabinet of Ministers of Ukraine. (2018). Pro zatverdzhennia Derzhavnoho standartu pochatkovoï osvity [On approval of the State Standard of Primary Education] (Resolution No. 87). <https://zakon.rada.gov.ua/laws/show/87-2018-%D0%BF#Text>
- Ministry of Education and Science of Ukraine. (2021). Pro zatverdzhennia standartu vyshchoï osvity za spetsialnistiu 013 "Pochatkova osvita" [On approval of the higher education standard for specialty 013 Primary Education] (Order No. 357).
- Stoika, O. Y. (2023). Formuvannia informatsiino-tsyfrovoi kompetentnosti maibutnikh uchyteliv v umovakh dystantsiinoho navchannia [Formation of future teachers' information and digital competence in distance learning]. *Pedahohichni nauky: teoriia ta praktyka*, 2(46), 66–72. <https://doi.org/10.26661/2786-5622-2023-2-09>
- Ministry of Education and Science of Ukraine. (n.d.). Typovi osvitni prohramy dlia 1–4 klasiv [Typical educational programmes for grades 1–4]. <https://mon.gov.ua/osvita-2/zagalna-serednya-osvita/osvitni-programy/tipovi-osvitni-programy-dlya-1-4-klasiv>
- Shkurenko, O. V., & Smyk, A. O. (2024). Metodychni osoblyvosti vyvchennia temy "Alhorytmy i vykonavtsi" u pochatkovii shkoli [Methodological features of studying algorithms and executors in primary school]. *Molodyi vchenyi*, 7(131), 8–11. <https://doi.org/10.32839/2304-5809/2024-7-131-2>
- Yatsenko, O. I., & Chumak, L. F. (2020). Kryterii doboru seredovyshcha navchannia prohramuvannia [Criteria for selecting a programming learning environment]. *Informatsiini tekhnologii i zasoby navchannia*, 78(4), 225–239. <https://doi.org/10.33407/itlt.v78i4.3469>
- Chen, P., Yang, D., Metwally, A. H. S., Lavonen, J., & Wang, X. (2023). Fostering computational thinking through unplugged activities: A systematic literature review and meta-analysis. *International Journal of STEM Education*, 10, 47. <https://doi.org/10.1186/s40594-023-00434-7>
- Redecker, C. (2017). *European framework for the digital competence of educators: DigCompEdu*. Publications Office of the European Union. <https://doi.org/10.2760/159770>
- Rodrigues, R. N., Andrade, W. L., Sampaio, L. M. R., & Campos, L. M. R. S. (2024). Integration of computational thinking in initial teacher education: A systematic review. *Frontiers in Education*, 9, 1330065. <https://doi.org/10.3389/feduc.2024.1330065>
- Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education*, 14(1), Article 5. <https://doi.org/10.1145/2576872>
- Yadav, A., Stephenson, C., & Hong, H. (2017). Computational thinking for teacher education. *Communications of the ACM*, 60(4), 55–62. <https://doi.org/10.1145/2994591>
- Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142–158. <https://doi.org/10.1016/j.edurev.2017.09.003>
- Fagerlund, J., Häkkinen, P., Vesisenaho, M., & Viiri, J. (2021). Computational thinking in programming with Scratch in primary schools: A systematic review. *Computer Applications in Engineering Education*, 29(1), 12–28. <https://doi.org/10.1002/cae.22255>
- Kite, V., Park, S., & Wiebe, E. (2021). The code-centric nature of computational thinking education: A review of trends and issues in computational thinking education research. *SAGE Open*, 11(2). <https://doi.org/10.1177/21582440211016418>
- Vasylykiv, I. B., & Romanchuk, R. M. (2021). Osoblyvosti formuvannia alhorytmichnoho myslennia molodshykh shkoliariv na urokakh informatyky [Features of developing primary school pupils' algorithmic thinking in informatics lessons]. *Molod i rynok*, 2(188), 90–95. <https://doi.org/10.24919/2308-4634.2021.230493>