

Калашикіова Тат'яна Юрійвна – кандидат технічних наук, Український державний університет залізничного транспорту, доцент кафедри «Управління експлуатаційною роботою»; майдан Фейєрбаха, 7, м. Харків, Україна, 61050; тел.: (066) 441-50-42; e-mail: bulavina_ty@ukr.net.

Калашикіова Тат'яна Юрьевна – кандидат технических наук, Украинский государственный университет железнодорожного транспорта, доцент кафедры «Управление эксплуатационной работой», площадь Фейєрбаха, 7, г. Харьков, 61050; тел.: (066) 441-50-42; e-mail: bulavina_ty@ukr.net.

Kalashnikova Tetyana – candidate of technical sciences, associate professor, Ukrainian State University of Railway Transport, department of "Management of operational work"; Kharkiv, Feuerbach sq, 7, 61050; tel.: (066) 441-50-42; e-mail: bulavina_ty@ukr.net.

Цвір Павло Александрович – магістр, Український державний університет залізничного транспорту, кафедра «Управління експлуатаційною роботою»; майдан Фейєрбаха, 7, м. Харків, Україна, 61050; тел.: (063) 155-48-15; e-mail: pavel.tsvir@ukr.net.

Цвир Павел Александрович – магистр, Украинский государственный университет железнодорожного транспорта, кафедра «Управление эксплуатационной работой»; площадь Фейєрбаха, 7, г. Харьков, 61050; тел.: (063) 155-48-15; e-mail: pavel.tsvir@ukr.net.

Cvir Pavlo – listener, Ukrainian State University of Railway Transport, department of "Management of operational work"; Kharkiv, Feuerbach sq, 7, 61050; tel. (093) 982-71-34; e-mail: pavel.tsvir@mail.ru.

Приступа Сергій Петрович – магістр, Український державний університет залізничного транспорту, кафедра «Управління експлуатаційною роботою»; майдан Фейєрбаха, 7, м. Харків, Україна, 61050; тел.: (098) 968-02-43; e-mail: prystupa_serhiy@ukr.net.

Приступа Сергей Петрович – магистр, Украинский государственный университет железнодорожного транспорта, кафедра «Управление эксплуатационной работой»; площадь Фейєрбаха, 7, г. Харьков, 61050; тел.: (098) 968-02-43; e-mail: prystupa_serhiy@ukr.net.

Pristupa Sergey – listener, Ukrainian State University of Railway Transport, department of "Management of operational work"; Kharkiv, Feuerbach sq, 7, 61050; tel. (093) 982-71-34; e-mail: prystupa_serhiy@ukr.net.

Полозун Роман Сергійович – магістр, Український державний університет залізничного транспорту, кафедра «Управління експлуатаційною роботою»; майдан Фейєрбаха, 7, м. Харків, Україна, 61050; тел.: (099) 286-83-45, (099)-286-83-45; e-mail: milano.roman01@gmail.com.

Полозун Роман Сергеевич – магистр, Украинский государственный университет железнодорожного транспорта, кафедра «Управление эксплуатационной работой»; площадь Фейєрбаха, 7, г. Харьков, 61050;

Polozun Roman – listener, Ukrainian State University of Railway Transport, department of "Management of operational work"; Kharkiv, Feuerbach sq, 7, 61050; tel. (093)982-71-34, (099)-286-83-45;

УДК 004.4:378.147

М. С. ШИРОКОПЕТЛЄВА, Ю. Ю. ЧЕРЕПАНОВА

ПРО ПІДХІД ДО ПРОЕКТУВАННЯ СИСТЕМИ ТЕСТУВАННЯ ЗНАТЬ З МОВИ SQL

Розглядаються питання проектування та розробки системи тестування знань з мови структурованих запитів до бази даних. Запропоновано критерії формування тестових наборів для тестування знань з мови структурованих запитів, наведена методика формування тестових завдань та оцінки проходження тестування. Визначено підходи до проектування програмної системи тестування знань з урахуванням можливості виконання запитів на модифікацію даних та зміну структури бази даних, з відкатом транзакції при помилці запиту редагування даних.

Ключевые слова: автоматизована перевірка, система тестування знань, мова структурованих запитів.

