

УДК 37.046 : 372.862

В.В. Федько, О.В. Тарасов

Харківський національний економічний університет імені С. Кузнеця, Харків

CAPTIVATE-РЕАЛІЗАЦІЯ ТЕСТІВ З ДИСЦИПЛІНИ "ОРГАНІЗАЦІЯ БАЗ ДАНИХ ТА ЗНАНЬ"

Розглянуто можливості інструментальних авторських засобів Adobe Captivate для створення тестів та методичні підходи до використання їх під час вивчення дисципліни "Організація баз даних та знань". Дослідження показало, що система Adobe Captivate підтримує обидва типи тестових завдань – відкриті і закриті і множини видів кожного з цих типів. Показано, що підготовані в Adobe Captivate тести можна використовувати як складову частину проекту електронного навчального посібника так і автономно. Досліджено всі необхідні засоби системи щодо налаштування формату подання тесту та звіту після його проходження, ваги і часу виконання окремих завдань тощо. Запропоновано як методичний прийом вивчення зв'язку між поняттями "база даних" та "тести" на заняттях з дисципліни "Організація баз даних та знань", що дозволяє поглибити знання з моделювання даних та проектування інтерфейсу користувача.

Ключові слова: педагогічний тест, Adobe Captivate, база даних, інтерфейс користувача.

Вступ

Тести як форма визначення навчальних досягнень з'явилися в середині дев'ятнадцятого сторіччя в роботах Ф. Гальтона, Дж. Фішера, Р. Каттела [2, 4]. Особливого поширення вони почали набувати у середині 90-х років минулого століття із запровадженням персональних комп'ютерів у навчальний процес [5, 6].

Для забезпечення інструментальної підтримки підготовки комп'ютерних педагогічних тестів інтенсивно почали розроблятися відповідні оболонки [1, 7 – 10]. Проблеми вибору оболонки та її застосування в навчальному процесі залишаються актуальними протягом останніх років.

Робота присвячена використанню програмних засобів, що входять до системи педагогічних інструментальних авторських засобів Adobe Captivate [3, 11], для створення тестів з дисципліни "Організація баз даних та знань" і їхнього використання під час вивчення зазначеної дисципліни. Вибір системи зумовлений з одного боку широкими можливостями, а з іншого – її пропрієтарним рівнем, тобто вона не задовольняє критеріям вільного програмного забезпечення. За допомогою Adobe Captivate можна створювати і редагувати інтерактивні демонстрації програм, симуляції, підкасти, скрінкасти, ігри та уроки [12].

Основна частина

Дидактичні матеріали, що готуються засобами Adobe Captivate, подаються у вигляді проектів. Тести можуть входити у вигляді складової частини проекту електронного навчального посібника, або розроблятися як окремий проект. Авторами використовувалися обидва різновиди тестів. Під час підготовки елект-

ронного навчального посібника з модуля "Основи баз даних та знань" тести увійшли його складовою. Їх розмістили наприкінці кожного розділу. Такий завершений комплекс дидактичних матеріалів дозволив повністю забезпечити самостійне вивчення модуля. Тест як окремий проект застосовано до вивчення модуля "Класичні засоби доступу до даних", що дозволило проводити аудиторне тестування після завершення кожної лабораторної роботи.

З Captivate-тестами можна працювати в двох режимах: конструктора і тестування. У проект додається тест, що складається з низки завдань (режим конструктора).

Також в середовищі Adobe Captivate можна перевірити його роботу. Формально можна перевіряти виконання таких структурних одиниць: всього проекту; від поточного слайда і до кінця проекту та наступні N слайдів. Зовсім не важливо тестові це слайди чи ні.

Отриманий у середовищі Captivate проект, можна експортувати в один з таких форматів: в SWF (HTML + SWF), HTML5, PDF (лише однієї сторінки з вбудованим флеш), FLV, EXE, MP4 і APP (для MAC).

Отже можна побудувати тести для різних пристроїв (комп'ютер, планшет, смартфон) і різних операційних систем (Windows, Android та IOS).

У Captivate реалізовано два типи тестових завдань: відкриті і закриті. До відкритих завдань відносяться такі:

Fill-In-The-Blank – замість пропуску в тексті слід вписати певне слово;

ShortAnswer – для відповіді потрібно вписати слово або фразу;

RatingScale (Likert) – потрібно вказати ступінь своєї згоди з наведеними фразами.

Закриті завдання подано такими видами:

MultipleChoice – потрібно вибрати одну або кілька правильних відповідей з числа запропонованих;

True/False – на питання тесту необхідно відповідати "Так" або "Ні";

Matching – необхідно встановити відповідність між смисловими одиницями, наведеними у вікні питань і вікні відповідей;

HotSpot – слід правильно ідентифікувати області на слайді, клацнувши їх;

Sequence – відповідь повинна мати правильно визначений порядок об'єктів, дій тощо.

