

УДК 004.415.538:37.091.212-048.24]:111.1

DOI: <https://doi.org/10.31866/2617-796X.9.1.2026.362595>**Ольга Ткаченко,**

кандидат фізико-математичних наук, доцент,
доцент кафедри інформаційних технологій,
Навчально-науковий інститут управління,
технологій та правових наук,
Національний транспортний університет,
Київ, Україна
oitkachen@gmail.com
<https://orcid.org/0000-0003-1800-618X>

Надія Музиченко,

магістрантка кафедри інформаційних технологій,
Навчально-науковий інститут управління,
технологій та правових наук,
Національний транспортний університет,
Київ, Україна
nadya.muzychenko@gmail.com
<https://orcid.org/0009-0009-1067-7254>

ВЕБОРІЄНТОВАНА СИСТЕМА ТЕСТУВАННЯ ЗНАТЬ СТУДЕНТІВ: ОНТОЛОГІЧНИЙ ПІДХІД

На сьогодні розширюється сфера застосування навчальних платформ, тому дослідження та аналіз наявних програмних рішень щодо забезпечення процесів навчання, зокрема процесу тестування знань студентів, є актуальними і обумовлюють формування нового підходу, а саме онтологічного, до розробки сучасних веборієнтованих систем, що забезпечують інформатизацію та інтелектуалізацію освітнього процесу в цілому та процесу тестування зокрема.

Метою статті є дослідження та аналіз сучасних програмних засобів (систем, платформ), призначених для забезпечення процесів навчання, зокрема тестування, формування відповідного онтологічного підходу до інтелектуалізації цих процесів та розробки на його основі системи тестування знань студентів.

Методами дослідження є методи системного підходу та порівняльного аналізу основних програмних рішень предметної області (система тестування знань студентів). У статті розглянуто онтологічний підхід до розробки веборієнтованої системи тестування знань студентів.

Новизною проведеного дослідження є аналіз сучасних методів і програмних рішень (систем, платформ) навчання та тестування знань студентів, формування онтологічного підходу до інтелектуалізації процесу тестування та розробки відповідної веборієнтованої системи.

Висновки. Проведене дослідження спрямоване на підвищення ефективності процесу тестування. Аналіз сучасних навчальних платформ та запропонований онтологічний під-

хід покладено в основу розробки відповідної веборієнтованої системи тестування знань студентів. У роботі проведено аналіз визначеної предметної області, аналіз сучасних навчальних платформ; запропоновано онтологічний підхід до тестування знань студентів та його програмної реалізації; проаналізовано основні методи генерації номерів тестових питань, що надають студентам для перевірки їхніх знань; розроблено веборієнтовану систему тестування знань студентів, яка сприяє, зокрема, автоматизації процесу тестування знань студентів, зменшенню рівня навантаження на викладачів через генерацію тестових завдань, підвищенню рівня якості навчання студентів, підтримці простого та комфортного користувальницького інтерфейсу та вирішенню можливих конфліктних ситуацій.

Ключові слова: освітній процес; тестування знань студентів; веборієнтована система тестування знань студентів; база даних тестів; база даних відповідей; онтологія; онтологічне моделювання предметної області.

Вступ. Під час загальної цифровізації (інформатизації, інтелектуалізації) важливим призначенням будь-якої інформаційної системи є оперативне забезпечення студента всім обсягом необхідної інформації через реалізацію відношень типу:

<питання → відповідь>,

які надають можливість виділення предметної області (ПрО), що втілюється у відповідній інформаційній навчальній системі (Організація баз даних та знань, 2015; Ткаченко, 2022; Кудін, Кудіна та Кархут, 2021; Милютіна, 2011).

Інформацію про ПрО часто подають у формі, що обмежує можливості змістовної інтерпретації інформації та конкретизує семантику її подання у відповідній інформаційній системі. Сукупність об'єктів (сутностей, реалій, речей), щодо яких можна ставити запитання, утворює об'єктне ядро ПрО, яке має онтологічний статус. Об'єктами (сутностями) в ПрО «Навчальний процес: тестування», зокрема, є:

- окремий студент;
- група студентів;
- аудиторія;
- час занять;
- слова, числа, символи, які є складниками освітнього контенту.

З об'єктами пов'язано два аспекти: ідентифікація і адекватний опис. Для ідентифікації використовують ім'я, зокрема вказівну функцію імені. Ім'я – це прямий спосіб ідентифікації об'єкта. До непрямих способів ідентифікації об'єкта зараховують визначення об'єкта через його властивості (характеристики або ознаки).

Об'єкти взаємодіють між собою через свої властивості, що породжує ситуації. Ситуації – це взаємозв'язки, які відображають відношення між об'єктами. Ситуації у ПрО описуються за допомогою висловлювань про ПрО з використанням виразів і обчислень предикатів, тобто формальної, математичної логіки. Тому проблему дослідження та аналізу наявних програмних рішень щодо забезпечення тестування знань студентів, що є невід'ємною частиною освітнього процесу, визначено актуальною і такою, яка потребує вирішення для формування підходу, зокрема онтологічного, до розробки сучасних веборієнтованих систем з метою

забезпечення інформатизації та інтелектуалізації освітнього процесу в цілому та процесу тестування зокрема.

Мета і завдання дослідження. Метою є дослідження та аналіз сучасних програмних засобів (систем, платформ), призначених для забезпечення процесів навчання, зокрема тестування, формування відповідного онтологічного підходу до інтелектуалізації цих процесів та розробки на його основі системи тестування знань студентів.

Досягнення мети передбачає виконання таких завдань:

- визначити основні проблеми інформатизації та інтелектуалізації процесів навчання (у тому числі й процесу тестування знань студентів);
- проаналізувати методи випадкового вибору тестів для студентів;
- провести аналіз наявного програмного забезпечення сучасних навчальних систем і платформ, які використовують для тестування знань;
- розробити онтологічний підхід до інтелектуалізації процесів навчання (у тому числі й процесу тестування знань студентів);
- розробити на основі запропонованого онтологічного підходу відповідну систему тестування знань студентів.