Рассматриваются вопросы проектирования и разработки системы тестирования знаний языка структурированных запросов к базе данных. Предложены критерии формирования тестовых наборов для тестирования знаний языка структурированных запросов, приведена методика формирования тестовых заданий и оценки прохождения тестирования. Определены подходы к проектированию программной системы тестирования знаний с учетом возможности выполнения запросов на модификацию данных и изменение структуры базы данных, с откатом транзакции при ошибке запроса редактирования данных.

Ключевые слова: автоматизированная проверка, система тестирования знаний, язык структурированных запросов.

The problems of the design and development of knowledge testing system Structured Query Language was discussed in the article. Test systems are widely applied in the educational process. But evaluation of the level of knowledge of SQL has problems with their use. There are usage isolated transaction for CRUD operations and database recovering for database modification query (create, alter, drop). The method of test tasks formation was described. Also two criteria for the test tasks were proposed and formalized: criterion completeness and lack of intersection criterion. There are set in the test questions should cover all elements of the topic you being tested (criterion completeness). The lack of intersection criterion is checking to be only fragments of the tasks for which formulated questions. Difficulty levels for thematic units can be used for formation of the test results.

The authors propose an approach to web system development using ASP .NET MVC platform. Web system supports three roles (teacher, student and administrator). System contains two databases (training DB and a test DB for recovery training database and execute queries). This approach allows to automate the testing of knowledge on SQL.

Keywords: automated testing, system testing, test task criteria, Structured Query Language

Вступ. Інформаційні технології - вельми широке визначення, під яке потрапляє ряд окремих технічних засобів і прийомів роботи з інформацією. Дедалі більшої популярності набувають веб-системи. Такі системи мають певні переваги, а саме: доступність за рахунок мережі Internet, швидкість розповсюдження, менші вимоги до пам'яті комп'ютерів та іншої техніки, і т.д. Для даної роботи ці переваги мають ключове значення, тому питання проектування веб-систем тестування знань набувають все більше важливості.

На сьогоднішній день підготовка фахівців в напрямку комп'ютерних наук практично неможлива без вивчення різних СУБД і написання запитів до бази даних. У більшості технічних навчальних закладів сфери ІТ вивчають ці питання, без цього у наш час неможливо розробляти більшість програмних продуктів. Тому важливими є закріплення та перевірка знань студентів у цій сфері. Система тестування знань мови запитів може використовуватись для спрощення завдань перевірки і навчання студентів. Автоматизація цього процесу дає багато переваг, до того ж у час технологій, в першу чергу Internet, така система буде більш відповідати вимогам сучасності. Різні веб-системи тестування вже застосовуються у багатьох навчальних установах, це зручний спосіб контролю та перевірки знань учнів та студентів.

Метою роботи є проектування системи тестування знань мови запитів до бази даних. Система повинна відповідати усім вимогам, окремо слід спроектувати функціонал різних груп користувачів. Також система має передбачати певну звітність результатів тестування та зручне і зрозуміле оброблення помилок з поясненнями для студентів у деяких випадках.

Аналіз літературних даних та постановка проблеми. Тестування як ефективний спосіб перевірки знань здобуває все більшу і більшу популярність в навчальних закладах вищої та середньої акредитації [1, 2]. Одною з основних і безперечних його переваг є мінімум часових витрат на отримання надійних підсумків контролю. При тестуванні використовують як паперові, так і електронні варіанти. Останні особливо привабливі, тому що дозволяють отримати результати практично відразу по завершенні тесту.

Тестування – більш справедливий метод, що ставить всіх учнів в рівні умови, як у процесі контролю, так і в процесі оцінювання, практично виключаючи суб'єктивізм викладача [3].

В дослідженнях [4] розглянуто питання формування тестового набору, який всебічно охоплює навчальний матеріал, але такий підхід не враховує особливості перевірки знань з мов програмування, зокрема, з мови запитів.

