

ПЕДАГОГІЧНІ УМОВИ ОРГАНІЗАЦІЇ ДИСТАНЦІЙНИХ КОНКУРСІВ З ПРОГРАМУВАННЯ В LMS MOODLE НА ОСНОВІ ІНТЕГРАЦІЇ З ПЛАТФОРМОЮ DOMJUDGE

Юрій Сіциліцин

Мелітопольський державний педагогічний університет імені Богдана Хмельницького

Анотація:

У статті обґрунтовано педагогічні умови організації дистанційних конкурсів з програмування в Moodle із використанням Domjudge для автоматизованої перевірки рішень. Показано, що інтеграція платформ забезпечує об'єктивність, прозорість і швидкість оцінювання та формує цілісне цифрове середовище, у якому студенти працюють у передбачуваних умовах і отримують миттєвий зворотний зв'язок. Розкрито значення розвитку алгоритмічного мислення, саморегуляції та академічної доброчесності в умовах змагального формату. Сформульовано ключові педагогічні умови ефективної організації конкурсів: методичний супровід викладача, чітко структуровані правила змагання, стабільна та передбачувана технічна інфраструктура, а також інтеграція контекстів у навчальну логіку дисципліни. Визначені умови становлять основу для побудови педагогічно обґрунтованої моделі дистанційних змагань з програмування та сприяють підвищенню якості професійної підготовки майбутніх фахівців ІТ-галузі.

Ключові слова:

дистанційні конкурси з програмування; алгоритмічне мислення; автоматизоване оцінювання; Moodle; Domjudge; цифрове освітнє середовище.

Resume:

Sitsylitsyn Yurii. Pedagogical conditions for organizing distance programming contests in LMS Moodle based on integration with the Domjudge platform.

The article substantiates the pedagogical conditions for organizing distance programming contests in Moodle, integrated with the Domjudge platform for automated code evaluation. The study argues that the combination of these systems ensures objectivity, transparency, and rapid assessment, forming a coherent digital learning environment in which students work under predictable conditions and receive immediate feedback. The analysis emphasizes the significance of algorithmic thinking, self-regulation, and academic integrity in learning environments that incorporate competition and challenge. Particular attention is given to defining the pedagogical conditions that make such contests educationally effective: comprehensive methodological support from the instructor, clearly structured contest rules, a stable and predictable technical infrastructure, and the integration of competitive activities into the instructional logic of the academic discipline. These conditions collectively create a pedagogically grounded model for distance programming contests, supporting meaningful learning, strengthening students' intellectual autonomy, and aligning competitive programming practices with the goals of professional training in computer science.

Key words:

distance programming contests; algorithmic thinking; automated assessment; Moodle; Domjudge; digital learning environment.

Постановка проблеми. Сучасна система підготовки фахівців у галузі комп'ютерних наук перебуває в умовах стрімкої цифровізації, що зумовлює необхідність адаптації освітнього процесу до нових організаційних і технологічних реалій. Перехід значної частини навчальних активностей у дистанційний формат змінив вимоги до форм контролю, структури взаємодії та перевірки програмних рішень студентів. За цих умов виникає потреба у створенні педагогічно обґрунтованих моделей, здатних забезпечити поєднання практикоорієнтованого навчального змісту, об'єктивності оцінювання, прозорості алгоритмічної діяльності та дотримання принципів академічної доброчесності.

Цифрові педагогічні підходи, що використовують платформи змагального програмування, демонструють значний позитивний вплив як на розвиток алгоритмічного мислення, так і на підсилення навчальної мотивації студентів університетів, що навчаються за спеціальностями комп'ютерних наук.

Аналіз останніх досліджень і публікацій. Дослідження розвитку алгоритмічного мислення показують, що змагальні платформи створюють практичне середовище для розв'язування задач. (L. Gonzalez-Guerra та ін., 2024) встановили, що

конкурси з програмування забезпечують студентам можливість застосовувати складні алгоритмічні методи, орієнтуючись на необхідність створення найбільш ефективних розв'язань. Ian Nery Bandeira та співавтори (Ian Nery Bandeira та ін., 2019) довели, що участь у таких змаганнях поглиблює розуміння логіки програмування й формує навички, які залишаються корисними навіть за межами суто програмістської діяльності. У дослідженні (Leticia Laura-Ochoa та ін., 2022) зазначено, що використання інструментів програмування у поєднанні з практиками обчислювального мислення покращує результати навчання, зокрема здатність до абстрагування та побудови алгоритмів.

Питання навчальної мотивації розглядаються у працях, де позитивний ефект від змагального форматування є послідовно підтвердженим. (L. Gonzalez-Guerra та ін., 2024) вказують на суттєве зростання мотивації та залученості студентів, оскільки гейміфікований підхід стимулює прагнення до самовдосконалення. Julián Moreno та колеги (Moreno Cadavid, J., & Pineda Corcho, A., 2018) підкреслюють позитивне ставлення студентів до поєднання змагань і елементів гейміфікації, що призводить до покращення академічних результатів.