Використання в освітньому процесі сучасних інформаційних технологій сприяє реалізації різноманітних методик викладання навчального матеріалу. До таких перспективних можливостей можна зарахувати, зокрема, моделювання різноманітних процесів у динаміці їх розвитку.

Створення інформаційних навчальних систем забезпечує можливість відійти від застосування пасивних традиційних методик навчання, характерних, наприклад, для автоматичного вичитування матеріалу лекцій за допомогою електронних підручників (Ілійчук, 2020).

У сучасній освіті часто використовують тести. Вибір варіанта з декількох – це не єдина форма тестування. В опитування можна вставляти картинки та відео, змінювати форми завдання тощо (Сім платформ для створення тестів, 2015; Tkachenko, 2022; Кудін, Кудіна та Кархут, 2021; Милютіна, 2011).

Одним із сучасних підходів до тестування є використання безкоштовної Google Forms (A flexible and secure foundation for teaching and learning, n.d.), яка є частиною офісного інструментарію Google Drive (Google Диск, б.д.). Це один з найшвидших і простих способів створити тест, який можна відправити студентам електронною поштою чи надати на своєму сайті. Для прискорення роботи доцільно додати плагін Flubaroo, який автоматично перевіряє відповіді та ставить оцінки відповідно до заданих критеріїв.

За допомогою платформи Quizlet (Як ви хочете займатися?, б.д.) можна створювати тести, у яких студенти будуть вибирати правильні відповіді із запропонованих, зіставляти зображення та інформацію або вписувати власні варіанти. Quizlet підтримує українську мову і працює в операційних системах Android та iOS (рис.1). Quizlet – безкоштовна платформа, але є і підписка Plus для завантажування власних картинок і створення великої кількості навчальних груп.

Платформа ProProfs (Protégé is a free, open-source OWL ontology editor, n.d.) забезпечує підготовку різних тестів (наприклад, вибір одного чи декількох варіантів відповідей, заповнення пропущеного слова, написання розгорнутої відпові-

ді). ProProfs надає можливість вставляти в завдання текстові документи, презентації, PDF-файли, зображення, аудіо- та відеофайли.

Завершивши роботу над тестом, його можна залишити в загальному доступі на сайті ProProfs або вбудувати на свою сторінку.

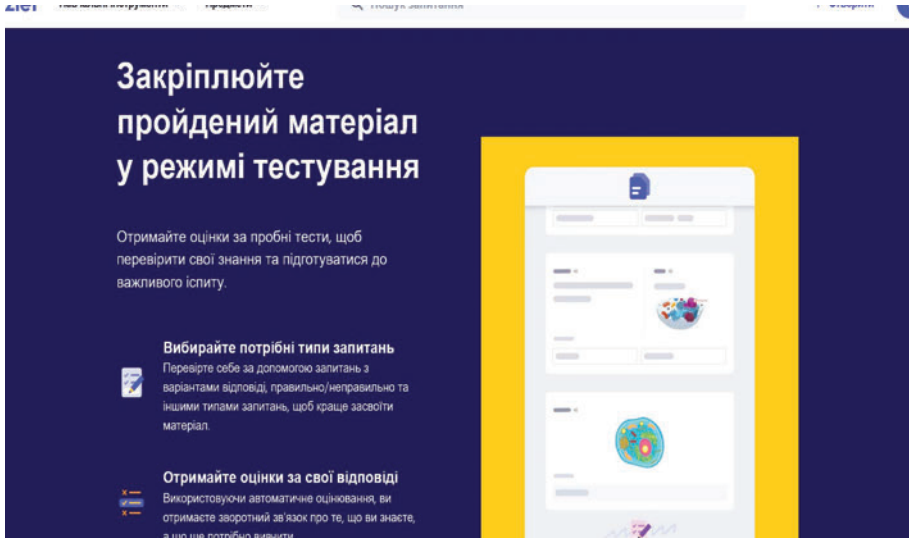


Рис. 1. Головна сторінка системи тестування Quizlet

Джерело: (Як ви хочете займатися?, б.д.).

ProProfs – безкоштовний, але його можливості розширюються в платних підписках Basic і Professional. Головну сторінку системи тестування ProProfs показано на рис. 2.



Рис. 2. Головна сторінка системи тестування ProProfs

Джерело: (Delightfully Smart Tools, n.d.).

Результати дослідження. Для розробки системи тестування слід обрати алгоритми, які сприятимуть автоматичній генерації тестових завдань. Одним з таких алгоритмів є використання генераторів випадкових чисел для створення масиву номерів питань, які будуть діставатися з бази даних (БД).

Суть алгоритму полягає у використанні генератора випадкових чисел для створення унікального набору ідентифікаторів питань, які будуть обрані з БД. Генератор випадкових чисел забезпечує непередбачуваність вибору питань, що мінімізує можливість підготовки до конкретного набору питань і підвищує об'єктивність тестування. Процес генерації масиву номерів питань передбачає:

- ініціалізацію генератора випадкових чисел;
- вибір необхідної кількості унікальних номерів питань;
- збереження масиву номерів для подальшого використання.

Ефективність цього алгоритму полягає в його здатності швидко та надійно генерувати унікальні набори питань з великого пулу наявних у БД. Це дає змогу зменшити ймовірність повторення однакових тестових завдань для різних студентів і забезпечити різноманітність тестів.

Упровадження такого алгоритму є достатньо простим, бо генерація випадкових чисел та робота з масивами є стандартними функціями в більшості мов програмування. Основні складнощі можуть виникнути на етапі інтеграції з БД, де необхідно забезпечити ефективне та швидке отримання питань за номерами, а також обробку можливих колізій або помилок під час доступу до БД.

