

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ГНУЧКОГО ТЕСТУВАННЯ РІВНЯ ЗНАТЬ

В статті досліджено проблему підвищення ефективності тестування рівня отриманих знань із використанням комп'ютерних засобів. Розглянуто сучасні зразки програмного забезпечення для тестування. Запропоновано інформаційну технологію гнучкого тестування рівня знань, що формує репрезентативні набори тестових завдань та адаптивно обирає тестові завдання в процесі тестування. Представлено тестові програмні продукти, що реалізують запропоновану інформаційну технологію гнучкого тестування й дозволяють підтвердити її високу практичну та наукову цінність.

In the article the problem of increasing the efficiency of testing of the level of obtained knowledge with the use of computers. The modern software samples for testing are considered. Information technology of dynamic testing of knowledge level, that forms the representative set of test exercises and adaptively chooses the test exercises in the test are considered. Presented test software products, that implement the proposed information technology of dynamic testing and confirm its high scientific and practical value.

Вивчаючи певний предмет чи предметну область, людина отримує нові знання та навички [1]. Знання є упевненим розумінням предмету, вміння самостійно поводитися з ним, розбиратися в ньому, а також використовувати для досягнення поставленої мети. Здобувши певні знання, виникає необхідність в визначенні їхнього рівня та компетентності людини в певній сфері [2].

Існує ряд методів для визначення рівня знань:

- попередній контроль;
- біжучий контроль;
- усна перевірка знань;
- письмова перевірка;
- практична перевірка;
- тестова перевірка.

Останнім часом все більшої популярності набуває тестова перевірка знань [3]. Вона дає змогу об'єктивно оцінити рівень обізнаності особи в конкретній галузі. Тестова перевірка включає в себе набір тестових завдань різної складності, що робить результат тестування більш об'єктивним [4]. За допомогою ретельного конструювання тесту можна забезпечити відповідний рівень дискримінативності, а це саме те, в чому тести виграють у порівнянні з іншими формами визначення рівня знань.

З розвитком комп'ютерних технологій і їх впровадженням в навчальний процес, з'явився великий вибір систем для тестування рівня знань. Умовно всі системи можна поділити на онлайн-системи, та системи – додатки, що встановлюються на комп'ютер користувача. Перевагою онлайн-систем є можливість працювати з ними з різних машин і централізовано проводити оновлення програмних алгоритмів. Однак для роботи необхідний постійний, стабільний доступ в Інтернет. Системи тестування, що є окремими додатками, дозволяють працювати без доступу в Інтернет, а це дозволяє забезпечити стабільнішу роботу програми й ставити менше вимог для її роботи. Ці системи складніші в обслуговуванні, так як оновлення програмного продукту повинне бути здійснене для кожного комп'ютера на який така програма встановлена.

Переваги комп'ютерного тестування знань, призвели до появи великої кількості подібних систем. Так, наприклад, Харківським національним університетом радіоелектроніки була розроблена система комп'ютерного тестування знань студентів під назвою OpenTEST [5]. Ця комп'ютерна система тестування знань створена для контролю за якістю засвоєння теоретичного матеріалу, здобутих знань та практичних навиків осіб, що проходять навчання в великих організаціях. Особливістю цієї системи є її спрямованість на забезпечення процесу тестування з максимально суворою звітністю (рис. 1).



Рисунок 1 – Приклад проходження тесту у системі OpenTEST

Особливістю системи тестування знань OpenTEST є її низькі вимоги до апаратного та програмного забезпечення. Дана система дозволяє проводити тестування рівня знань, містить ряд інструментів для керування користувачами, тестами і процесом тестування. Основною особливістю процесу тестування в OpenTEST є нелінійне проходження тесту з можливістю недискретного переходу між запитаннями. Дана система не має функціоналу для показу правильних відповідей, по завершенню тестування, та відображенню підказок. Для проходження тестування потрібно знаходитися в аудиторії проведення тестування, а процес тестування може бути розпочатий тільки після підтвердження його викладачем з іншого комп'ютера.

