

УДК 004.8

Левченко А.С., Кліменко В.І.

*Хмельницький національний університет*

## **МЕТОД ФОРМУВАННЯ GIFT-ФАЙЛІВ ДЛЯ СЕРЕДОВИЩА MOODLE ДЛЯ АВТОМАТИЗОВАНОГО ФОРМУВАННЯ ТЕСТОВИХ ЗАВДАНЬ**

*Подано метод формування GIFT-файлів у складі інформаційної технології автоматизованого створення тестових завдань для середовища Moodle. Тестове завдання розглядається як об'єкт із внутрішньою об'єктною моделлю та зовнішнім текстовим представленням у форматі GIFT, між якими побудовано уніфіковане відображення. Запропоновано ієрархію викликів, у якій генерація змісту завдання здійснюється моделями типу BaseModel, а серіалізація до GIFT-файлу – спеціалізованим методом WriteTestCase. Показано інтеграцію методу з понятійно-тезовою моделлю навчального матеріалу, підтримку різних типів тестових завдань і використання випадкового вибору для формування варіативних тестів. Зазначено переваги методу щодо масштабованості, сумісності з Moodle та зручності експлуатації сформованих GIFT-файлів у навчальному процесі.*

*The paper presents a method for generating GIFT files within an information technology for automated test item creation in the Moodle environment. A test item is treated as an object with an internal model and an external text representation in GIFT format, and a unified mapping between these two levels is defined. The proposed call hierarchy separates content generation, implemented by BaseModel-derived item models, from serialization to a GIFT file performed by the WriteTestCase method. The method is integrated with the concept–thesis model of instructional texts, supports multiple item types and employs random selection to produce variable test versions. Its advantages in terms of scalability, compatibility with Moodle and practical usability of the resulting GIFT files in educational practice are outlined.*

Актуальність дослідження зумовлена зростаючою потребою у високоефективних інструментах автоматизованого формування тестових завдань, які можуть підтримувати індивідуальні освітні траєкторії та адаптивні сценарії контролю знань [1, 2]. Традиційні підходи до створення тестів у Moodle вимагають значних часових і кадрових ресурсів [3], особливо коли йдеться про масштабні курси або системи, що повинні динамічно змінювати складність і зміст завдань залежно від рівня підготовки студента [4, 5]. У цьому контексті формати автоматизованої генерації, зокрема GIFT, виступають ключовим інструментом для забезпечення гнучкого та швидкого оновлення банку тестових завдань [6, 7].

Завдяки можливості алгоритмічного формування змісту питання [8], а також програмної варіації параметрів і структур [9], автоматизація дозволяє створювати великі набори тестів зі збереженням педагогічної валідності [10, 11]. Для адаптивного тестування це має особливе значення, оскільки система потребує великої кількості еквівалентних за змістом [12], але різних за формою завдань [13], щоб забезпечити коректне визначення рівня знань студента та мінімізувати ефект запам'ятовування [14, 15, 16]. Використання уніфікованих моделей представлення тестового завдання й автоматизованої серіалізації у формат GIFT забезпечує можливість швидкого масштабування, централізованого оновлення та автоматичного формування різнорівневих тестових наборів.

Таким чином, розробка методу формування GIFT-файлів для автоматизованого створення тестових завдань не лише підвищує ефективність розробки систем тестового контролю, але й створює основу для побудови адаптивних освітніх систем, здатних гнучко реагувати на індивідуальні особливості користувача та вимоги сучасного цифрового навчального середовища.

Метод формування GIFT-файлів для середовища Moodle у розробленій інформаційній технології ґрунтується на поданні тестового завдання як об'єкта, що має чітко визначену внутрішню структуру [17] та однозначне текстове відображення у форматі розмітки GIFT. Формат GIFT розглядається як цільова мова подання результатів роботи системи, що забезпечує сумісність із механізмами імпорту тестів у Moodle (рисунок 1) та не вимагає від користувача додаткових знань мов розмітки або спеціалізованих інструментів редагування [18].