Після генерації масиву номерів питань виконується звернення до БД для отримання самих питань. Цей процес охоплює:

- формування запиту до БД, використовуючи згенерований масив номерів тестових питань, завдань тощо;
- отримання результатів запиту у вигляді набору питань;
- відображення питань у тестовому інтерфейсі для студента.

Алгоритм забезпечує автоматизацію процесу підготовки тестових завдань, що підвищує ефективність та об'єктивність освітнього процесу.

Алгоритм генерації випадкових чисел, відомий як метод серединних квадратів (Генерація випадкових чисел, 2014), застосовували для обчислення послідовностей псевдовипадкових чисел за допомогою квадратичного множення початкового значення та використання середніх цифр результату.

Метод лінійного конгруентного генератора (Що таке комбінований лінійний конгруентний метод, 2025) є одним з найпростіших і найпоширеніших алгоритмів генерації псевдовипадкових чисел, який використовує формулу:

$$X_{n+1} = (aX_n + c) \bmod m \quad (1),$$

де X – послідовність випадкових чисел, a і c – константи, а m – модуль. Цей метод є ефективним і швидким, що зробило його дуже популярним у комп'ютерних програмах.

На сьогодні використовують більш складні та надійні алгоритми генерації випадкових чисел, такі як Mersenne Twister (What is Mersenne Twister?, n.d.), що відомий своєю довгою періодичністю і високою швидкістю генерації.

Ефективність генераторів випадкових чисел забезпечується їхньою здатністю швидко генерувати великі набори унікальних значень. Це особливо важливо для навчальних систем, де потрібна швидка та надійна генерація тестів для великої кількості студентів. Порівняно з іншими алгоритмами, такими як метод послідовного вибору або ручний вибір питань, генератори випадкових чисел значно скорочують час та ресурси, необхідні для підготовки тестових завдань. Використання генераторів випадкових чисел для автоматичної генерації тестових завдань є ефективним і практичним рішенням для системи тестування. Цей підхід забезпечує об'єктивність та різноманітність тестування, що сприяє підвищенню якості освітнього процесу.

Алгоритм оцінювання відповідей у системі тестування ґрунтується на порівнянні відповідей студентів з правильними відповідями, збереженими в БД. Перший крок полягає в отриманні відповідей студентів з БД, які були збережені під час проходження тесту. Для кожного питання, на яке студент дав відповідь, здійснюється порівняння його відповіді з правильною, яка зберігається у БД. Якщо відповідь студента збігається з правильною відповіддю, вона позначається як правильна. Якщо ж відповідь не збігається – вона позначається як неправильна. Після оцінювання всіх відповідей студента підраховується загальна кількість правильних, на основі чого визначається підсумковий бал за тест.

Перевагами такого алгоритму оцінювання, зокрема, є:

- автоматизація процесу оцінювання: алгоритм дає змогу автоматизувати процес оцінювання, що значно зменшує час та зусилля, необхідні для перевірки великої кількості відповідей;
- об'єктивність оцінювання: оцінювання здійснюється за чіткими критеріями (відповідність або невідповідність правильної відповіді);
- швидкість та ефективність: порівняння відповідей студентів з правильними відповідями здійснюється швидко, що важливо в процесі обробки багатьох тестів одночасно.

Недоліками такого методу є вразливість до помилок в БД та обмеженість у типах питань. Якщо правильні відповіді у БД некоректні, то це призведе до неправильного оцінювання відповідей студентів. Алгоритм добре працює для питань з однозначною правильною відповіддю (наприклад, тестові питання з вибором варіантів), але є менш ефективним для оцінювання відкритих питань або відповідей у вільній формі.

Алгоритми на основі машинного навчання можуть забезпечувати гнучкіше оцінювання, особливо для відкритих питань, але потребують значних обчислювальних ресурсів і часу на навчання відповідних нейромережових моделей. Якщо порівнювати з ними, описаний алгоритм є простішим і менш ресурсоємним.

Точність такого алгоритму оцінювання залежить від якості даних у БД. Якщо правильні відповіді зберігаються без помилок, алгоритм забезпечує високу точність оцінювання. Алгоритм гарантує, що кожна відповідь студента буде перевірена відповідно до правильної відповіді, що забезпечує об'єктивність і точність у виставленні оцінок.

Отже, описаний алгоритм є ефективним і точним методом автоматичного оцінювання відповідей у тестових завданнях, що дає змогу значно підвищити про-

дуктивність та об'єктивність освітнього процесу. Онтологія в інформаційних системах є формальним представленням знань у певній предметній області (ПрО) (Болотова, 2010; Tkachenko, 2022). Компонентами онтології є класи, індивіди, властивості та відношення. Класи – абстрактні категорії об'єктів, які мають спільні властивості. До індивідів зараховують конкретні об'єкти або екземпляри класів. Властивості описують відношення між об'єктами або атрибути об'єктів, тому вони поділяються на об'єктні й атрибутивні.

Об'єктні властивості описують відношення між двома об'єктами. Наприклад, властивість з назвою *hasStudent* вказує на відношення між об'єктами *курс* і *студенти*.

Атрибутивна властивість представляє відношення між індивідом та даними. Наприклад, атрибутивною властивістю студента може бути його дата народження або вік. Відношення демонструють зв'язки підкласів з класами. Наприклад, класи ІПЗ, КН є підкласами класу «*Спеціальності*».

Онтології, зокрема:

- сприяють стандартизації термінів та визначень, що полегшує обмін інформацією між різними системами;
- забезпечують єдину модель для інтеграції даних, отриманих з різних джерел, підвищуючи рівень узгодженості даних;
- допомагають організувати знання в певній ПрО, що полегшує пошук і використання;
- сприяють автоматизації аналізу даних, наприклад знаходження прихованих закономірностей або аномалій;
- покращують результати пошуку завдяки врахуванню значень термінів, а не лише ключових слів.