Іншою відомою системою, що широко використовується для тестування знань, є Test-W2 [6]. Дана система призначена для контролю за рівнем знань учнів, перевірки його відповідності до вимог навчальних програм, та визначення рівня успіхів у навчанні учнів з різних предметів за допомогою комп'ютера. Система Test-W2 підходить для контролю знань, у випадку, коли опитування може бути сформульовано у вигляді коротких запитань із зазначенням 2-5 варіантів відповідей (рис. 2). Кожен тест у системі Test-W2 має ряд індивідуальних параметрів, таких як: час тестування, кількість запитань, шкала оцінювання тощо. В даній системі передбачена можливість проходження тестування з діагностикою, в процесі такого тестування будуть відображатися правильні відповіді.

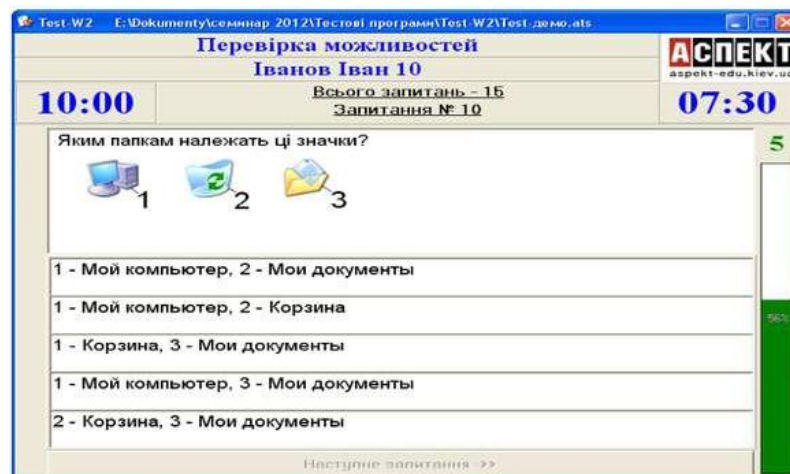


Рисунок 2 – Приклад проходження тесту у системі Test-W2

Оцінка, у системі Test-W2, виставляється відповідно до вибраної шкали з урахуванням відсотку правильних відповідей. Генерація тестів відбувається випадковим чином. Такий спосіб генерації забезпечує різну послідовність запитань на сусідніх комп'ютерах. По закінченню тестування, викладач може переглянути протокол результатів проведених тестувань.

Прикладом онлайн-системи є Moodle – безкоштовна, відкрита система управління навчанням [7]. Вона реалізує взаємодію між викладачами та учнями через мережу Інтернет. Система надає викладачам інструменти для розробки курсів, можливість створення тестів різних типів, швидке та зручне їхнє редагування, автоматичне форсування тестів, перегляд результатів тестування та обрахунок статистичних даних по проведеному тестуванню. При генерації тесту викладач може вказувати необхідну кількість завдань, час проходження тесту, кількість спроб які може здійснити учень, час початку та кінця проведення тестування. Використовуючи дану систему, студенти можуть проходити тестування (рис. 3) та переглядати результати своїх тестів, завантажувати файли з виконаними завданнями.

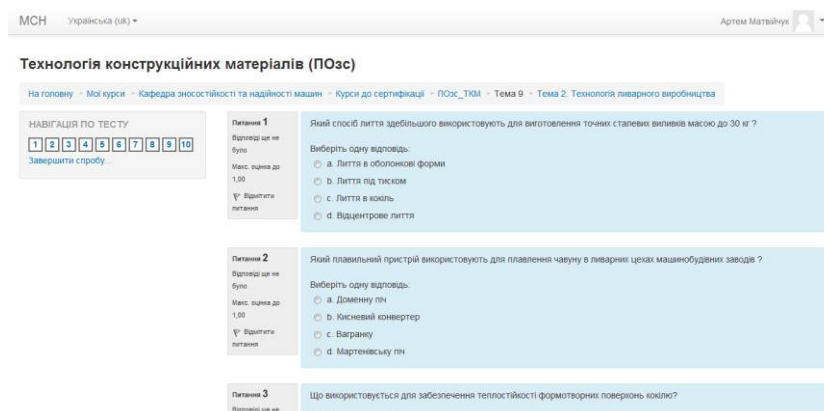


Рисунок 3 – Приклад проходження тесту у середовищі Moodle

Система Moodle потребує постійного і стабільного підключення до Інтернету, цей фактор не дає змоги використовувати дану систему в містах, де є проблеми з підключенням до Інтернету. Вона працює як серверна система і для забезпечення можливості одночасної роботи великої кількості користувачів потрібно мати потужний сервер, на якому вона буде запущена. Попри це, в класі онлайн-систем Moodle є достатньо ефективним та функціональним зразком програмного забезпечення. Недоліком її тестової складової є негнучкий метод випадкової генерації тестів, що при підборі запитання не враховує його складності та результат відповіді на попередні запитання. Така генерація тесту призводить до того, що різні учні отримують різний за сукупною складністю набір запитань, що ставить їх в нерівні умови і зменшує точність процесу тестування.