При тестуванні запитів до бази даних потрібно мати на увазі особливості запитів до різних СУБД. Також різні запити виконують різну роботу з даними. Найбільш безпечні просто зчитують і повертають деяку інформацію. Ці запити важливо перевіряти на помилки та перевіряти отримані дані з потрібними. Такі запити, як Insert, Update, Delete, Create, Drop, вже впливають на дані, змінюючи їх, тому такі запити більш небезпечні при виникненні помилок. При роботі з запитами до бази даних потрібно розділити обро-

бку помилок різних запитів і передбачити різний порядок рядків у отримуваних даних при виконанні запити.

Окрему увагу приділимо проектуванню системи тестування.

Ціль та задачі дослідження. Метою дослідження є обґрунтування методів створення тестових завдань з мови SQL та вибір архітектури веб-системи тестування знань з мови SQL.

Для досягнення поставленої мети були поставлені наступні завдання:

1. Описати методику формування тестових завдань та оцінки результатів тестування.

2. Визначити підходи до проектування програмної системи тестування знань з урахуванням можливості виконання запитів на модифікацію даних та зміну структури бази даних.

Опис методики формування тестових завдань.

Для системи тестування знань важливу роль відіграє галузь, інформацію якої підлягає перевірці. Для розділів дисциплін, що містять теоретичні знання, в першу чергу, рекомендується застосовувати оцінку рівня володіння основними поняттями та відношеннями між ними, а лише після успішного контролю цього рівня – перехід до більш абстрактних питань та питань, які можуть містити узагальнення та приклади практичної реалізації набутих теоретичних знань. При цьому для оцінки базового рівня володіння теоретичними знаннями пропонується використовувати тести з вибором однієї правильної відповіді з багатьох та багатозначні питання [5]. Таким чином можна запобігти ускладненню програмного забезпечення для проведення тестування та на достатньому рівні оцінити знання користувача. При формуванні питань підвищеної складності може бути застосовані питання з визначеннями відповідності елементів та формування коректної послідовності дій [6]. Такі варіанти питань призводять до ускладнення програмної системи тестування, але значно розширюють перелік навичок та компетенцій, які підлягають перевірці. Для дисциплін, що носять теоретичний характер, використання питань з відкритою відповіддю (заповнення таблиць, схем, довільне висловлювання, тощо) призводить до зниження рівня формалізації теоретичної інформації та необхідності введення підсистем синтаксичного та семантичного аналізу до системи перевірки знань (при автоматичній перевірці тестів), що значно перевантажує систему, або до так званої «ручної» перевірки завдань, що збільшує навантаження на викладача і та може привести до необ'єктивності оцінки знань, що неприпустимо у системах тестування.

Для галузей, які потребують перевірки набутих навичок розв'язання завдань, слід застосовувати варіанти одиночного та множинного вибору (при наявності декількох рішень задачі) з відкладеними відповідями, щоб учасники тестування мали змогу самостійно знайти рішення, а не підставляти надані у відповідях варіанти.

Окрему увагу приділимо тестуванню навичок в галузі програмування, коли необхідно не лише перевірити теоретичні знання (наприклад, знання основних управляючих конструкцій), а й навички виконання практичних завдань – написання фрагментів про-

грам. Складність полягає в тому, що такого роду завдання можуть бути виконані по-різному, тобто завдання може мати безліч вірних рішень і для перевірки коректності виконання завдання не можливо застосувати жоден з класичних підходів до тестування. На даний час існують програмні засоби - системи тестування рішень, які дозволяють автоматизувати перевірку коректності виконання рішень, обраних для тестування [7].

Для наведення методики формування тестових завдань окрім стандартних критеріїв для тестів [6] (критерій надійності, валідності, складності тесту, дискримінативності) запропонуємо критерії, яким повинні задовольняти тестові набори:

1. Критерій повноти - наведені у тестовому наборі питання повинні покривати усі елементи теми, за якою проводиться тестування.

2. Критерій відсутності перетину тестових питань: перевірки підлягають лише фрагменти завдання, за яким сформульоване питання.