Отже в Captivate можна використовувати усі можливі види тестових завдань, а найпопулярніші види завдань з однією та кількома правильними відповідями подаються як Multiple Choice (відмінність між ними зазначається як опція).

Для окремих завдань можна встановити ліміт часу на очікування відповіді. Це важливо для автономних тестів, що виконуються в аудиторії для реалізації проміжного та підсумкового контролю навчальних досягнень студентів.

Для кожного завдання може встановлюватися вага. Для цього завдання позначається як Graded або Survey. У першому разі можна зазначити його вагу (важливість), у другому – ні. Тест може включати як завдання з вагою так і без неї. Для результату окремо виконується підрахунок правильних і неправильних відповідей та суми набраних ваг і можливої максимальної суми. Більш того, можна передбачити штраф (Penalty) за неправильну відповідь на певне питання, який віднімається від набраних балів. Все це дозволяє з максимальною точністю визначати знання студента.

Для проведення сеансу тестування завдання можуть вибиратися різним чином. Результати вибору можуть бути такими:

- всі завдання;
- N-завдань, які, випадковим чином вибираються з так званого пулу завдань;
- комбінація попередніх варіантів.

Для тестів, що вмонтовані в електронний навчальний посібник, було використано перший варіант, коли студент сам був зацікавлений в об'єктивній перевірці своїх знань. Для проведення тестування в аудиторії застосовувався другий варіант, що позбавило можливості списування. При цьому потужність пулу завдань значно перевищувала кількість завдань сеансу.

Результат тестування виводиться на екран. Однак в налаштуваннях для тестів є опція Enable reporting for this project, включення якої дає змогу експортувати або зберегти звіт в одному з основних міжнародних стандартів для систем дистанційного навчання. Ця можливість дозволяє автоматизувати

аналіз навчальних досягнень як окремого студента так і всієї групи.

На заняттях з дисципліни "Організація баз даних та знань" автори вважають доцільним виявити зв'язок між поняттями "база даних" та "тести". Той факт, що тест як дані подається у вигляді бази даних, є достатньо очевидним. Однак правильне поєднання вивчення матеріалу дисципліни "Організація баз даних та знань" з проведенням тестів може мати синергетичний ефект.

У найпростішому випадку база даних має схему, що подана на рис. 1. У ній містяться завдання виду "з однією правильною відповіддю".

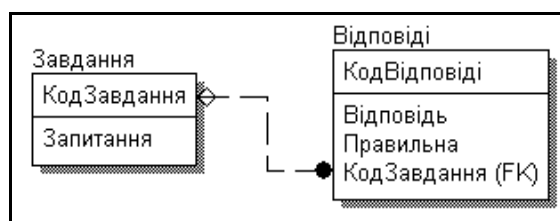


Рис. 1. Схема найпростішої бази даних тестів

Розгляд такої схеми дає ще один приклад найпоширенішого відношення між таблицями "один-до-багатьох". У наведеній схемі поле "Правильна" є ознакою правильності чи хибності відповіді і може мати булевий тип, але з урахуванням подальшої модифікації БД доцільно для цього поля обрати тип даних "ціле число". Тоді значення нуль відповідатиме помилці, а будь-яке інше – правильності відповіді.

Подання різних видів тестів доцільно використати для формування професійних навичок побудови інтерфейсу користувача. Під час тестування студент взаємодіє з інтерфейсом модуля тестування.

Спочатку розглянемо інтерфейс тестового завдання виду "з однією правильною відповіддю" (рис. 2). Напис "Одинарний вибір" нашоує студента на закладене раніше знання, що такий вид тесту подається у базі даних відношенням "один-до-багатьох". Візуальне сприйняття тестового завдання дає розуміння, що таке відношення може подаватися множиною елементів керування "RadioButton". Причому кількість цих елементів співпадає з кількістю записів у таблиці "Відповіді", що пов'язані з відповідним записом у таблиці "Завдання". Окрім варіанту завдання "з однією правильною відповіддю" дуже часто зустрічаються такі, що потребують декілька правильних відповідей – Multiple Choice ("множинний вибір") (рис. 3).

Реалізацію такого варіанту можна також здійснити за допомогою схеми, що подана на рис. 1. У цьому разі ознака правильної відповіді для завдання буде зустрічатися декілька разів у таблиці "Відповіді". Однак більш правильним підходом буде введення додаткового довідника "Типи Завдання", що дозволить диференціювати самі питання за типами, та,

відповідно до цього, організувати обробку даних і проведення тестування.