R. Lawtence та ін. (Lawtence, R., 2004) відзначають, що змагальні формати є дієвим чинником підвищення результативності діяльності студентів через дружнє суперництво та активну взаємодію.

Проблема педагогічної ефективності платформ змагального програмування також розглядається в низці досліджень. Verdú, E. (E. Verdú та ін., 2012) довели, що інтегровані онлайн-платформи з автоматичним оцінюванням сприяють покращенню навчальних результатів і спрощують опанування матеріалу, незалежно від рівня підготовленості чи цифрових навичок студентів. J. García та ін. (García J. та ін., 2014) на основі аналізу кривих навчання підтвердили педагогічну успішність конкурсних форматів, хоча й зазначили потребу у поглиблених дослідженнях впливу таких моделей на структуру й динаміку навчального процесу.

Узагальнені результати вказують, що платформи змагального програмування формують залучене, практикоорієнтоване навчальне середовище, яке одночасно сприяє розвитку алгоритмічного мислення та зміцненню навчальної мотивації завдяки гейміфікованим і проблемно-орієнтованим підходам.

Формулювання цілей статті. Попри численні позитивні ефекти, дослідження підкреслюють брак системних педагогічних моделей, які б одночасно враховували особливості LMS та платформ автоматичного оцінювання. Відсутність теоретично обґрунтованої моделі інтеграції Moodle і Domjudge створює помітну прогалину, яка ускладнює широке впровадження дистанційних змагань як педагогічно цілісної форми діяльності.

У такому контексті метою статті є визначення та обґрунтування педагогічних умов ефективної організації дистанційних конкурсів з програмування у структурі Moodle з використанням Domjudge, що дозволить сформувати цілісну модель цифрового освітнього середовища та підвищити якість професійної підготовки студентів у галузі комп'ютерних наук.

Виклад основного матеріалу дослідження. Організація дистанційних конкурсів з програмування вимагає не лише наявності окремих технічних інструментів, але й побудови цілісного цифрового освітнього середовища, у межах якого навчальна, організаційна та контрольно-оцінювальна діяльність студентів утворюють логічно узгоджену систему. Єдність такого середовища має вирішальне значення, оскільки саме воно забезпечує педагогічну керованість процесу, стабільність навчального досвіду для студентів і безперервність усіх етапів змагання – від ознайомлення з інструкціями до аналізу отриманих результатів.

Використання Moodle як базової інфраструктури дозволяє структурувати освітній процес у спосіб, який відповідає сучасним вимогам цифрової педагогіки. Moodle виконує роль організаційного «ядра», що забезпечує доступ до навчальних матеріалів, підтримує систему ролей і прав, інтегрує модулі оцінювання, дозволяє викладачеві формувати інструктивні ресурси, надавати підтримку, здійснювати комунікацію та зберігати всі елементи навчальної діяльності в єдиній системі. У цьому середовищі студент працює з навчальною інформацією послідовно та без необхідності переходити між різними сайтами або платформами, що значною мірою знижує когнітивне навантаження та підвищує продуктивність у роботі із завданнями.

Педагогічна важливість інтеграції Moodle і Domjudge полягає в тому, що студент сприймає конкурс як органічну складову курсу, а не як зовнішню або технічно складну процедуру. Уже на етапі підготовки до конкурсу він отримує весь необхідний супровід: календарні позначки, інструкції до роботи з платформою, пояснення структури змагання, рекомендації щодо розв'язання задач і особливості оцінювання. Усе це створює атмосферу передбачуваності, яка є ключовою для зниження тривожності та формування позитивного навчального досвіду. Натомість фрагментована інфраструктура, у якій кожен етап конкурсу відбувається в окремому середовищі, призводить до втрати педагогічної керованості та знижує ефективність навчання.

Автоматизація оцінювання у дистанційних конкурсах з програмування стає ключовим педагогічним чинником, що забезпечує об'єктивність, прозорість і відтворюваність результатів. У традиційних формах перевірки програмних робіт викладач часто опиняється перед складністю оцінювання не лише кінцевого результату, але й процесу розв'язання, особливостей алгоритмічного підходу, коректності обробки граничних ситуацій і відповідності рішення вимогам задачі. Людський фактор неминує впливає на сприйняття якості роботи: інтерпретація помилок відрізняється залежно від досвіду викладача, обсягу перевірюваних матеріалів, рівня навантаження та суб'єктивних очікувань щодо студентських результатів. У дистанційному форматі цей фактор посилюється, оскільки обсяг робіт збільшується, а механізми контролю зменшуються. Тому автоматизовані системи на зразок Domjudge виконують важливу педагогічну функцію – вони мінімізують суб'єктивність та забезпечують однакові умови для всіх учасників.