В онтології, розробленій у редакторі онтологій *Protege* (*Protégé is a free, open-source OWL ontology editor, n.d.*) для системи тестування, є такі класи (рис. 3):

1. Дисципліни.
2. Основи програмної інженерії.
3. Основи програмування.
4. Проектування систем з розподіленими БД.
5. Технології створення OLAP-систем.
6. Курс.
7. Групи.
8. Спеціальності.
9. ІПЗ.
10. КН.
11. Студенти.
12. Тести.
13. Важкі тести.
14. Середні тести.
15. Легкі тести.

Класи «Основи програмної інженерії», «Основи програмування», «Проектування систем з розподіленими БД», «Технології створення OLAP-систем» є підкласами до класу «Дисципліни». Клас «Групи» є підкласом класу «Курс».

Класи «Важкі тести», «Середні тести», «Легкі тести» є підкласами класу «Тести».

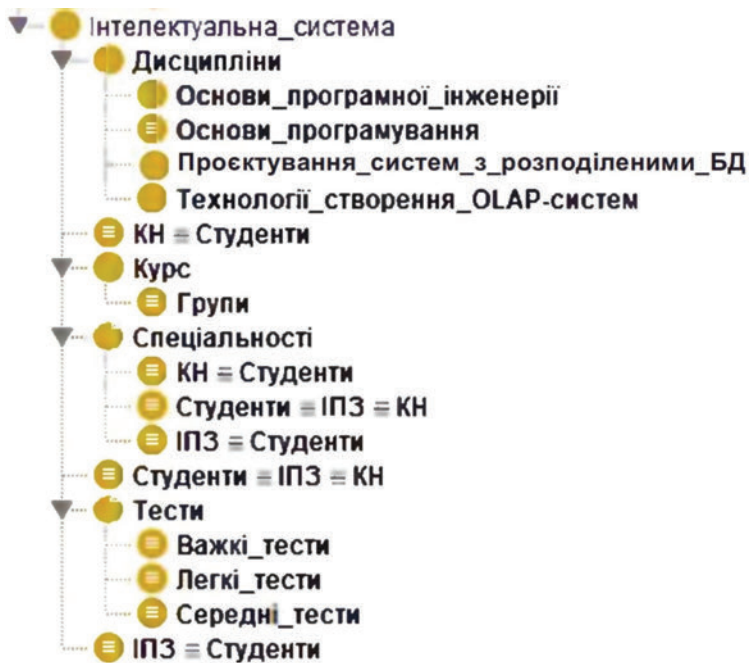


Рис. 3. Онтологія системи тестування
Джерело: авторська розробка

Клас «Студенти» пов'язаний з класом «Групи» за допомогою об'єктної властивості `hasStudent`. Також цей клас пов'язаний з класами «Важкі тести», «Середні тести», «Легкі тести» за допомогою об'єктної властивості `hasTakenTest`.

Класи «Основи програмної інженерії», «Основи програмування», «Проєктування систем з розподіленими БД», «Технології створення OLAP-систем» пов'язані з класом «Дисципліни» за допомогою `isSubject`, яка вказує що ці класи є дисциплінами.

Система тестування знань розроблена мовою програмування C# (C# documentation, n.d.) в середовищі Visual Studio (Dream big, n.d). Вона має сім основних форм, що пов'язані між собою через звернення до БД, яку створено в MS SQL Server (Microsoft SQL Server 2025, n.d.). Першою формою є «Система тестування» (є блоком авторизації), у якій за введеними логіном і паролем виконується перевірка наявності такого студента в БД системи, отримання ID студента з бази. Цю операцію виконує функція `CheckIfUserExists()`, що викликається в методі `button1_Click()`, коли студент натискає на кнопку Увійти.

Після виклику методу `button1_Click()` і виконання функції `CheckIfUserExists()` ID студента вноситься в змінну `userID`, яка переноситься між формами, та відкривається форма «Оберіть дію». За отриманим значенням змінної `userID` функція `DBConnection()` виконує пошук в БД дисциплін, які відповідають спеціальності та курсу студента.

Форма «Оберіть рівень складності» містить три кнопки, які представляють рівень складності. Якщо натиснути на ці кнопки, то можна викликати метод

button1_Click(), button2_Click() або button3_Click(), що зчитує назву обраної дисципліни, присвоює цю назву змінній selectedSub та передає разом з userID на відповідну форму (разом з тестом обраного рівня складності).

Метод button4_Click() – на формі «Оберіть дію», що призначена для передачі змінних selectedSubject та userID на форму «Результати та рекомендації». На цій формі під час відкриття викликається функція GetPrefixBySubject(), яка отримує префікс дисципліни з таблиці *subjects* та вносить її в змінну subjectPrefix.

Функція LoadWrongAnswers() порівнює відповіді студента на питання тестів з правильними відповідями; аналізує, у яких темах дисципліни було надано неправильні відповіді; підраховує кількість неправильних відповідей з кожної теми та викликає функцію CalculateCorrectAnswers(), яка на основі правильних відповідей студента підраховує оцінку за дисципліну.

Після виконання функції в resultLabel виводиться оцінка в рейтинговій системі ECTS, а також теми, на які було надано неправильні відповіді, та кількість неправильних відповідей у відповідних темах. Також у масиві wrongAnswersByTheme вказуються назви тем дисципліни, з яких було надано неправильні відповіді.

Під час натискання кнопки Отримати рекомендації викликається метод GetRecommendationsButton_Click(). Тут ініціалізується змінна themeName, яка отримує значення теми з масиву wrongAnswersByTheme. Далі створюється змінна themeID, яка отримує значення з функції GetThemeIDByName() за значенням themeName. Вказана функція опрацьовує значення змінної під час виклику select до таблиці themes.