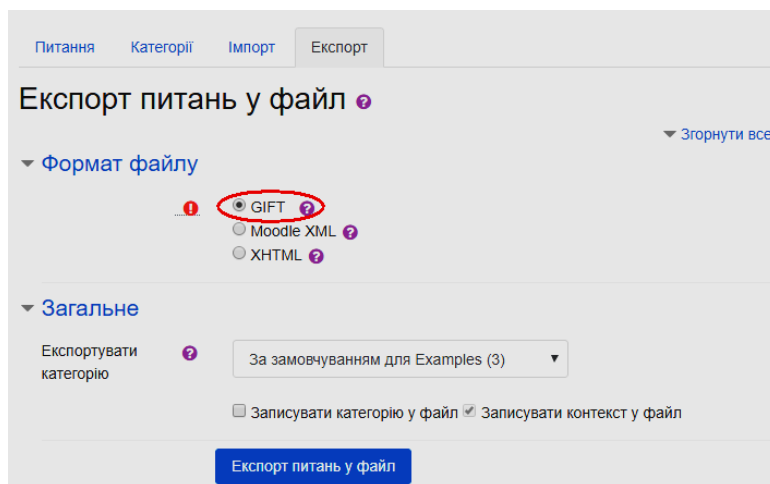


Рисунок 1 – Експорт питань у файл

У вихідному тексті підкреслено, що GIFT є простим текстовим форматом, здатним у межах одного файлу акумулювати велику кількість тестових завдань різних типів [19], підтримує запитання закритої та відкритої форми і допускає опис назви завдання, коментарів, вагових коефіцієнтів та інших параметрів, необхідних для організації тестування у Moodle.

На концептуальному рівні метод формування GIFT-файлів спирається на уявлення про дві взаємопов'язані площини подання тестового завдання – внутрішню об'єктну модель, яка використовується у програмному модулі TestGenerator, та зовнішнє текстове представлення, що відповідає формату GIFT. Внутрішня модель репрезентує завдання у вигляді екземпляра абстрактного класу BaseModel із заданими полями заголовка питання, множини варіантів відповідей, ознак правильності та, за потреби, додаткових параметрів [20]. Зовнішня форма задається через рядки текстового файлу, де питання починається з заголовка, далі в дужках фіксується перелік варіантів, кожен з яких маркується символом «=>» для правильної відповіді або «~>» для дистракторів, а структура файлу допускає також використання службових коментарів, які у розробленій системі позначають початок нового тестового завдання.

Ключовим завданням методу є побудова відображення між цими двома представленнями, яке забезпечує коректну і повну передачу інформації про форму і зміст тестового завдання. Для цього в базовому класі моделей формування завдань реалізовано уніфікований метод запису питання у файл WriteTestCase. Саме він інкапсулює конкретний синтаксис GIFT та приховує його від інших компонентів системи. Під час виклику цього методу модуль послідовно додає до вихідного потоку службовий коментар, що фіксує початок питання, текст заголовка з відповідним форматом, а потім у циклі обробляє всі елементи колекції відповідей, кожен з яких представлено об'єктом із полями «текст відповіді» та «ознака правильності». Для кожного варіанта за ознакою правильності обирається маркер «=>» або «~>», який у GIFT визначає, чи є відповідь правильною, після чого формується текстовий рядок, що містить цей маркер і власне текст відповіді. Завершальним кроком є запис закриваючої дужки і порожнього рядка, що відокремлює поточне питання від наступного.

Такий підхід має кілька методологічних наслідків. По-перше, задається чітка межа між логікою побудови змісту завдання та механізмом його текстового подання. Моделі формування завдань концентруються на виборі термінів, тез, дистракторів, побудові формулювання питання та комбінуванні відповідей, тоді як метод WriteTestCase відповідає виключно за коректну упаковку отриманого змісту у формат GIFT. Це суттєво спрощує розширення системи: додавання нового типу тестового завдання потребує розроблення лише спеціалізованого підкласу

BaseModel із власною реалізацією побудови питання, але не вимагає змін у механізмі запису у файл [21]. По-друге, уніфікована процедура запису гарантує однорідність вихідних GIFT-файлів незалежно від конкретної моделі, що підвищує надійність імпорту до Moodle та мінімізує ризики синтаксичних помилок.

Центральну роль у реалізації методу формування GIFT-файлів відіграє метод `GenerateTests` класу `TestGenerator`, який організовує послідовність викликів моделей формування завдань. У вихідному коді цей метод приймає на вхід тип тестового завдання, кількість питань, що мають бути сформовані, та засіб запису у файл. Перед генерацією система коригує бажану кількість питань з урахуванням розміру доступної множини термінів, щоб уникнути ситуацій, коли запитується більше завдань, ніж дозволяє інформаційний ресурс. Використовуючи список уже використаних індексів, метод забезпечує невідтворення однакових термінів у межах однієї сесії генерації, що підвищує різноманітність сформованого тесту. Для кожного нового завдання випадково обирається термін, на основі якого, через фабричний метод `ModelFactory`, створюється відповідна модель `BaseModel`. Далі послідовно викликаються методи `BuildTestCase`, який формує питання і набір відповідей відповідно до обраної моделі, та `WriteTestCase`, який записує результат у GIFT-файл.