Domjudge здійснює перевірку програмних рішень шляхом компіляції, запуску коду у

захищеному середовищі, виконання тестів, вимірювання часу, пам'яті та аналізу результатів за чіткими критеріями, які є спільними для всіх студентів. Такий підхід формує стандарт оцінювання, у якому виключено варіативність інтерпретацій. Студент отримує результат, який ґрунтується не на порівнянні з очікуваннями викладача, а на жорсткій відповідності технічним умовам задачі. Це забезпечує педагогічну справедливість, що є ключовою цінністю навчального процесу.

Система автоматизованого оцінювання створює умови для формування прозорої навчальної траєкторії. Студенти бачать свої результати у вигляді детальних логів виконання, повідомлень про помилки, невдалих тестів, перевищення ліміту часу або пам'яті. Така деталізація формує ефективний механізм рефлексії, бо дозволяє студентам аналізувати власні помилки, порівнювати підходи до розв'язання задач, удосконалювати алгоритми та повторно перевіряти рішення. Це важливий елемент формування оцінювання, який підтримує поступовий розвиток компетентностей. Саме в цьому полягає педагогічна сутність автоматизованої системи: вона не лише виставляє підсумкову оцінку, а й створює простір для самонавчання, де кожна помилка стає матеріалом для подальшого вдосконалення.

У контексті дистанційних конкурсів автоматизація також забезпечує часову рівність між учасниками. Усі студенти виконують задачі в однакових умовах, і їхні рішення перевіряються за ідентичними критеріями. Це повністю усуває можливість нерівності між групами або учасниками з різних локацій. Стабільність і повторюваність перевірки формують педагогічну передбачуваність, яка, у свою чергу, знижує тривожність студентів та підвищує їхню довіру до системи оцінювання (Perretta J. та інші, 2019).

Автоматизація дозволяє безперервно відстежувати середовище виконання, що виключає можливість випадкового втручання, несанкціонованої допомоги або використання сторонніх ресурсів. У педагогічному вимірі це відповідає принципам академічної доброчесності, оскільки результат, отриманий системою, є незалежним від будь-яких сторонніх факторів. Прозорий алгоритм перевірки унеможливує ситуацію, коли студент намагається вплинути на оцінку не через якість роботи, а через зовнішні обставини. Саме Domjudge, як система з використанням ізольованого середовища, сприяє формуванню чесної поведінки студентів, оскільки виключає можливість маніпуляції результатами. Схемі роботи judgehost, показує послідовність перевірки рішень.

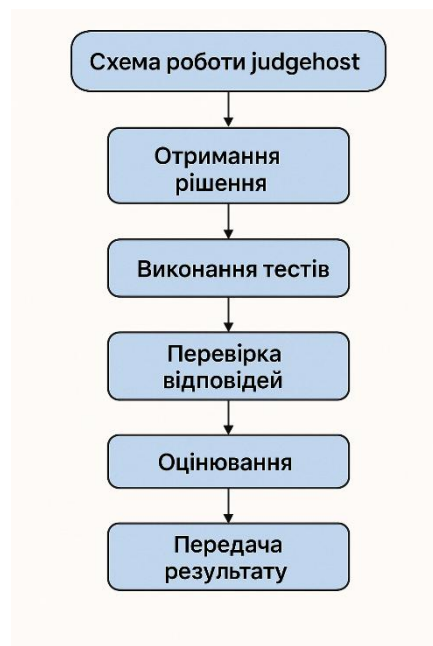


Рис. 1. Послідовність перевірки рішень

Важливо підкреслити, що автоматизована система не замінює педагогічного впливу викладача, а навпаки – дозволяє йому зосередитися на аналізі стратегій розв'язання, консультуванні студентів, формуванні алгоритмічного мислення та розвитку здатності до самостійного пошуку ефективних рішень. Викладач отримує доступ до всіх даних про діяльність студентів, що дозволяє більш глибоко аналізувати індивідуальні труднощі, будувати персоналізовані траєкторії навчання та надавати адресний зворотний зв'язок. Таким чином автоматизація стає не просто технічним інструментом, а елементом педагогічного дизайну, який посилює навчальний ефект.

Узагальнимо, що автоматизація оцінювання забезпечує комплексну педагогічну об'єктивність: вона усуває суб'єктивність людського чинника, гарантує рівність умов, підсилює навчальну мотивацію, створює умови для глибокої рефлексії й самоаналізу, забезпечує дотримання академічної доброчесності та робить конкурси з програмування органічною частиною сучасного цифрового освітнього простору.