Таким чином, основною проблемою сучасних систем тестування є негнучкий алгоритм вибору запитань для тестування. Більшість систем обирає запитання випадковим чином, що не дає змоги охопити весь матеріал. Також системи не враховують результати відповідей на попередні запитання. Для вирішення цих проблем була спроектована система гнучкого тестування.

Метою даної роботи є розробка інформаційної технології гнучкого тестування рівня отриманих знань.

Оскільки система Moodle є прийнятним і широкоживим зразком онлайн-системи, доречним є удосконалення її можливостей тестування в напрямку гнучкості й адаптивності до ходу тестування, а не розробка альтернативної системи тестування з перспективним алгоритмом.

Окрім гнучкості самого процесу тестування, важливим аспектом коректної перевірки знань є репрезентативність самих завдань у тесті, база яких має бути структурно відповідною основним поняттям, що висвітлені у навчальних матеріалах.

У зв'язку з вищенаведеним, було розроблено структуру інформаційної системи (рис. 4), призначеної для реалізації гнучкого методу тестування.

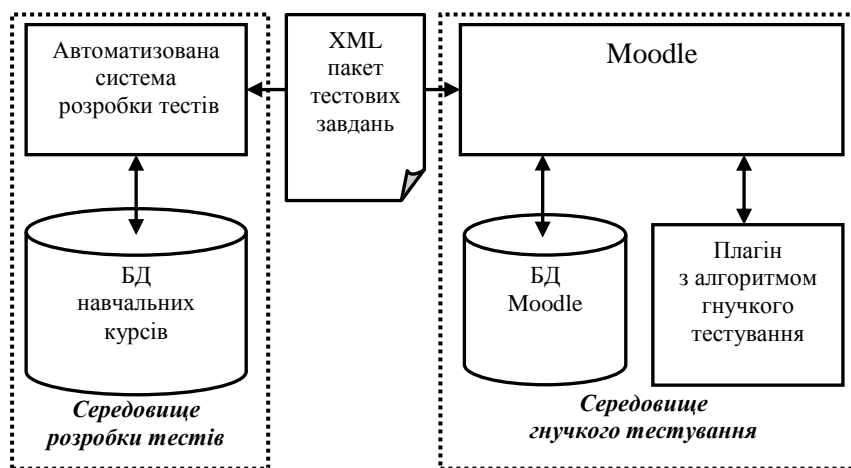


Рисунок 4 – Структура автоматизованої системи для гнучкого тестування

Автоматизована система для гнучкого тестування складається з двох основних складових, що відрізняються за типом роботи та метою застосування – середовища розробки тестів й середовища гнучкого тестування. Проміжною ланкою тут виступає пакет тестових завдань у вигляді файлу *.xml. Цей файл формується за стандартами Moodle так, щоб він міг бути завантажений в це середовище стандартними засобами Moodle. Після завантаження ці запитання зберігаються в базі даних Moodle і можуть бути використані будь-якою системою тестування в рамках цього середовища.

Середовище розробки тестів складається з автоматизованої системи розробки тестів та бази даних навчальних курсів. *Автоматизована система розробки тестів* призначена для допомоги розробникові тестів та контролю за тим, щоб тестові завдання були збалансовані за типом запитання, рівнем складності та рівномірно покривали предметну область, знання з якої належить перевірити. Для цього система розробки тестів за допомогою алгоритму аналізу текстів на базі дисперсної оцінки [8] визначає ключові терміни в заданому навчальному матеріалі та пропонує розробнику створювати запитання відповідно до цих термінів.

База даних навчальних курсів призначена для зберігання навчальних матеріалів, переліків термінів до них та тестових завдань з усіма властивостями, розробка яких проводиться. Пропонується представлена на рисунку 5 даталогічна модель бази даних навчальних курсів.