Докладніше опишемо критерій повноти. Для цього наведемо терміни «зміст поняття» та «об'єм поняття»: змістом поняття є сукупність суттєвих ознак, які характеризують даний предмет, а об'єм поняття – сукупність або множина предметів, які входять до складу поняття. Провівши аналогію між по-

няттям (основним об'єктом предметної галузі) та дисципліною, за якою проводиться тестування, можна зауважити, що до змісту дисципліни входить увесь перелік тематичних одиниць (тем, розділів, підрозділів, тощо), а об'єм складає множина тестових питань, які покривають увесь зміст кожної теми. Формалізуємо наведене:

$$S = \cup R^i,$$

де S – об'єм всієї дисципліни; R^i – i -е тематична одиниця, $i \in [1; n]$, n – кількість тематичних одиниць в дисципліні.

Для запобігання «обміну думками» під час проведення тестування тест повинен містити не менш ніж 10 питань за кожною темою.

Наприклад, для перевірки знань з мови структурованих запитів, виділимо розділи: DML, DDL, DCL. В кожному з них визначимо тематичні одиниці, взаємозв'язок яких для розділу DML наведено в табл.1. Наведений рівень складності може застосовуватися для автоматизації формування тестових наборів для стійкої групи користувачів або дискримінаційного тестування (для отримання максимальної оцінки «задовільно» за успішне проходження тесту першого рівня складності).

Таблиця 1 – Тематичні одиниці підрозділу DML

Підрозділ	Тема	Тематична одиниця	Рівень
SELECT	Прості запити	Формування виведення	1
		Обмеження на дані	1
		Спеціальні оператори, упорядкування	1
		Групові операції	1
		Обмеження на групи	2
	Запити на декількох таблицях	Об'єднання таблиць	2
		Об'єднання таблиці з собою	2
		Обмеження на дані та групи з декількох таблиць	3
	Підзапити	Прості підзапити	2
		Корельовані підзапити	3
Предикати ANY, ALL		3	
INSERT	Додавання одиничного запису	1	
	Множинне додавання	2	
UPDATE	Проста модифікація даних	1	
	Модифікація даних на підставі даних з іншої таблиці	2	
DELETE	Просте вилучення даних	1	
	Вилучення даних на підставі даних з іншої таблиці	2	

Тобто для формування тесту з цього підрозділу необхідно скласти не менш ніж 170 питань відповідно до наведеної тематики.

На підставі критерію відсутності перетину тестових питань, пропонується обмежити перевірку відповідей лише тематикою питання, наприклад, при перевірці знань за тематичним розділом «Обмеження на дані» необхідно перевірити результуючу вибірку на кількість рядків та на їх відповідність, а не звертати увагу на кількість та порядок полів виведення та на упорядкування рядків в результаті запиту. Формалізуємо:

$$\forall i, j: R^i \cap R^j = \emptyset,$$

$R^i, R^j - i (j)$ -а тематична одиниця, $i, j \in [1; n]$, n – кількість тематичних одиниць в дисципліні.

Таким чином, уся тестова множина може бути представлена:

$$S = \cup Q_i,$$

де Q_i – i -те питання тестового набору, яке входить до складу $R^i, i \geq 10$.

Так, для формування тестового набору з одного тематичного розділу, слід навести однотипні питання, наприклад, для тематичного розділу «Обмеження на дані» наведемо таблицю Students (id, Name, DOB, Group, Address) та запити:

- Виведіть студентів, які навчаються у групі IT-21;
- Виведіть студентів, які мешкають у гуртожитку (адреса містить текст «Харків, вул. Перша, 1», тощо).

А для визначення коректної відповіді слід для кожного запиту сформулювати еталонний запит, за допомогою якого можна отримати результуючу вибірку. Також необхідно розробити показники оцінки результуючих запитів для кожної з тематичних одиниць і ввести їх до бази тестових питань з вказівкою тематичної одиниці, а всі завдання повинні відноситися до певної одиниці для здійснення перевірки.

Тестовий набір для підсумкового контролю знань повинен містити питання з кожної тематичної одиниці, але при досить великій кількості тем та обмежені часу пропонується проводити тестування, призначене для окремих груп користувачів, а в тестові набори включати питання одного рівня складності. Наприклад, в першу чергу можна сформулювати тест з питань рівнем складності 2 (оцінка «добре»), після складання якого можна сформулювати 2 набору з рівнем складності 1 (для перевірки володіння базовими знаннями для тих, хто не склав першу частину) та рівнем 3 – для користувачів, які можуть претендувати на відмінну оцінку.