Таким чином формується ієрархія викликів, де `GenerateTests` відповідає за стратегічну організацію процесу (вибір терміну, типу завдання, кількості питань), моделі типу `BaseModel` – за побудову змісту конкретного завдання, а метод `WriteTestCase` – за текстову реалізацію сформованого завдання у форматі GIFT. Важливо, що всі ці рівні взаємодіють через чітко визначені інтерфейси, що дає змогу розглядати метод формування GIFT-файлів як окремий компонент в архітектурі інформаційної технології, який може бути використаний повторно в інших системах, орієнтованих на Moodle.

Метод формування GIFT-файлів інтегровано з попередніми етапами обробки навчальних текстів та побудови моделей формування завдань. На вході методу перебуває множина термінів і тез, структурована відповідно до понятійно-тезової моделі навчального матеріалу та описана набором тегів, що використовуються в правилах конвертації.

Для кожного обраного терміна моделі формування завдань, спираючись на ці теги, генерують основу питання, коректну відповідь і один або декілька дистракторів. Метод `GenerateTests` забезпечує послідовне застосування цих моделей до елементів вхідної множини, тоді як запис у файл реалізує перехід від внутрішнього подання завдання до зовнішнього текстового представлення. Це дозволяє розглядати формування GIFT-файлу як завершальний етап конвеєра, що починається з аналізу лекційного тексту, виділення термінів, побудови тегової

структури і завершується готовим набором завдань, доступним для імпорту у Moodle.

Особливе значення для практичної придатності методу має підтримка різних типів тестових завдань, передбачених форматом GIFT. У вихідному тексті відзначено, що GIFT дозволяє описувати завдання типу вибір з множини, вірно–невірно, коротка відповідь, вибір відповідності, числова відповідь, доповнювати їх коментарями та ваговими коефіцієнтами, а також акумулювати в одному файлі всі завдання незалежно від типу.

У розробленій інформаційній технології це реалізується через окремі моделі для кожного типу завдання, які успадковуються від BaseModel і перевизначають логіку побудови колекції відповідей. Метод WriteTestCase при цьому залишається спільним, оскільки формат подання варіантів відповідей у GIFT уніфіковано через маркери « $\Rightarrow$ » і « $\sim$ » та фігурні дужки, що обрамляють перелік відповідей. Така організація коду дозволяє надалі розширювати спектр підтримуваних типів завдань, зберігаючи загальний механізм формування GIFT-файлу незмінним.

Метод формування GIFT-файлів має також експлуатаційні переваги, пов'язані з простотою зберігання і перенесення сформованих тестів. Оскільки результати роботи системи подаються у вигляді звичайного текстового файлу з розширенням, що інтерпретується Moodle як GIFT, користувач може архівувати, копіювати, версійно контролювати та редагувати цей файл без спеціалізованих засобів. При необхідності вручну виправити окремі формулювання або додати коментарі достатньо відкрити GIFT-файл у текстовому редакторі, що знижує вхідний поріг для викладача й робить систему більш прозорою. У той же час завдяки суворому дотриманню синтаксису GIFT, яке забезпечується методом WriteTestCase, зберігається гарантія коректного імпорту тестів у Moodle без додаткової ручної адаптації (рисунок 2).

З точки зору архітектури інформаційної системи, описаний метод формування GIFT-файлів функціонує як модуль вихідної серіалізації, який використовує абстракції вищих рівнів, не впливаючи на них. Розробник може модифікувати або розширювати моделі формування завдань, змінювати правила конвертації, вводити нові теги, не змінюючи базовий механізм запису у формат GIFT. У граничному випадку можливо навіть реалізувати альтернативні модулі серіалізації для інших форматів, наприклад, Moodle XML або QTI, залишивши незмінними алгоритми формування змісту тестових завдань. Така модульність є важливою передумовою масштабованості та довготривалої підтримки системи.

Окремим аспектом методу є використання механізмів випадкового вибору при формуванні набору завдань. У фрагментах коду, наведених у роботі, показано

використання генератора псевдовипадкових чисел та структур спискового контролю вже використаних індексів, що забезпечує нерівномірність розподілу термінів і запобігає повторенню однакових завдань у межах одного тесту.