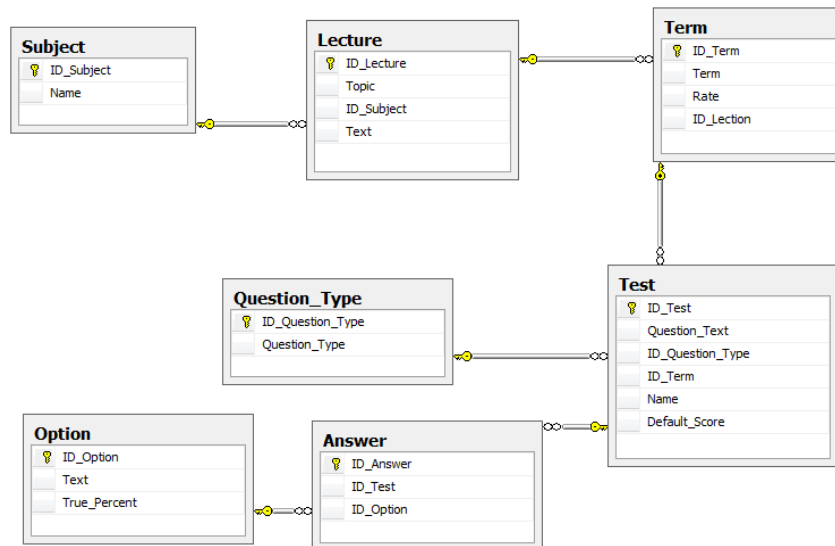


Рисунок 5 – Структура бази даних середовища розробки тестів

Наведена структура бази даних середовища розробки тестів призначена не тільки для збереження вхідних даних, а й даних проміжних етапів роботи. Так, таблиця «Subject» призначена для об'єднання групи лекцій за предметом, до складу якого вони входять, в одну структуру. Інформація про тему лекції та її контент зберігаються у таблиці «Lecture». Кожна з лекцій містить терміни, які студент повинен засвоїти, й розуміння яких відповідно слід перевірити тестом. Для збереження переліку цих термінів, їх відповідності певній лекції та рівня складності (що відповідає важливості терміна в рамках лекції) використовується таблиця «Term». В процесі роботи з даним модулем користувач створюватиме тестові завдання, параметри та умови використання яких зберігаються у таблиці «Test». Вона також містить інформацію про тип запитання, відповіді, термін що перевіряється, назву завдання, нумерацію відповідей. Текст відповіді зберігається у таблиці «Option» із зазначенням відсотку правильності кожного варіанта. Поєднувати декілька варіантів відповідей з тестовим завданням призначена таблиця «Answer».

Середовище гнучкого тестування включає в себе безпосередньо платформу Moodle й відповідну базу даних, а також пагін, що реалізує алгоритм гнучкого тестування.

Система Moodle є безкоштовною системою з відкритим програмним кодом, і цей фактор дозволяє не тільки використовувати її безкоштовно у будь-яких закладах, а й дописувати модулі для розширення її функціоналу. Платформа Moodle надає можливість для створення окремих програмних блоків – плагінів, які інтегруються в систему та додають нові можливості [9]. Існує понад відомих 40 різновидів плагінів, що додають їй різний функціонал. Так, існують плагіни, що формують різного типу звіти, створюють нові політики безпеки, формують статистику, створюють тести, тощо.

Основною задачею Moodle є організація навчального процесу через мережу Інтернет, і ключову роль в цьому процесі відіграє спосіб визначення рівня здобутих знань. Для цього в системі Moodle є вбудований модуль, що дозволяє формувати тестові запитання та проводити тестування. Власний алгоритм Moodle, що проводить тестування, не є оптимальним, адже він проводить вибірку запитань випадковим чином. В результаті такого тестування не можна отримати повної картини про рівень та глибину знань учня, що проходив дане тестування. Алгоритм вибору запитань випадковим чином є досить примітивним і не забезпечує об'єктивність перевірки. Використовуючи його, не можливо гарантувати, що у тестовій вибірці будуть обрані запитання різної складності та з різних лекційних блоків.