Організація дистанційних конкурсів з програмування в інтегрованому середовищі Moodle-Domjudge створює умови, у яких розвиток алгоритмічного мислення відбувається не лише як опанування технік програмування, а як формування здатності застосовувати структуроване логічне міркування для розв'язання нових, невідомих задач. Алгоритмічне мислення постає не як окрема навичка, обмежена навчальними прикладами, а як спосіб інтелектуальної діяльності, що передбачає аналіз проблеми, декомпозицію,

проектування оптимальних стратегій, перевірку гіпотез та поступову побудову ефективного рішення. Дистанційний формат, особливо за участю автоматизованої системи контролю, посилює цей ефект, оскільки студенти працюють у ситуації, де результат їхніх рішень визначається виключно коректністю алгоритму та його відповідністю технічним вимогам.

У такому середовищі студент засвоює, що будь-яка програма – це відображення логічної структури задачі, а успішність її виконання залежить не від інтуїтивних здогадок, а від чіткості побудови алгоритму. Domjudge,

перевіряючи рішення в різних сценаріях, примушує студента мислити послідовно і системно. На практиці це означає, що студент повинен передбачити крайні випадки, проаналізувати поведінку алгоритму на великих обсягах даних, оцінити часову складність та оптимізувати використання ресурсів. Порівняння типових помилок студентів під час конкурсу з показниками успішності можна побачити у таблиці, яка відображатиме зв'язок між некоректним алгоритмічним підходом та кількістю отриманих помилок.

Таблиця 1

Зв'язок між типами алгоритмічних помилок і кількістю отриманих системних відмов

№	Типова помилка студента	Причина виникнення	Найчастіша системна відповідь	Кількість випадків
1	Неправильна обробка граничних випадків	Відсутність тестування на мінімальних/максимальних значеннях	Wrong Answer	42
2	Неефективний алгоритм із надмірною складністю	Використання $O(n^2)$, де потрібен $O(n \log n)$	Time Limit Exceeded	37
3	Неправильна робота з пам'яттю	Вихід за межі масивів, некоректні індекси	Runtime Error	25
4	Неточність у роботі з числами з рухомою комою	Некоректні обчислення, помилки округлення	Wrong Answer	18
5	Некоректний формат виводу	Пропуски, зайві пробіли, неправильний перенос рядків	Presentation Error	21
6	Використання невірної підходу до задачі	Хибна логіка алгоритму, неправильна інтерпретація умови	Wrong Answer	30

Важливо, що така діяльність водночас формує навички саморегуляції. У дистанційному форматі викладач не може оперативно контролювати кожен крок студента, тому останній змушений самостійно планувати час, розподіляти зусилля, визначати послідовність виконання задач, аналізувати власні помилки та обирати оптимальну траєкторію роботи. Domjudge надає негайний зворотний зв'язок, який виконує роль зовнішнього регулятора, але саме студент здійснює контроль над власними діями, приймає рішення про внесення змін до алгоритму, перевіряє проміжні результати й визначає, наскільки коректно розвивається його стратегія розв'язання. Цей якісний перехід від зовнішньої регуляції через викладача до внутрішньої саморегуляції формує одну з ключових компетентностей майбутнього програміста – здатність управляти власним процесом мислення.

Студенти, які регулярно працюють із системами автоматичного оцінювання, демонструють більш високу здатність до аналізу власних помилок та виправлення недоліків,

оскільки кожне рішення, відхилене системою, стає об'єктом рефлексії. Вони навчаються визначати причину відмови – логічну помилку, некоректну обробку крайніх випадків, неефективний алгоритм або перевищення лімітів ресурсів. Розвиток алгоритмічного мислення в такій моделі невіддільний від розвитку здатності до предиктивного аналізу – студент починає прогнозувати можливі помилки ще до відправлення рішення. Це свідчить про формування внутрішньої моделі контрольних дій, що є суттєвим компонентом саморегуляції. Студент набуває вміння передбачати наслідки застосованого алгоритму, оцінювати його потенційні обмеження та мислити «на кілька кроків уперед». Саме ця властивість інтелектуального передбачення є характерною для досвідчених програмістів і формується значно швидше завдяки роботі з автоматизованими системами тестування. Такі дані мають педагогічну цінність для аналізу ефективності навчання як для викладача, так і для самого студента.

Розвиток саморегуляції також посилюється завдяки психологічним властивостям конкурсного середовища. Змагання створюють атмосферу відповідальності та автономності: студент усвідомлює, що результат залежить лише від нього, оскільки система оцінювання не допускає випадкових коливань і не враховує факторів, що не пов'язані з якістю роботи. Відчуття контролю над ситуацією сприяє підвищенню внутрішньої мотивації та цілеспрямованості. Йдеться про розвиток здатності не лише розв'язувати задачі, а й керувати власною поведінкою у процесі їх виконання.