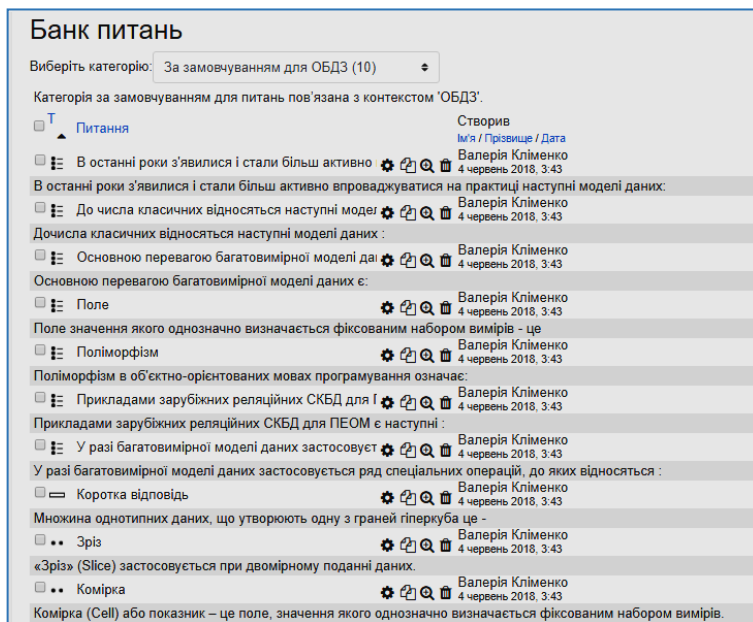


Рисунок 2 – Сформовані тестові завдання після імпорту до середовища Moodle

Це дозволяє на основі однієї й тієї самої множини термінів створювати різні варіанти тестів, що є важливим для організації поточного та підсумкового контролю знань. У поєднанні з можливостями GIFT щодо вагових коефіцієнтів та коментарів це створює підґрунтя для адаптивних сценаріїв тестування, де складність і структура тесту можуть змінюватися залежно від педагогічних цілей.

Узагальнюючи, метод формування GIFT-файлів для середовища Moodle у рамках розробленої інформаційної технології являє собою формалізований процес переходу від внутрішніх моделей тестових завдань до текстового представлення, стандартизованого у форматі GIFT. Він базується на чіткій об'єктній моделі завдання, уніфікованому механізмі серіалізації, використанні фабричних методів для побудови завдань різних типів, механізмах випадкового вибору термінів і надійній інтеграції з тегованим поданням навчального матеріалу. Така організація забезпечує сумісність із Moodle, підтримку широкого спектра типів завдань,

можливість масштабування та розширення системи, а також практичну зручність використання сформованих GIFT-файлів у навчальному процесі.

### Перелік посилань

1. Як школи впроваджують технології в освіту. News IO. URL: <https://io.ua/yak-shkoly-integruyuetsya-v-tsyfrovyj-svit/>
2. Методика використання тестової перевірки знань. Vseosvita. URL: <https://vseosvita.ua/library/metodika-vikoristanna-testovoi-perevirki-znan-204638.html>
3. Використання тестів в освітньому процесі. Vseosvita. URL: <https://vseosvita.ua/library/vikoristanna-testiv-v-osvitnomu-procesi-487976.html>
4. Multiple Choice Questions Types With Examples. SurveySparrow. URL: <https://surveysparrow.com/blog/multiple-choice-questions/>
5. Класифікація тестів. Форми тестових завдань. StudFiles. URL: <https://studfile.net/preview/8839472/page:72/>
6. Murava V., Zalutska O., Didur V., Mazurets O. Software architecture of information system for exchanging LLM thematic prompts. Global Trends in the Development of Information Technology and Science. Proceedings IV International Scientific and Practical Conference. June 25-27, 2025. Stockholm, Sweden. Pp. 121-127.
7. Mazurets O., Vit R. Practical Application of Method of Thematic Classification of Text Information Using LDA. Information Technology and Implementation (Satellite). Proceedings 11th International Conference. November 21, 2024. Kyiv, Ukraine. 2024. Pp. 151-152.
8. Yurchenko D., Mazurets O., Didur V., Molchanova M. Approach to Using Cloud Services for Visual Analytics of Neural Network Analysis of Texts Emotional Tonality. The Future of Scientific Discoveries: New Trends and Technologies. Proceedings of the XLVII International scientific and practical conference. November 13-15, 2024. Marseille, France. 2024. Pp. 108-113.
9. Blazhuk V., Mazurets O., Zalutska O. An Approach to Using the mBERT Deep Learning Neural Network Model for Identifying Emotional Components and Communication Intentions. The Impact of Scientific Research on the Development of the Modern World. Proceedings of the XLIV International scientific and practical conference. October 23-25, 2024. Dubrovnik, Croatia. 2024. Pp. 79-84.
10. Mazurets O., Molchanova M., Klimenko V., Prosvitliuk M Practice Implementation of Neural Network Model BART-Large-CNN for Text Annotation. Prospects of Scientific Research in the Conditions of the Modern World. Proceedings of XXVII International scientific and practical conference. June 12-14, 2024. Rotterdam, Netherlands. 2024. Pp. 97-102.
11. Shevchuk P., Molchanova M., Mazurets O. Software for Text Messages Reliability Analysis Based on the Machine Learning Models Ensemble. Proceedings of IV International Scientific and Practical Conference «Innovative research and perspectives of the development of science and technology». January 29-31, 2024. Stockholm, Sweden. 2024. Pp. 347-354.
12. Мазурець О.В., Тищенко О.О., Гардиш Д.О. Реляційна даталогічна модель для прикладного аналізу репрезентативності навчальних тестів засобами обробки природної мови. Наука і техніка сьогодні. 2025. № 13(41). С. 1129-1142.