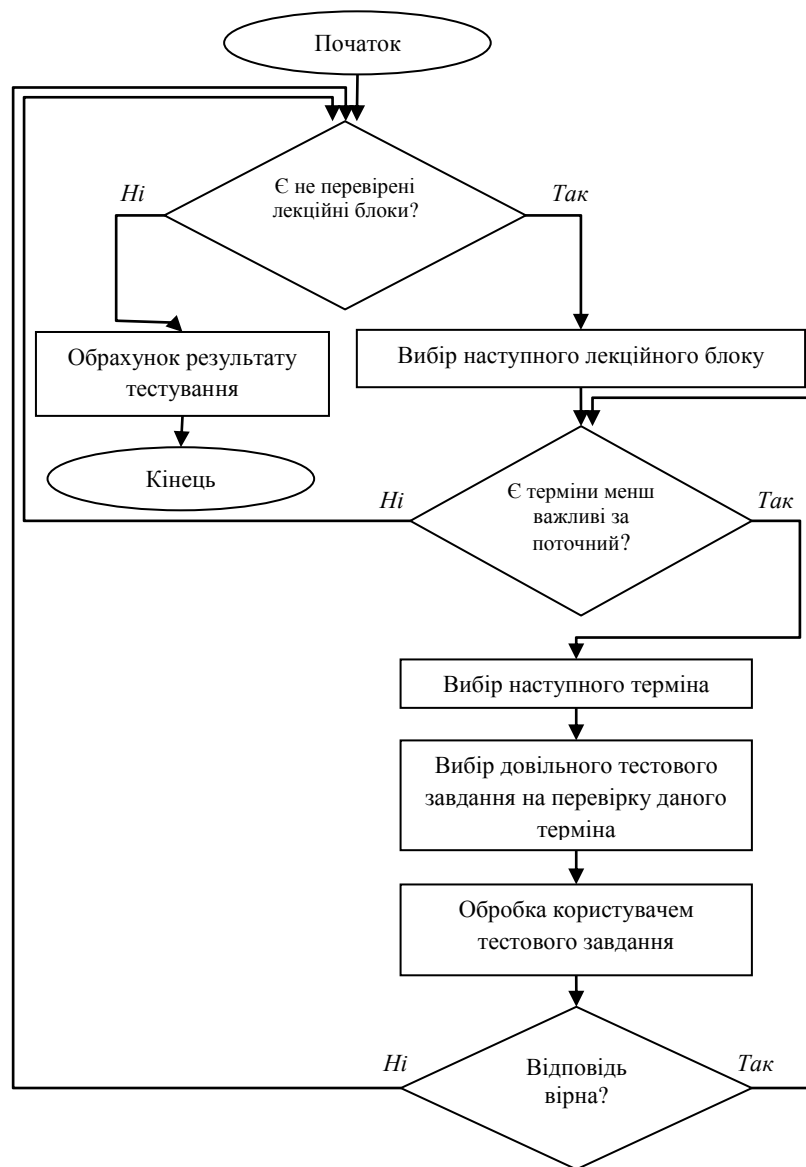


Рисунок 6 – Алгоритм гнучкого вибору тестових запитань

Для вирішення поставленої задачі, було розроблено алгоритм (рис. 6), ключовими особливостями якого є наступні:

- Охоплення максимальної кількості лекційних блоків;
- Нисхідний аналіз, що починається з перевірки розуміння найбільш семантично вагомих термінів, й поступово переходить до менш важливих термінів;
- Мінімізація затраченого на процес тестування часу при збереженні об'єктивного оцінювання рівня знань.

Наведений алгоритм гнучкого вибору тестових запитань дозволяє проводити тестування по максимальній кількості лекційних блоків, розпочинаючи тестування від найбільш базового терміну. Складність запитань збільшується у разі правильної відповіді. Якщо ж користувач не спроможний дати вірну відповідь навіть на найлегше запитання, то не має сенсу продовжувати опитування блоку та перевіряти знання на складніших запитаннях. Тому повна перевірка знань поточного лекційного блоку закінчується достроково у разі невірних відповідей або ж після правильних відповідей на усі запитання, включаючи найскладніші. У разі завершення тестування по одному блоку, відбувається перехід до тестування наступного, якщо такий є. Тест визнається пройденим у випадку, коли оброблені усі лекційні блоки. Оцінка за тест визначається як оцінка за кожен лекційний блок, помножена на ваговий коефіцієнт цього блоку, що визначається розробником навчального курсу.

З метою перевірки ефективності запропонованої технології, її складові були реалізовані у вигляді тестових програмних продуктів.

Для створення тестових завдань, що будуть покривати весь лекційний матеріал, була розроблена програма – «Question Creator». Вона визначає ключові терміни в заданому лекційному матеріалі та пропонує користувачеві створити тестові завдання відповідно до цих термінів (рис. 7). Створені запитання зберігаються в файлі з розширенням *.xml та можуть бути завантажені з нього в систему Moodle.