Потім створюється змінна themeLink, яка отримує посилання на навчальні матеріали з теми через функцію GetThemeLinkById() за значенням змінної з попередньої функції. Теми та посилання на навчальні матеріали виводяться для студента в themeLinkLabel. На формах, де виводиться тест обраного рівня складності, викликається функція GetPrefixBySubject().

Значення, отримане з цієї функції, передається у змінну selectedPrefix, яка використовується у функції RetrieveRandomQuestionsFromDatabase(). Вона створює масив рандомних значень від 1 до 100. Ці значення використовуються для отримання 20 рандомних питань з відповідного тесту з БД. Потім ці питання разом з варіантами відповідей вносяться в словник questionData.

Цей словник опрацьовується у функції DisplayCurrentQuestion(), за допомогою якої виводиться запитання та варіанти відповідей в label та radioButton. Функція SaveUserAnswerToDatabase() викликається в методі button1_Click(), натиснувши кнопку Відповісти. Зчитує значення radioButton за допомогою функції GetSelectedAnswer() та вносить нову відповідь до БД або змінює значення вже наявної відповіді, якщо студент скористався методом button2_Click(), натиснувши кнопку Попереднє.

Після встановлення програмного продукту на ПК та його запуску студенту надається форма «Система тестування», через яку він може увійти в систему, увівши логін та пароль. Коли студент натискає кнопку Увійти, програма зчитує інформацію з введених полів і, якщо інформація введена правильно і відповідну інформацію про студента знайдено в БД, виводить повідомлення з текстом «Студента знайдено». Далі виконується перехід до наступної форми.

Якщо інформація, введена в полях, не відповідає записаній в БД, тобто відповідний запис про студента не буде знайдено, то буде виведено відповідне повідомлення з текстом «Студент відсутній в БД. Перевірте введені дані».

Після успішної авторизації студенту буде виведена форма «Оберіть дію» (рис. 4). У цій формі студент може обрати необхідну дисципліну з наведених у виглядковому списку й обрати наступний крок: «Перейти до тестування» або «Переглянути результати».

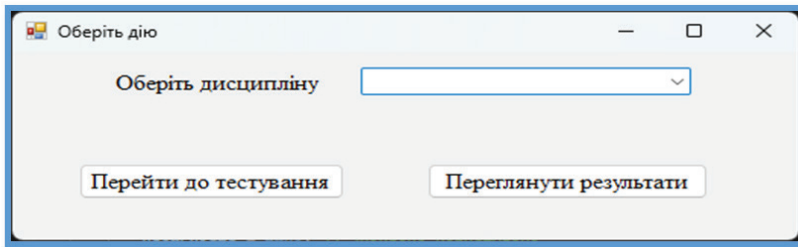


Рис. 4. Форма «Оберіть дію»
Джерело: авторська розробка

Кнопка Переглянути результати виводить форму «Результати та рекомендації», у якій студентові надаються результати, отримані на попередній формі з обраної дисципліни. Натиснувши кнопку Отримати рекомендації, студент отримує посилання на навчальні матеріали з тем дисципліни, де він, попередньо пройшовши тест будь-якого рівня складності, зробив помилки.

Кнопка Перейти до тестування виводить студентові форму «Оберіть рівень складності» (рис. 5), на якій можна обрати один з трьох рівнів складності тесту:

- легкий;
- середній;
- важкий.

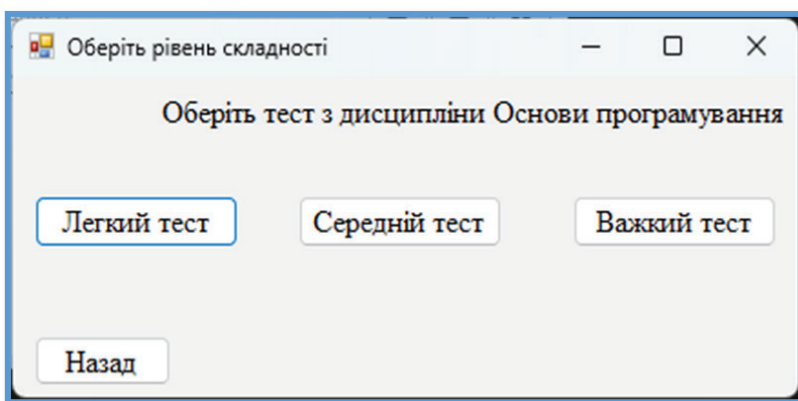


Рис. 5. Форма «Оберіть рівень складності»
Джерело: авторська розробка

Також студент може повернутися на попередню форму, натиснувши кнопку Назад. Обравши необхідний рівень складності, студенту буде виведено форму «Вікно тестування» (рис. 6).

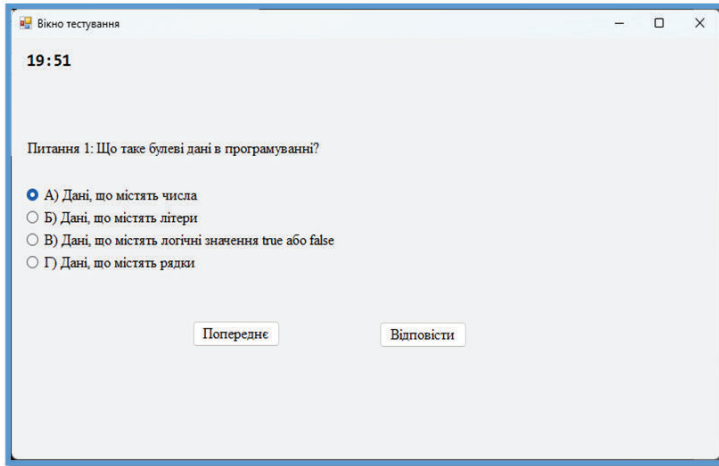


Рис. 6. Форма «Вікно тестування»

Джерело: авторська розробка

У цій формі студент може пройти тестування відповідного рівня складності, а потім отримати повідомлення з результатом та інформацію про те, з яких тем було надано неправильні відповіді (рис. 7).