Отже, взаємодія із системою Domjudge у межах єдиного цифрового середовища Moodle створює педагогічні умови, у яких формування алгоритмічного мислення та розвиток саморегуляції взаємно підсилюють одне одного. Студент не лише опановує технічні аспекти програмування, а й набуває інтелектуальної гнучкості, здатності до аналізу, передбачення, планування та контролю власних дій – компетентностей, які лежать в основі професійного становлення майбутнього програміста.

Ефективність дистанційних конкурсів з програмування значною мірою залежить від того, наскільки методично організованим є весь процес навчання. Навіть за наявності потужної автоматизованої системи оцінювання Domjudge та структурованого середовища Moodle роль викладача не зменшується; навпаки, педагогічне значення його діяльності зростає, оскільки саме він виступає координатором, фасилітатором і наставником, який формує навчальний контекст

та створює умови для осмисленої діяльності студентів. Автоматизація не замінює викладача, а змінює характер його участі: від прямого контролю – до надання методичного керівництва, пояснення стратегій, підтримки саморегуляції та спрямування студентського мислення в продуктивне русло.

Методичний супровід починається задовго до проведення конкурсу. Викладач формує інформаційне середовище, у якому студенти отримують зрозумілі інструкції, опис правил, приклади розв'язання типових задач, пояснення формату тестування, опис помилок, що найчастіше виникають під час подання рішень, та рекомендації щодо організації власного часу. Важливо, щоб методичні матеріали були не лише зібрані в одному місці, але й структуровані відповідно до етапів виконання змагання. Таке впорядкування створює загальну педагогічну рамку, в межах якої студент орієнтується у своїх діях і не відчуває інформаційного перевантаження. У процесі підготовки важливо також надати студентам можливість потренуватися в умовах, максимально наближених до реальних. Для цього створюються демонстраційні контекти, що дозволяють ознайомитися з інтерфейсом Domjudge, типами повідомлень системи, механізмами перевірки рішень та логікою подання відповідей. Такий підхід знижує тривожність, формує відчуття передбачуваності та сприяє підвищенню впевненості студентів у власних силах. Результати участі у тренувальних змаганнях зведені у таблицю для аналізу типових помилок і динаміки підготовки.

Таблиця 2

Результати участі студентів у тренувальних змаганнях Domjudge

№	Студент	Кількість розв'язаних задач	Кількість спроб	Найчастіша помилка	Час першого успішного розв'язання	Динаміка прогресу (тренд)
1	Студент А	5 із 7	14	Wrong Answer	00:27:15	Покращення
2	Студент Б	3 із 7	19	Time Limit Exceeded	00:41:03	Стабільність
3	Студент В	2 із 7	11	Presentation Error	01:05:44	Повільне зростання
4	Студент Г	6 із 7	9	Wrong Answer	00:18:22	Високий прогрес
5	Студент Д	1 із 7	8	Runtime Error	01:22:10	Низька активність
6	Студент Е	4 із 7	12	Wrong Answer	00:33:19	Покращення
7	Студент Ж	0 із 7	3	Compilation Error	—	Нульовий прогрес
8	Студент З	5 із 7	16	Time Limit Exceeded	00:29:50	Стабільне покращення

У день проведення конкурсу педагогічна роль викладача полягає у забезпеченні стабільного

інформаційного супроводу. Він повинен бути доступним для консультацій, оперативного

роз'яснення технічних нюансів або уточнення умов задач, але при цьому не втручатися у зміст програмування. Викладач підтримує психологічну рівновагу студентів, адже змагальний формат може створювати емоційне напруження, яке потрібно пом'якшувати завдяки чітким інструкціям, доброзичливому тону та впевненій комунікації. Після завершення конкурсу викладач переходить до етапу педагогічного аналізу. Його завданням є не лише перевірити підсумкові бали, але й здійснити огляд основних тенденцій, з якими працювали студенти: рівня алгоритмічної складності, типів помилок, стратегії вибору задач, часу, витраченого на кожну спробу. Domjudge надає деталізовані журнали виконання, які можна інтерпретувати у навчальному контексті, формуючи у студентів розуміння власних сильних і слабких сторін.