Для вирішення проблеми з охопленням лекційного матеріалу потрібно, щоб тестові завдання були сформовані до усіх ключових елементів лекції. Для досягнення цієї цілі було використано алгоритм для знаходження ключових термінів лекції за допомогою алгоритму аналізу текстів на базі дисперсної оцінки. Більш важливі терміни мають більше значення дисперсної оцінки. Кожне запитання, що створюється, співставлене одному із знайдених термінів та зберігає даний зв'язок. Таким чином, вдається не лише повністю покрити лекційний матеріал запитаннями, а й визначити рівень складності кожного запитання відповідно до значення дисперсної оцінки терміна, що ним перевіряється.

Розроблені набори запитань можуть бути збережені в *.xml файлі. Цей файл формується за стандартами Moodle так, щоб він міг бути завантажений в це середовище. Після завантаження ці запитання зберігаються в базі даних Moodle і можуть бути використані для тестування. При збереженні запитання маркуються програмою «Question Creator» таким чином, щоб зберегти взаємозв'язок між запитанням та складністю терміна, який воно перевіряє. В подальшому розроблений плагін Moodle використовує це маркування для визначення складності запитання та вибору наступних запитань.

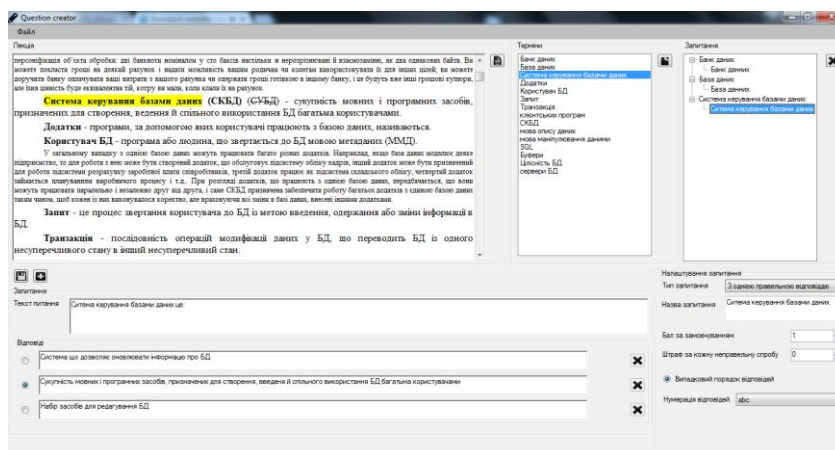


Рисунок 7 – Приклад створення тестових завдань у додатку «Question Creator»

Наведений вище алгоритм гнучкого тестування, який базується на аналізі попередніх відповідей користувача, було інтегровано в середовище Moodle шляхом розробки відповідного плагіна. Для оптимізації алгоритму тестування в Moodle передбачена можливість створення плагінів, що проводять процес тестування та аналіз відповідей, реалізуючи нові ідеї та алгоритми. Так як вся система написана за допомогою технології PHP, то і написання програмних модулів до неї теж здійснюється з використання цієї технології [9]. Для створення різних типів плагінів потрібно обов'язково наслідувати і перевизначити існуючі класи та методи Moodle, це дає можливість системі взаємодіяти та інтегруватися з значною кількістю нових плагінів.

Підтримка додавання плагінів дозволяє розширювати можливості Moodle в залежності від потреб, а також дозволяє писати плагіни не лише розробника даної системи, а й людям, що мають бажання чи потребу в додаванні нового функціоналу.

Таким чином, було розроблено плагін, що реалізує гнучке тестування в системі Moodle. Гнучкість системі забезпечує алгоритм вибору запитань. Такий алгоритм дозволяє більш гнучко охопити весь лекційний матеріал, завчасно припинити тестування у разі якщо користувач не дав правильної відповіді на найлегші запитання, а також максимально точно визначити рівень знань людини що проходить тестування.

Даний плагін використовує маркування тестових завдань для визначення складності запитання та вибору наступного запитання необхідної складності. Плагін задає запитання послідовно (рис. 8) для того, щоб аналізувати відповідь на попереднє запитання та в залежності від її результату обирати наступне. Відтак, користувач не знає, на яку кількість запитань йому ще доведеться дати відповідь.