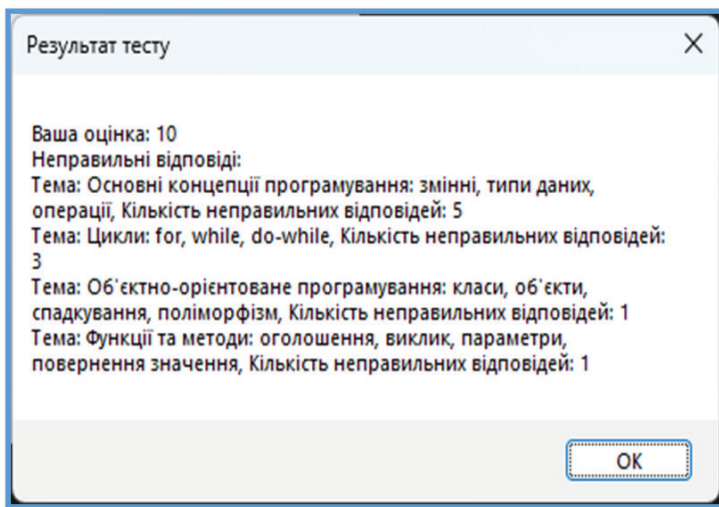


Рис. 7. Повідомлення «Результат тесту»

Джерело: авторська розробка

Після проходження тесту студент автоматично повертається до форми з вибором рівня складності тесту. На кожний рівень складності відведено фіксований проміжок часу, за який студент має пройти тестування. Для легкого рівня – 20

хвилин, для середнього – 30 хвилин, для важкого – 50 хвилин. Якщо студент не проходить тестування за відведений час, йому буде виведено повідомлення «Час вичерпано. Тест завершено». Тест буде провалено, вікно буде закрито, а студента повернуть на форму з вибором рівня складності.

Після проходження всіх тестів всіх рівнів складності (або лише одного) студент повертається на форму «Оберіть дію», де він ще раз обирає дисципліну та переходить до вікна «Результати та рекомендації» (рис. 8). На цій формі студент отримує дані про свою оцінку на момент проходження одного тестування або тестування всіх рівнів складності.

Система тестування знань студентів надає можливість дізнатися, у яких темах студент зробив помилки та в якій кількості питань було надано неправильні відповіді.

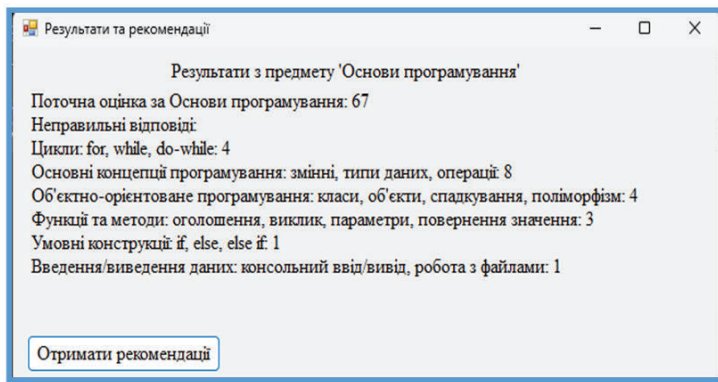


Рис. 8. Форма «Результати та рекомендації»

Джерело: авторська розробка

Якщо студент скористається кнопкою Отримати рекомендації, йому буде представлено список посилання на навчальні матеріали на основі тем, викладених на формі (рис. 9). Якщо студент натисне на будь-яке посилання, що наведено в цій формі, його буде направлено у вікно браузера за замовчуванням та виведено навчальний матеріал, що закріплено за цим посиланням.

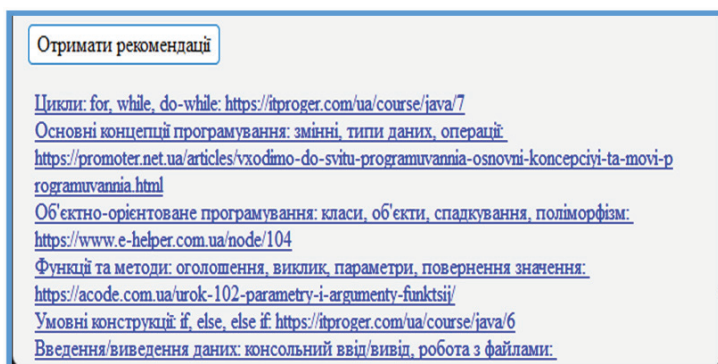


Рис. 9. Рекомендації студенту щодо навчальних матеріалів

Джерело: авторська розробка

Висновки. Проведене дослідження спрямоване на підвищення ефективності процесу тестування. Аналіз сучасних навчальних платформ та запропонований онтологічний підхід покладено в основу розробки відповідної веборієнтованої системи тестування знань студентів.

Отже, у роботі:

- проведено аналіз визначеної предметної області, сучасних навчальних платформ, призначених як для надання навчального контенту, так і для тестування знань студентів;
- запропоновано онтологічний підхід до тестування знань студентів та його програмної реалізації;
- проаналізовано основні методи генерації номерів тестових питань, що надають студентам для перевірки їх знань;
- розроблено веборієнтовану систему тестування знань студентів, яка сприяє, зокрема:
 - автоматизації процесу тестування знань студентів;
 - зменшенню рівня навантаження на викладачів через генерації тестових завдань;
 - підвищенню рівня якості навчання студентів за допомогою об'єктивної оцінки знань та надання рекомендацій щодо усунення так званих вузьких місць у знаннях здобувачів;
 - підтримці простого та комфортного користувальницького інтерфейсу, а також вирішенню можливих конфліктних ситуацій.

Розроблена веборієнтована система може бути впроваджена в освітній процес для тестування знань студентів, що навчаються за спеціальностями «Інженерія програмного забезпечення» та «Комп'ютерні науки».