Також можна передбачити ведення поточного контролю шляхом автоматичного формування тестових наборів на підставі питань з обраного розділу / підрозділу без урахування складності завдань. Кількість питань в завданні викладач може самостійно визначити, але ця кількість не повинна бути меншою, ніж кількість тематичних одиниць. Тобто при ієрархічній структурі організації завдань полегшується процес формування тестових наборів з урахуванням критерію валідності за змістом.

Методики оцінки результатів тестування. Для перевірки знань можуть використовуватися різні критерії формування оцінки [6]. Методи на основі кількісних критеріїв припускають використання кількісної шкали, тобто оцінка в цьому випадку задається числом. Наприклад, коефіцієнт засвоєння $K = A / P$, де A – кількість правильних відповідей в тесті, P – загальна кількість відповідей. Також при формуванні оцінки можна врахувати типи і характеристики тестових завдань, рівня підготовленості користувача, що проходить тестування, віднесення користувача до стійких класів, тощо. Усі перелічені підходи можуть застосовуватися для тестів, які не передбачають декілька правильних семантично різних варіантів відповідей. Для перевірки знань та навичок в галузі програмування слід передбачити можливість автоматизованої перевірки запропонованого рішення, тобто створення системи тестування, яка надає можливість компіляції коду (фрагменту коду) програми та перевірки результатів за темами.

Для програмної системи перевірки знань з мови SQL необхідно створити набір вимог за кожною тематичною одиницею та алгоритм порівняння результуючих вибірок (еталонної та запропонованої користувачем).

Наприклад, для теми «Обмеження даних» результати отриманих вибірок пропонується порівнювати

по рядкам (для даної тематичної одиниці) за такими показниками: кількість рядків повинна збігатися, ідентифікатори рядків (ROWID) повинні збігатися за умови вибірки з даних, частково невідомих користувачеві (для запобігання некоректного формування запиту). При виконанні цих правил завдання можна вважати виконаним правильно.

Для перевірки запитів на модифікацію слід передбачити виконання двох ізольованих транзакцій (для еталонного та запропонованого запитів) для унеможливлення довільної зміни початкових даних, а результати виконання завдання перевіряти за допомогою перевірки даних таблиць за таким критерієм: повний збіг даних таблиці при виконанні запиту на вибірку усіх даних з упорядкуванням по ключовому полю.

При проведенні поточного контролю і формуванні завдань з рівним рівнем складності для оцінювання можна застосовувати коефіцієнти, які відповідають рівням складності. Пропонується загальну оцінку розраховувати за формулою: $M = k * A / P$, де k – коефіцієнт складності. Для середнього рівня складності (2) – $k = 1$, для низького (1) – $k = 0.7$, для завдань підвищеної складності (3) – $k = 1.3$. А за наявності більшої кількості рівнів можна коефіцієнт складності задавати самостійно.

Опис проектування програмної системи тестування знань з SQL. Одним з особливостей розробки системи тестування знань мови запитів є передбачення зміни даних після кожного сеансу роботи студентів з системою. Таким чином потрібно відновлювати тестову базу даних після кожного сеансу роботи студента. Важливо також зберігати певний звіт про тестування студентів певної групи, і звіт про роботу окремих студентів цієї групи для звітності навчально-го закладу, якщо це потрібно.

Для реалізації веб-сервісу тестування знання мови запитів у вигляді веб-серверу та веб-клієнта було вирішено використовувати клієнт-серверну архітектуру. Веб-система розроблена з використанням платформи ASP.NET MVC [8]. Для реалізації серверної частини був використаний принципи проектування SOLID [9]. Серверна частина виконує функцію обробки введених клієнтом даних та надання клієнту певної інформації відповідно до його запитів. Перед виконанням запиту на стороні клієнта доцільна перевірка запиту за допомогою регулярних виразів (на наявність ключових слів).

На клієнтському боці для створення дизайну і динамічної роботи веб-сайту слід використовувати JQuery бібліотеку, що значно спрощує роботу з JavaScript. Клієнт являє собою web-додаток, створений також з використанням ASP.NET MVC та він виконує роль динамічної навігації між сторінками та взаємодії з клієнтом.