13. Мазурець О.В., Собко О.В., Гардиш Д.О. Прикладні аспекти оцінювання релевантності бази тестових завдань семантичній структурі навчальних матеріалів засобами штучного інтелекту. Інформаційні технології і автоматизація. Матеріали XVII міжнародної науково-практичної конференції. Одеса, ОНТУ, 2024. С. 381–383.
14. Гардиш Д.О., Мазурець О.В. Метод автоматизованого оцінювання відповідності тестових завдань семантичній складовій навчальних матеріалів за інтелектуальним аналізом їх текстового контенту. Збірник наукових праць за матеріалами XVI Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2024». Хмельницький, 2024. С. 121–139.
15. Hardysh D., Mazurets O., Tyschenko O. Datalogic Relation Model for Automated Evaluating the Semantic Integrity of Test Tasks Sets by Machine Learning Means. Innovative Solutions in Science: Balancing Theory and Practice. Proceedings 2nd International Scientific and Practical Conference. December 23–25, 2024. San Francisco, USA, 2024. P. 114–125.
16. Hardysh D., Tyschenko O., Mazurets O. Intelligent System for Automated Assessment of Test Tasks Sets Conformity to Semantic Structure of Educational Materials. The Role of Science and Technology in Solving Global Problems of Humanity. Proceedings LIII International Scientific and Practical Conference. December 25–27, 2024. Vienna, Austria, 2024. P. 110–115.
17. Гардиш Д.О., Мазурець О.В. Метод автоматизованого аналізу відповідності множин тестових завдань семантичній структурі навчальних матеріалів засобами штучного інтелекту. Сучасні інформаційні технології в освіті і науці. Збірник матеріалів VI Всеукраїнської науково-практичної конференції. 14–15 листопада 2024. Умань, 2024. С. 43–48.
18. Гардиш Д.О., Кліменко В.І., Мазурець О.В. Підхід до аналізу відповідності множини тестових завдань семантичній структурі навчальних матеріалів засобами обробки природної мови. Тези доповідей XII Міжнародної науково-практичної конференції «Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій». 10–12 грудня 2024. Запоріжжя, Національний університет «Запорізька політехніка», 2024. С. 344–348.
19. Крива Д.О., Мазурець О.В., Кліменко В.І., Тищенко О.О. Підхід до автоматизованого визначення рівня емпатії з використанням нелінійного тестування. Збірник наукових праць за матеріалами XVI Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2024». 15-16 листопада 2024. Хмельницький, 2024. с. 316-324. URL: <https://kn.khmnu.edu.ua/wp-content/uploads/sites/18/apkn-2024-corporpaper.pdf>
20. Шебетко О.В., Кліменко В.І., Мазурець О.В. Метод адаптивного тестування з використанням продукційних правил. Збірник наукових праць за матеріалами XV Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2023». Хмельницький, 2023. с. 311-322.
21. Бармак О.В., Крак Ю.В., Мазурець О.В., Манзюк Е.А. Ментально-формальні рішення машинного навчання для інформаційної технології автоматизованого створення тестів у сфері безпеки та медицини. Матеріали X Міжнародної науково-практичної конференції «Інформаційні управляючі системи та технології ICST-ODESSA-2021». Одеса – 2021. – С.105-107.