СПИСОК ПОСИЛАНЬ

- Болотова, В., 2010. Онтології і подання знань. *Портал Знань*. [online] Доступно: <<http://www.znannya.org/?view=ontology-give-knowledge>> [Дата звернення 15 лютого 2026].
- Генерація випадкових чисел, 2014. *Godot Docs*. [online] Доступно: <https://docs.godotengine.org/uk/4.x/tutorials/math/random_number_generation.html> [Дата звернення 21 лютого 2026].
- Ілійчук, Л., 2020. Електронний підручник як інноваційний засіб навчання. *Освітні обрії*, [online] 1 (50), с.61-65. Доступно: <<https://lib-old.pnu.edu.ua/bitstream/123456789/11741/1/4076-Текст%20статті-8645-1-10-20200729.pdf>> [Дата звернення 23 лютого 2026].
- Кудін, А., Кудіна, Т. та Кархут, В., 2021. Технологія комп'ютерного тестування при змішаному навчанні в університеті. *Фізико-математична освіта*, [e-journal] 29 (3), с.93-98. <https://doi.org/10.31110/2413-1571-2021-029-3-014>
- Милютіна, І.М., 2011. Тестування як ефективний метод перевірки професійної компетентності студентів. *Урок.ОСВІТА.UA*, [online] 1 березня. Доступно: <http://osvita.ua/school/lessons_summary/edu_technology/15024/> [Дата звернення 25 лютого 2026].

- Організація баз даних та знань, 2015. *СумГУ*. [online] Доступно: <https://elearning.sumdu.edu.ua/free_content/lectureid:89b3d175c06a6b137e410cb14821d0e94549ad5a/20151030211833/44197/index.html> [Дата звернення 17 лютого 2026].
- Сім платформ для створення тестів, 2015. *Освіта.ua*, [online] 5 лютого. Доступно: <<https://osvita.ua/school/method/technol/45747/>> [Дата звернення 25 лютого 2026].
- Що таке комбінований лінійний конгруентний метод генерації випадкових чисел?, 2025. *Плани виставок. Покрокова думка*. [online] Доступно: <<https://night.kozak.cx.ua/ukraincyam/shho-take-kombinovaniy-liniyniy-kongruentniy-metod-generacii-vipadkovikh-chisel.html>> [Дата звернення 21 лютого 2026].
- Як ви хочете займатися?, б.д. *Quizlet*. [online] Доступно: <<https://quizlet.com/ua>> [Дата звернення 20 лютого 2026].
- A flexible and secure foundation for teaching and learning, n.d. *Google*. [online] Available at: <<https://edu.google.com/workspace-for-education/editions/education-fundamentals/>> [Accessed 23 February 2026].
- C# documentation, n.d. *Microsoft*. [online] Available at: <<https://learn.microsoft.com/uk-ua/dotnet/csharp/tour-of-csharp/>> [Accessed 17 February 2026].
- Delightfully Smart Tools For Smarter Employees & Happier Customers, n.d. *ProProfs*. [online] Available at: <<https://www.proprofs.com>> [Accessed 21 February 2026].
- Dream big. Achieve more. Visual Studio 2026, n.d. *Microsoft*. [online] Available at: <<https://visualstudio.microsoft.com/>> [Accessed 17 February 2026].
- Google Диск, б.д. *Хмарні технології в освіті*. [online] Доступно: <<https://sites.google.com/view/cloudinedu/google-диск>> [Дата звернення 20 лютого 2026].
- Microsoft SQL Server 2025, n.d. *Microsoft*. [online] Available at: <<https://www.microsoft.com/en-gb/sql-server>> [Accessed 17 February 2026].
- Protégé is a free, open-source OWL ontology editor, n.d. *Protégé*. [online] Available at: <<https://protege.stanford.edu>> [Accessed 15 February 2026].
- Ткachenko, К., 2022. Using Ontological Modeling by Intellectualization of Learning Processes. *Digital Platform: Information Technologies in Sociocultural Sphere*, [e-journal] 5 (2), pp.261-269. <https://doi.org/10.31866/2617-796X.5.2.2022.270130>
- What is Mersenne Twister?, n.d. *Educative*. [online] Available at: <<https://www.educative.io/answers/what-is-mersenne-twister>> [Accessed 21 February 2026].

REFERENCES

- A flexible and secure foundation for teaching and learning, n.d. *Google*. [online] Available at: <<https://edu.google.com/workspace-for-education/editions/education-fundamentals/>> [Accessed 23 February 2026].
- Bolotova, V., 2010. Ontologii i podannia znan [Ontologies and knowledge representation]. *Portal Znan*. [online] Available at: <<http://www.znannya.org/?view=ontology-give-knowledge>> [Accessed 15 February 2026].
- C# documentation, n.d. *Microsoft*. [online] Available at: <<https://learn.microsoft.com/uk-ua/dotnet/csharp/tour-of-csharp/>> [Accessed 17 February 2026].
- Delightfully Smart Tools For Smarter Employees & Happier Customers, n.d. *ProProfs*. [online] Available at: <<https://www.proprofs.com>> [Accessed 21 February 2026].
- Dream big. Achieve more. Visual Studio 2026, n.d. *Microsoft*. [online] Available at: <<https://visualstudio.microsoft.com/>> [Accessed 17 February 2026].