Для зберігання даних пропонується використання бази даних. В якості СУБД обрано Microsoft SQL Server, що краще всього інтегрується з ASP.NET проектами, при використанні технологій Entity Framework та LINQ та моделлю Code First [10].

Слід зауважити, що існують різні діалекти мови SQL та спеціальні функції, реалізовані у конкретних СУБД. При реалізації системи тестування слід перед-

бачити можливість вибору типу СУБД для виконання запиту, але цей підхід вимагає необхідність підключення різних СУБД для коректної перевірки запитів, що в свою чергу підвищує вимоги до апаратного забезпечення серверної частини. На даний час в реалізованій системі тестова база використовує СУБД Microsoft SQL Server, та у майбутньому планується розширити перелік СУБД.

В базі даних зберігається вся динамічна інформація, що може бути редагована адміністраторами та викладачами, а саме: дані користувачів; групи студентів; завдання; групи тестових завдань, критерії оцінки груп, питання з еталонними запитими, відповіді та оцінки.

Сервер включає основний функціонал обробки даних, відновлення тестової бази даних. Певний функціонал, та доступ до даних з бази даних винесен

до окремих бібліотек класів. Також в окрему бібліотеку класів винесені ресурси локалізації. На рисунку 1 зображена діаграма компонентів. З неї видно, що пропонується використовувати компонент серверу, базу даних та компонент «веб-клієнт». При програмній реалізації з використанням платформи .Net, компонент «база даних» використовує DbContext, що потрібен для роботи EntityFramework. Компонент «Сервер» використовує «Controller» для формування сторінок та обробки запитів від клієнта і обробки потрібних для відображення даних, а також використовує бібліотеку EntityFramework. Веб-клієнт, в свою чергу, використовує бібліотеку jQuery.

Усі компоненти взаємодіють між собою: клієнт робить запит до серверу, сервер запитує дані у серверу БД, а тоді повертає їх клієнту.

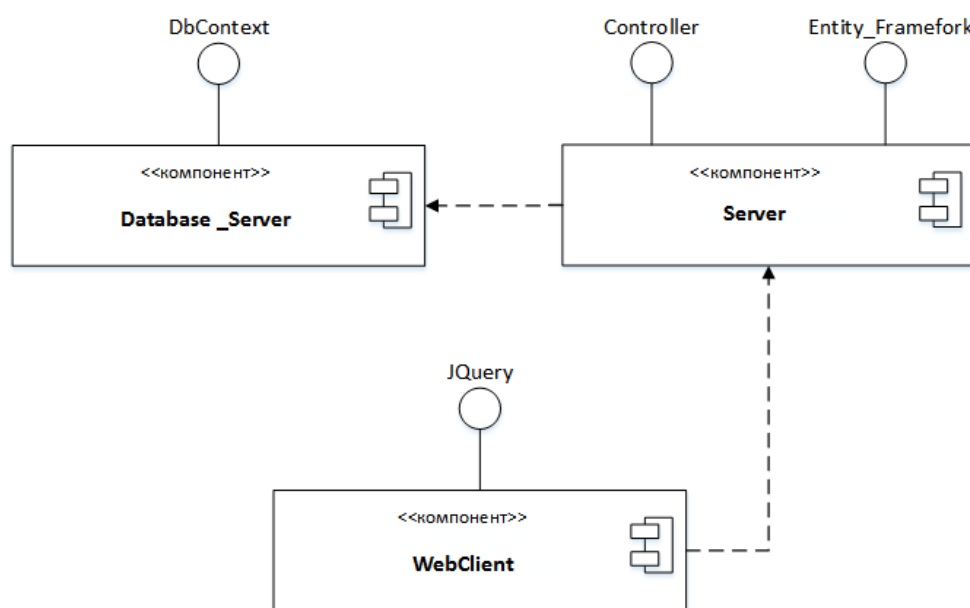


Рис. 1 – Діаграма компонентів

Висновки. В результаті проведених досліджень запропоновано критерії формування тестових наборів для тестування знань з мови структурованих запитів, наведена методика формування тестових завдань та оцінки проходження тестування. Також в роботі наведений підхід до створення програмної системи тестування знань, яка передбачає можливість відновлення, редагування тестової бази, та відкат транзакції при помилці запиту редагування даних та відображення помилки.