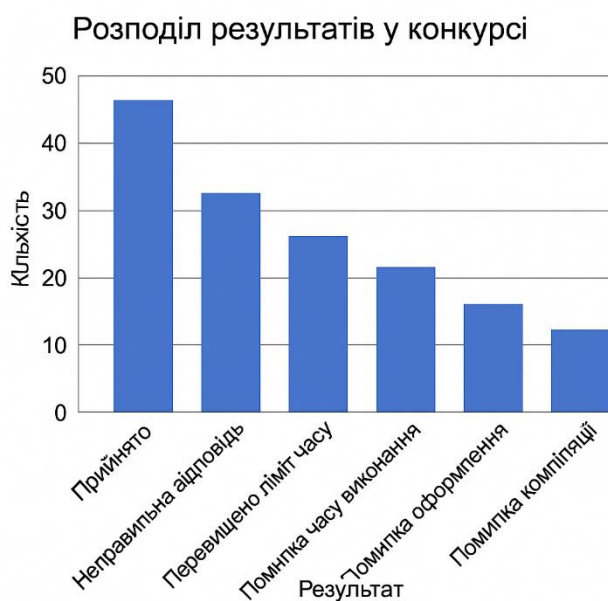


Рис. 2. Розподіл результатів у конкурсі

Педагогічний сенс цього етапу полягає у підтримці рефлексії. Викладач не просто повідомляє студентам їхні результати, а пояснює, чому певні рішення були оптимальними, які алгоритми продемонстрували кращу ефективність, які помилки були типовими і як їх можна було уникнути. Це перетворює конкурс на навчальний інструмент, а не лише на засіб оцінювання. У цьому сенсі методичний супровід є одним із ключових чинників формувального оцінювання, що дозволяє студенту побачити особистий поступ і спроектувати подальші кроки.

Таким чином роль викладача у дистанційних конкурсах з програмування полягає у створенні цілісного методичного середовища, яке підтримує студента на всіх етапах роботи – від підготовки до рефлексії. Викладач формує умови, у яких автоматизована система перевірки працює не ізольовано, а як частина педагогічної стратегії,

спрямованої на розвиток алгоритмічного мислення, саморегуляції та професійної відповідальності студентів. Саме завдяки цьому конкурси з програмування набувають значення комплексного навчального інструмента, що поєднує інтелектуальний виклик, технологічну точність і педагогічну підтримку.

Педагогічна цінність дистанційних конкурсів з програмування значною мірою залежить від технічної інфраструктури, на якій вони функціонують. Незалежно від якості методичного супроводу, продуманості завдань чи рівня підготовки студентів, навчальна діяльність не буде ефективною, якщо вона розміщується у середовищі, що не гарантує стабільність, швидкодію та відсутність технічних збоїв. Для студента технічна непередбачуваність завжди перетворюється на бар'єр, який руйнує логіку навчального процесу, підвищує тривожність, знижує мотивацію та спотворює справедливність оцінювання. Тому технічна інфраструктура є не лише технологічним елементом, але й суттєвим педагогічним чинником.

Інтегроване середовище Moodle-Domjudge потребує чіткої інфраструктурної архітектури, що включає серверну частину, judgehost-и, системи резервування, канали мережевої взаємодії та захищені механізми зберігання даних. У такій архітектурі важлива не лише наявність серверів, а й їхня конфігурація, узгодженість і поведінка під навантаженням. Педагогічна передбачуваність виникає тоді, коли система реагує стабільно незалежно від кількості учасників, обсягу поданих рішень або тривалості конкурсу. Domjudge працює з judgehost-ами – окремими серверами або контейнерами, що відповідають за компіляцію та виконання рішень. Їхня кількість та продуктивність визначають швидкість перевірки, а отже, впливають і на навчальний ефект. Якщо студент тривалий час чекає результату виконання рішення, він втрачає фокус, перериває алгоритмічний процес мислення або навіть отримує упереджене ставлення до системи. Натомість швидке отримання результатів стимулює саморегуляцію, підтримує інтелектуальний темп роботи та сприяє формуванню циклів «спроба – аналіз – корекція – повторна спроба». Порівняння часу обробки рішень за різних конфігурацій judgehost-ів можна представити у таблиці. Резервування системи – це ще один важливий аспект педагогічної передбачуваності. Використання резервних judgehost-ів, дзеркальних серверів або snapshot-технологій Proxmox забезпечує стійкість платформи до збоїв. Педагогічне значення резервування полягає у гарантії того, що навчальна діяльність студентів не буде зруйнована непередбачуваними технічними

проблемами. Викладач може бути впевненим, що конкурс завершиться коректно, а студенти – що їхні роботи не буде втрачено. Окремого значення набуває і безпека даних. У процесі конкурсу система обробляє не лише коди програм, але й персональні дані, облікові записи, журнали подій, приватні повідомлення та технічні логи. Захист цих даних є етичним та педагогічним обов'язком, оскільки порушення конфіденційності не лише підриває довіру студентів, але й впливає на їхню здатність конструктивно взаємодіяти з цифровим середовищем. Коли технічна інфраструктура працює узгоджено, студенти не помічають її наявності – вона стає прозорою, а отже педагогічно ефективною. Технічні системи повинні не привертати до себе увагу, а підтримувати структуру навчальної діяльності, роблячи її стабільною, передбачуваною та безперервною. Будь-яка технічна помилка руйнує навчальний контекст, у той час як стабільна система підтримує плавність навчального мислення та дозволяє студенту повністю зосередитися на розв'язанні задач.