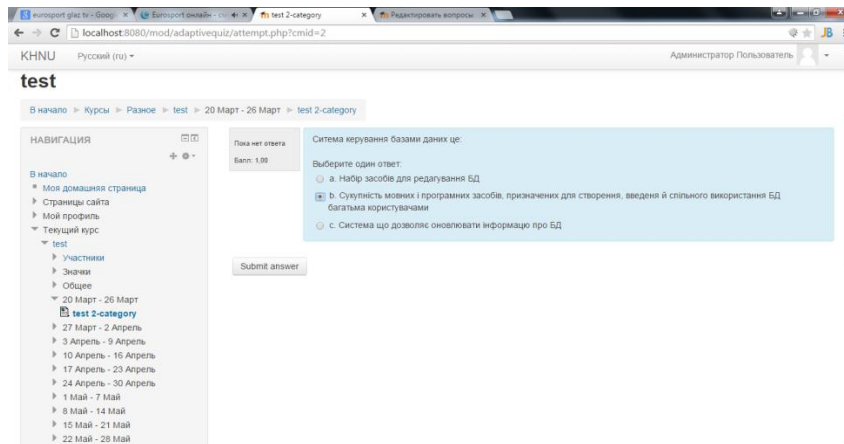


Рисунок 8 – Приклад тестування у Moodle з використанням розробленого плагіна

Для встановлення плагіна в Moodle потрібно заархівувати усі файли з програмним кодом в *.zip архів, та за допомогою панелі адміністрування завантажити цей архів у систему. Moodle розархівовує файли в папку, що відповідає типу даного модуля, і в подальшому працює з ними.

Отже, в результаті роботи було розроблено два програмних продукти, які дозволяють створювати тестові завдання та реалізують алгоритм гнучкого тестування. Розроблена система є більш гнучкою за існуючі аналоги, оскільки обирає запитання не випадковим чином, а аналізуючи їхню складність та відповіді користувача на попередні запитання.

Також на об'єктивність результатів впливає правильність сформованих запитань. Для створення коректного тестового набору, тестові завдання формуються, щоб вони рівномірно покривали навчальний матеріал.

Запропонований підхід дозволяє більш точно визначити глибину знань та виявити прогалини в розумінні вивченого матеріалу.

В статті було досліджена проблема підвищення ефективності тестування рівня отриманих знань із використанням комп'ютерних засобів. Розглянуто сучасні зразки програмного забезпечення для тестування. Запропоновано інформаційну технологію гнучкого тестування рівня знань, що формує репрезентативні набори тестових завдань та адаптивно обирає тестові завдання в процесі тестування. Представлено тестові програмні продукти, що реалізують запропоновану інформаційну технологію гнучкого тестування й дозволяють підтвердити її високу практичну та наукову цінність.

Література

1. Снитюк В.Е., Юрченко К.Н. Интеллектуальное управление оценением знаний. – Черкассы, 2013. – 262с.
2. Знання – вільна енциклопедія [Електронний ресурс]. – Режим доступу: <http://uk.wikipedia.org/wiki/Знання>.
3. Нові інформаційні технології в освіті [Електронний ресурс]. – Режим доступу: <http://it-tehnolog.com/statti/novi-informatsiyni-tehnologiyi-navchannya/>.
4. Аванесов В.С. Композиция тестовых заданий. – М., Центр тестирования, 2002.
5. Комп'ютерна програма тестування знань OpenTEST 2 [Електронний ресурс]. – Режим доступу: <http://opentest.com.ua/kompyuternaya-programma-testirovaniya-znanij-opentest-2/>
6. Контрольно-діагностична система Test-W2 [Електронний ресурс]. – Режим доступу: http://teach-inf.at.ua/load/programi/testi/test_w2_kontrolno_diagnostichna_sistema/16-1-0-7
7. Moodle – вільна енциклопедія [Електронний ресурс]. – Режим доступу: <http://uk.wikipedia.org/wiki/Moodle>
8. Ландэ Д.В., Снарский А.А. Компактифицированный горизонтальный граф видимости для сети слов // Труды Международной научной конференции «Интеллектуальный анализ информации ИАИ-2013. Знания и рассуждения» – КПИ, Киев: 2013. – с. 158-164.
9. Особливості розробки плагінів до Moodle [Електронний ресурс]. – Режим доступу: https://docs.moodle.org/dev/Main_Page