В подальшому розвиток системи можливий у декількох напрямках: додавання нового функціоналу з тестування видів запитів щодо забезпечення цілісності даних (indexes, constraints), додавання підтримки додаткових СУБД та тестування запитів до бази даних цих СУБД.

Список літератури:

1. Майоров, А. Н. Теория и практика создания тестов для системы образования. [Текст] / А. Н. Майоров. – М.: «Интеллектуальный центр», 2002. – 296 с.
2. Битюев, Ю. В. Тестирование в образовательном процессе: методические указания [Текст] / Ю. В. Битюев, Е. Л. Очирова. – Улан-Удэ: Изд-во БГСХА им. В. Р. Филиппова, 2009. – 32 с.

3. Різун, Н. О. Методика розробки автоматизованої системи управління якістю тестового контролю знань [Текст] / Н. О. Різун, Ю. К. Тараненко // Вісник НТУ "ХПІ". Тематичний випуск: Інформатика і моделювання. – Харків: НТУ "ХПІ". – 2010. – № 31. – С. 145–152.
4. Кабанова, Т. А. Тестирование в современном образовании: уч. пособие. [Текст] / Т. А. Кабанова, В. А. Новиков. – М.: Высшая школа, 2010. – 384 с.
5. Аванесов, В. С. Форма тестовых заданий: Учебное пособие. [Текст] / В. С. Аванесов. – М.: Центр тестирования, 2005. – 155 с.
6. Карпенко, А. П. Тестовый контроль качества обучения и критерии качества образовательных тестов. Обзор [Электронный ресурс] / А. П. Карпенко, А. С. Домников, В. В. Белоус // Наука и образование: электронное научно-техническое издание. – 2011. – № 04. – Режим доступа: <http://technomag.edu.ru/doc/184741.html>. – 15.06.2016г. – Загол. с экрана.
7. Бондарев, В. М. Мережева підтримка навчання програмуванню [Текст] / В. М. Бондарев, Ю. Ю. Черепанова // V Міжнародна науково-практична конференція "Інформаційні технології та комп'ютерна інженерія", 27–29 мая 2015 р. – Івано-Франківськ: Супрун В. П., 2015. С. 49–50.
8. Freeman, A. Pro ASP.NET MVC 5 5th ed. [Текст] / A. Freeman. – Apress, 2013. – 832 с.
9. Кинг, Б. Опасность нарушения принципов SOLID в C# [Электронный ресурс] / Microsoft Developer Network org – Режим доступа: <https://msdn.microsoft.com/ru-ru/magazine/dn683797.aspx/> – 01.06.2016 р. – Назва з екрана.

10. Code First Migrations and Deployment with the Entity Framework in an ASP.NET MVC Application [Електронний ресурс] / The ASP.NET Site. – Режим доступу: <http://www.asp.net/mvc/overview/getting-started/getting-started-with-ef-using-mvc/migrations-and-deployment-with-the-entity-framework-in-an-asp-net-mvc-application> – 10.05.2016 р. – Назва з екрану.
6. Karpenko, A. P., Domnikov, A. S., Belous, V. V. (2011). Testovyi kontrol kachestva obucheniia i kriterii kachestva obrazovatelnykh testov. Obzor. Nauka i obrazovanie: elektronnoe nauchno-tekhnicheskoe izdanie, 04. <http://technomag.edu.ru/doc/184741.html> (15.06.2016).
7. Bondarev, V. M., Cherepanova, Y. Y. (2015). Merezheva pidtrinka navchannia programuvanniu. Abstracts of Papers. Conference V Mizhnarodna nauково-praktichna konferenciia "Informatiini tekhnologii ta komp'uterna inzheneriia", Ivano-Frankivsk, May 27-29 2015, 49–50, Ivano-Frankivsk: Suprun V.P.
8. Freeman, A. (2013). Pro ASP.NET MVC 5 5th ed. Apress.
9. King, B.(2016). Opasnost narusheniia printcipov SOLID v C#. Microsoft Developer Network org. <https://msdn.microsoft.com/ru-ru/magazine/dn683797.aspx>.
10. Code First Migrations and Deployment with the Entity Framework in an ASP.NET MVC Application. The ASP.NET Site. <http://www.asp.net/mvc/overview/getting-started/getting-started-with-ef-using-mvc/migrations-and-deployment-with-the-entity-framework-in-an-asp-net-mvc-application> (10.05.2016).