- Google Dysk [Google Drive], n.d. *Khmarni tekhnolohii v osviti*. [online] Available at: <<https://sites.google.com/view/cloudinedu/google-диск>> [Accessed 20 February 2026].
- Heneratsiia vypadkovykh chisel [Random Number Generation], 2014. *Godot Docs*. [online] Available at: <https://docs.godotengine.org/uk/4.x/tutorials/math/random_number_generation.html> [Accessed 21 February 2026].
- Ilichuk, L., 2020. Elektronnyi pidruchnyk yak innovatsiinyi zasib navchannia [E-textbook as an innovative means of education]. *Educational Horizons*, [online] 1 (50), pp.61-65. Available at: <<https://lib-old.pnu.edu.ua/bitstream/123456789/11741/1/4076-Tekst%20statti-8645-1-10-20200729.pdf>> [Accessed 23 February 2026].
- Kudin, A., Kudina, T. and Karkhut, V., 2021. Tekhnolohiia kompiuternoho testuvannia pry zmishanomu navchanni v universyteti [Technology of computer testing in physics for blended learning in university]. *Physical and Mathematical Education*, [e-journal] 29 (3), pp.93-98. <https://doi.org/10.31110/2413-1571-2021-029-3-014>
- Microsoft SQL Server 2025, n.d. *Microsoft*. [online] Available at: <<https://www.microsoft.com/en-gb/sql-server>> [Accessed 17 February 2026].
- Myliutina, I.M., 2011. Testuvannia yak efektyvnyi metod perevirky profesiinoi kompetentnosti studentiv [Testing as an effective method of testing students' professional competence]. *Urok. OSVITA.UA*, [online] March 1. Available at: <http://osvita.ua/school/lessons_summary/edu_technology/15024/> [Accessed 25 February 2026].
- Orhanizatsiia baz danykh ta znan [Database and Knowledge Organization], 2015. *SumHU*. [online] Available at: <https://elearning.sumdu.edu.ua/free_content/lectured:89b3d175c06a6b137e410cb14821d0e94549ad5a/20151030211833/44197/index.html> [Accessed 17 February 2026].
- Protégé is a free, open-source OWL ontology editor, n.d. *Protégé*. [online] Available at: <<https://protege.stanford.edu>> [Accessed 15 February 2026].
- Shcho take kombinovanyi liniinyi konhruentnyi metod heneratsii vypadkovykh chysel? [What is a combined linear congruent method of generating random numbers?], 2025. *Plany vystavok. Pokrokovna dumka*. [online] Available at: <<https://night.kozak.cx.ua/ukraincyam/shho-take-kombinovaniy-liniyniy-kongruentniy-metod-generacii-vipadkovikh-chisel.html>> [Accessed 21 February 2026].
- Sim platform dlia stvorennia testiv [Seven platforms for creating tests], 2015. *Osvita.ua*, [online] February 5. Available at: <<https://osvita.ua/school/method/technol/45747/>> [Accessed 25 February 2026].
- Tkachenko, K., 2022. Using Ontological Modeling by Intellectualization of Learning Processes. *Digital Platform: Information Technologies in Sociocultural Sphere*, [e-journal] 5 (2), pp.261-269. <https://doi.org/10.31866/2617-796X.5.2.2022.270130>
- What is Mersenne Twister?, n.d. *Educative*. [online] Available at: <<https://www.educative.io/answers/what-is-mersenne-twister>> [Accessed 21 February 2026].
- Yak vy khochete zaimatysia? [How do you want to study?], n.d. *Quizlet*. [online] Available at: <<https://quizlet.com/ua>> [Accessed 20 February 2026].

UDC 004.415.538:37.091.212-048.24]:111.1

Tkachenko Olha,

*Phd in Physical and Mathematical Sciences, Associate Professor,
Associate Professor at the Department
of Information Technologies Educational
and Scientific Institute of Management,
Technology and Legal Sciences,
National Transport University,
Kyiv, Ukraine
oitkachen@gmail.com
<https://orcid.org/0000-0003-1800-618X>*

Muzychenko Nadiya,

*PhD Student at the Department of Information Technologies,
Educational and Scientific Institute of Management,
Technology and Legal Sciences,
National Transport University,
Kyiv, Ukraine
nadya.muzychencko@gmail.com
<https://orcid.org/0009-0009-1067-7254>*

WEB-ORIENTED STUDENT KNOWLEDGE TESTING SYSTEM: ONTOLOGICAL APPROACH

Nowadays, the scope of application of educational platforms is expanding. Therefore, the study and analysis of existing software solutions for ensuring learning processes (in particular, the process of testing students' knowledge) are relevant and require their solution for the formation of an approach (in particular, ontological) to the development of modern web-oriented systems that ensure the informatization and intellectualisation of the educational process as a whole and the testing process in particular.

The purpose of this research is to study and analyse modern software tools (systems, platforms) designed to ensure learning processes, in particular testing, the formation of an appropriate ontological approach to the intellectualisation of these processes and the development of an appropriate system for testing students' knowledge on its basis.

The research methodology combines methods of a systems approach and comparative analysis of the main software solutions of the subject area (system for testing students' knowledge). This article considers an ontological approach to the development of a web-oriented system for testing students' knowledge.

The scientific novelty of this research is the analysis of modern methods and software solutions (systems, platforms) for teaching and testing students' knowledge, the formation of an ontological approach to the intellectualisation of the testing process and the development of an appropriate web-oriented system.

Conclusions. The conducted research aimed at increasing the efficiency of the testing process, the analysis of modern educational platforms and the offered ontological approach will be used as a basis for the development of an appropriate web-oriented system for testing students' knowledge. The work: analysed the specified subject area; analysed modern

educational platforms; offered an ontological approach to testing students' knowledge and its software implementation; analysed the main methods for generating test question numbers provided to students to test their knowledge; developed a web-oriented system for testing students' knowledge (which contributes, in particular, to the automation of the process of testing students' knowledge; reducing the level of workload on teachers by generating test tasks; increasing the quality of student learning; supporting a simple and comfortable user interface and resolving possible conflict situations).

Keywords: educational process; testing students' knowledge; web-oriented testing students' knowledge; tests' database; answers' database; ontology; ontological modelling of the subject area.

Надійшла 10.02.2026

Прийнята 18.03.2026

Стаття була вперше опублікована онлайн 29.05.2026



This is an open access journal, and all published articles are licensed under a Creative Commons Attribution 4.0.