Таким чином технічна інфраструктура є фундаментом педагогічної передбачуваності. Вона забезпечує стабільність роботи середовища, рівність умов, безпеку даних, швидкість зворотного зв'язку, безперервність навчальної діяльності та довіру студентів до системи оцінювання. Тільки за наявності такої інфраструктури дистанційні конкурси з програмування можуть виконувати не лише вимірювальну, але й глибоку навчальну функцію, сприяючи розвитку компетентностей і формуванню професійної культури майбутніх програмістів.

Висновки. Організація дистанційних конкурсів з програмування у середовищі Moodle з використанням платформи Domjudge демонструє високу педагогічну ефективність за умови дотримання цілісної моделі, що поєднує технічну інфраструктуру, методичний супровід та дидактичні цілі навчального процесу.

Проведене дослідження дало змогу окреслити комплекс педагогічних умов, необхідних для ефективної організації дистанційних конкурсів з програмування в інтегрованому середовищі

Moodle-Domjudge. Насамперед визначено важливість створення єдиного цифрового освітнього середовища, у якому навчальні матеріали, інструкції, механізми подання рішень і система оцінювання утворюють логічно узгоджену структуру, що забезпечує стабільність і передбачуваність для студентів. Важливою умовою є забезпечення об'єктивності та прозорості оцінювання через автоматизовану перевірку програмних рішень, яка усуває суб'єктивність людського чинника й формує стандартизовані критерії успішності.

Окрему педагогічну цінність становить розвиток алгоритмічного мислення, що виникає в процесі побудови оптимальних рішень, аналізу задач і взаємодії з тестовими наборами. Цей процес нерозривно пов'язаний із формуванням саморегуляції: студенти навчаються планувати власну діяльність, контролювати час, виправляти помилки й аналізувати прогрес, що сприяє становленню відповідального, автономного способу навчання. Значущою умовою виступає методичний супровід викладача, який організовує навчальний контекст, забезпечує якість інструкцій, підтримує комунікацію під час змагання та допомагає інтерпретувати результати після його завершення.

Педагогічний ефект конкурсу неможливий без підтримки академічної доброчесності, яка забезпечується ізольованим середовищем виконання, фіксацією дій студента, механізмами автентифікації, контролем подібності коду та прозорими правилами змагання. Завершальною умовою є надійність технічної інфраструктури, адже стабільна робота серверів, judgehost-ів, мережових каналів і систем резервування створює рівні умови, забезпечує безперервність навчальної діяльності та формує відчуття безпеки, що є фундаментом педагогічної передбачуваності.

Усі ці умови в сукупності формують цілісну модель організації дистанційних конкурсів з програмування, у якій педагогічна логіка гармонійно поєднується з технічною досконалістю та потребами професійної підготовки майбутніх програмістів.

Список використаних джерел

- Bacelo, A., & Gómez-Chacón, I. M. (2023). Characterising algorithmic thinking: A university study of unplugged activities. *Thinking Skills and Creativity*, 48. <https://doi.org/10.1016/j.tsc.2023.101284>
- Bandeira, I. N., Machado, T. V., Dullens, V. F., & Canedo, E. D. (2019). Competitive programming: A teaching methodology analysis applied to first-year programming classes. In 2019 IEEE Frontiers in Education Conference (FIE) (pp. 1–8). IEEE. <https://doi.org/10.1109/FIE43999.2019.9028518>

References

- Bacelo, A., & Gómez-Chacón, I. M. (2023). Characterising algorithmic thinking: A university study of unplugged activities. *Thinking Skills and Creativity*, 48. <https://doi.org/10.1016/j.tsc.2023.101284> [in English]
- Bandeira, I. N., Machado, T. V., Dullens, V. F., & Canedo, E. D. (2019). Competitive programming: A teaching methodology analysis applied to first-year programming classes. In 2019 IEEE Frontiers in Education Conference (FIE) (pp. 1–8). IEEE. <https://doi.org/10.1109/FIE43999.2019.9028518> [in English]