Bibliography (transliterated):

1. Maiorov, A. N. (2002). Teoriia i praktika sozdaniia testov dlia sistemy obrazovaniia. Moscow: «Intellekt-tcentr».
2. Bituev, Iu. V., Ochirova E. L. (2009). Testirovanie v obrazovatel'nom protsesse. Ulan-Ude: Izd-vo BGSKhA im. V. R. Filippova.
3. Rizun, N. O., Taranenko, Iu. K. (2010). Metodika rozrobki avtomatizovanoi sistemi upravlinnia iakistiu testovogo kontroliu znan. Visnik NTU "KhPI". Tematichnii vipusk: Informatika i modeliuvannia, 31, 145–152.
4. Kabanova, T. A., Novikov, V. A. (2010). Testirovanie v sovremennom obrazovanii. Moscow: Vysshiaia shkola.
5. Avanesov, V. S. (2005). Forma testovykh zadani. Moscow: Tcentr testirovaniia.

Надійшла (received) 20.03.2016

Бібліографічні описи / Библиографические описания / Bibliographic descriptions

Про підхід до проектування системи тестування знань з мови SQL/ М. С. Широкопетлева, Ю. Ю. Черепанова // Вісник НТУ «ХПІ». Серія: Механіко-технологічні системи та комплекси. – Харків : НТУ «ХПІ», 2016. – No 17(1189). – С.50–55. – Бібліогр.: 10 назв. – ISSN 2079-5459.

О подходе к проектированию системы тестирования знаний по языку SQL/ М. С. Широкопетлева, Ю. Ю. Черепанова // Вісник НТУ «ХПІ». Серія: Механіко-технологічні системи та комплекси. – Харків : НТУ «ХПІ», 2016. – No 17(1189). – С.50–55. – Бібліогр.: 10 назв. – ISSN 2079-5459.

The approach to the design of system of testing knowledge of SQL / M. Shirokopetleva, Y. Cherepanova //Bulletin of NTU “KhPI”. Series: Mechanical-technological systems and complexes. – Kharkov: NTU “KhPI”, 2016. – No 17 (1189).– P.50–55. – Bibliogr.: 10. – ISSN 2079-5459.

Відомості про авторів / Сведения об авторах / About the Authors

Широкопетлева Марія Сергіївна – Харківський національний університет радіоелектроніки, старший викладач кафедри програмної інженерії; пр. Науки, 14, м. Харків, Україна, 61166; тел. 066-403-91-64; e-mail: mshirokopetleva@gmail.com.

Широкопетлева Марія Сергеевна – Харьковский национальный университет радиоэлектроники, старший преподаватель кафедры программной инженерии; пр. Науки, 14, г. Харьков, Украина, 61166; тел. 066-403-91-64; e-mail: mshirokopetleva@gmail.com.

Shirokopetleva Mariia – Kharkiv National University of Radio Electronics; Senior lecturer; Kharkiv, 61166, Nauka ave., 14; phone: 066-403-91-64; e-mail: mshirokopetleva@gmail.com.

Черепанова Юлія Юрійівна – Харківський національний університет радіоелектроніки, старший викладач кафедри програмної інженерії; пр. Науки, 14, м. Харків, Україна, 61166; тел. 066-161-72-42; e-mail: cher_y@list.ru.

Черепанова Юлия Юрьевна – Харьковский национальный университет радиоэлектроники, старший преподаватель кафедры программной инженерии; пр. Науки, 14, г. Харьков, Украина, 61166; тел. 066-161-72-42; e-mail: cher_y@list.ru.

Cherepanova Yulia– Kharkiv National University of Radio Electronics; Senior lecturer; Kharkiv, 61166, Nauka ave., 14; phone: 066-161-72-42; e-mail: cher_y@list.ru.