- García, J. R., García-Sánchez, P., & Castro, M. (2014). The learning curves of competitive programming. *AIP Advances*, 1 (1), 934. <https://doi.org/10.1063/1.903475>
- Gonzalez-Guerra, L. H., Juárez, G. H., Pérez-Lezama, C. V., González, E. G. Q., Campos-García, G. A., & Cacho, M. L. B. (2024). Enhancing problem-solving skills: The synergy of competitive programming and gamification strategy. In 2024 IEEE Global Engineering Education Conference (EDUCON) (pp. 1–7). IEEE. <https://doi.org/10.1109/EDUCON60312.2024.10578845>
- Juškevičienė, A. (2020). Developing algorithmic thinking through computational making. In G. Dzemyda, J. Bernatavičienė, & J. Kacprzyk (Eds.), *Data Science: New Issues, Challenges and Applications* (Studies in Computational Intelligence, Vol. 869). Springer. https://doi.org/10.1007/978-3-030-39250-5_10
- Laura-Ochoa, L., & Bedregal-Alpaca, N. (2022). Incorporation of computational thinking practices to enhance learning in a programming course. *International Journal of Advanced Computer Science and Applications*, 13 (2). <https://doi.org/10.14569/IJACSA.2022.0130204>
- Lawrence, R. (2004). Teaching data structures using competitive games. *IEEE Transactions on Education*, 47 (4), 459–466. <https://doi.org/10.1109/TE.2004.825053>
- Moreno Cadavid, J., & Pineda Corcho, A. F. (2018). Competitive programming and gamification as strategy to engage students in computer science courses. *Revista Espacios*, 39 (35), 11.
- Perretta, J., Weimer, W., & DeOrio, A. (2019). Human vs. automated coding style grading in computing education. *ASEE Annual Conference & Exposition*. <https://doi.org/10.18260/1-2--32906>
- Udvaros, J., & Takáč, O. (2020). Developing computational thinking by microcontrollers. In *ICERI2020 Proceedings* (pp. 6877–6882). IATED. <https://doi.org/10.21125/iceri.2020.1600>
- Verdú, E., Regueras, L. M., Verdú, M. J., Leal, J. P., de Castro, J. P., & Queirós, R. (2012). A distributed system for learning programming online. *Computers & Education*, 58 (1), 66–84. <https://doi.org/10.1016/j.compedu.2011.08.015>

- García, J. R., García-Sánchez, P., & Castro, M. (2014). The learning curves of competitive programming. *AIP Advances*, 1 (1), 934. <https://doi.org/10.1063/1.903475> [in English]
- Gonzalez-Guerra, L. H., Juárez, G. H., Pérez-Lezama, C. V., González, E. G. Q., Campos-García, G. A., & Cacho, M. L. B. (2024). Enhancing problem-solving skills: The synergy of competitive programming and gamification strategy. In 2024 IEEE Global Engineering Education Conference (EDUCON) (pp. 1–7). IEEE. <https://doi.org/10.1109/EDUCON60312.2024.10578845> [in English]
- Juškevičienė, A. (2020). Developing algorithmic thinking through computational making. In G. Dzemyda, J. Bernatavičienė, & J. Kacprzyk (Eds.), *Data Science: New Issues, Challenges and Applications* (Studies in Computational Intelligence, Vol. 869). Springer. https://doi.org/10.1007/978-3-030-39250-5_10 [in English]
- Laura-Ochoa, L., & Bedregal-Alpaca, N. (2022). Incorporation of computational thinking practices to enhance learning in a programming course. *International Journal of Advanced Computer Science and Applications*, 13 (2). <https://doi.org/10.14569/IJACSA.2022.0130204> [in English]
- Lawrence, R. (2004). Teaching data structures using competitive games. *IEEE Transactions on Education*, 47 (4), 459–466. <https://doi.org/10.1109/TE.2004.825053> [in English]
- Moreno Cadavid, J., & Pineda Corcho, A. F. (2018). Competitive programming and gamification as strategy to engage students in computer science courses. *Revista Espacios*, 39 (35), 11. [in English]
- Perretta, J., Weimer, W., & DeOrio, A. (2019). Human vs. automated coding style grading in computing education. *ASEE Annual Conference & Exposition*. <https://doi.org/10.18260/1-2--32906> [in English]
- Udvaros, J., & Takáč, O. (2020). Developing computational thinking by microcontrollers. In *ICERI2020 Proceedings* (pp. 6877–6882). IATED. <https://doi.org/10.21125/iceri.2020.1600> [in English]
- Verdú, E., Regueras, L. M., Verdú, M. J., Leal, J. P., de Castro, J. P., & Queirós, R. (2012). A distributed system for learning programming online. *Computers & Education*, 58 (1), 66–84. <https://doi.org/10.1016/j.compedu.2011.08.015> [in English]

Відомості про автора:

Сіцилицин Юрій Олександрович

Sicylicyn_Yuriy@mspu.edu.ua

Мелітопольський державний педагогічний університет імені Богдана Хмельницького
Наукове Містечко, вулиця, 59, Запоріжжя,
Запорізька обл., 69000, Україна

doi: 10.33842/22195203-2025-35-136-219-226

Матеріал надійшов до редакції 16. 11. 2025 р.

Прийнято до друку 19. 11. 2025 р.

Information about the author:

Sitsylitsyn Yuriy Oleksandrovych

Sicylicyn_Yuriy@mspu.edu.ua

Bogdan Khmelnytsky Melitopol State
Pedagogical University
Scientific Town, Street 59, Zaporizhzhia,
Zaporizhzhia region, 69000, Ukraine

doi: 10.33842/22195203-2025-35-136-219-226

Received at the editorial office 16. 11. 2025.

Accepted for publishing 19. 11. 2025.