З огляду на це схема бази даних подана на рис. 4.

Одиночный выбор

Какой класс предназначен для установки связи между таблицами DataTable в кэше DataSet?

- A) DataRelation
- B) DbRelation
- C) TableRelation
- D) DataLink

Очистить!
Назад!
Пропустить!
Принять!

Рис. 2. Приклад завдання з однією правильною відповіддю

Множественный выбор

При создании запроса с помощью мастера все поля выбираются:

- A) только из одной таблицы
- B) только из всех таблиц базы данных
- C) только из таблиц
- D) только из существующих запросов и таблиц
- E) из разного количества таблиц

Очистить!
Назад!
Пропустить!
Принять!

Рис. 3. Приклад завдання з декількома правильними відповідями



Рис. 4. Модифікація найпростішої бази даних тестів

Приклад тестових завдань виду Sequence (послідовність) подано на рис. 5. Схема бази даних для завдань такого виду цілком вписується у схему, що подана на рис. 4, і не потребує додаткових полів. Це впливає з таких міркувань. Для завдань виду Sequence потрібно визначити правильний порядок відповідей чи дій, які вказані як допустимі відповіді. У цьому разі в полі "Правильна" можна зберігати порядковий номер відповіді у відповідній послідовності і теж задовольнятися схемою, що подано на рис. 4.

Зазначене вище щодо схеми бази даних можна також сказати про завдання типу Matching (відповідність). Його приклад подано на рис. 6.

Для завдань виду Matching необхідно визначити відповідність між виразами (реченнями, фразами тощо), що розташовані зліва та справа. Реалізацію подібних завдань також можна забезпечити схемою, що подана на рис. 4, якщо зробити прості припущення, а саме:

1. Тексти правої частини тестового питання мають значення поля "Правильна" дорівнює нулю.

2. Тексти лівої частини у полі "Правильна" мають посилання на поле "Код Відповіді" тексту правої частини.

Оскільки тестових завдань розглянутих вище видів достатньо багато, проходження тесту сприяє не тільки встановленню навчальних досягнень студента,

а також закріпленню знань з моделювання даних.

Особливого значення набуває вироблення навичок проектування інтерфейсу користувача. Студентам надається приклад того, що при тій самій схемі бази даних завдання різного виду подаються різним інтерфейсом. Це положення зазвичай важко сприймається під час навчання. Тому фокусування уваги на ньому під час занять сприяє виробленню креативного підходу під час проектування інтерфейсу користувача.

Отже, у найпростішому вигляді схема бази даних має усього три таблиці. Однак, звісно, її не можна вважати повною. У реальних умовах з'являються додаткові вимоги як до самих питань так і до організації проведення тестування, наприклад: можливість задавати вагу завдання, можливість встановлювати

довільний порядок виведення завдань у тесті чи відповідей, можливість зазначати час проходження тесту тощо. Крім того, тестів може бути багато і деякі завдання можуть входити до декількох з них. Враховуючи додаткові умови, отримуємо концептуальну базу даних, схема якої подана на рис. 7.

Якщо вага того самого завдання відрізняється для різних тестів, то відповідне поле буде присутнє в асоціативній таблиці "Склад Тесту" замість "Завдання". Обговорення цих питань на заняттях з дисципліни "Організація баз даних та знань" дозволяє продемонструвати властивість розвитку схеми бази даних залежно від розширення кола функціональних задач, які будуть розв'язуватися на основі даних, що зберігаються в базі.

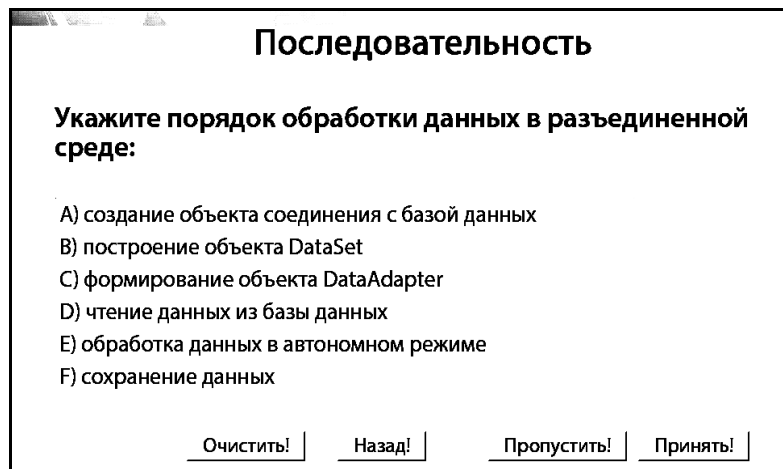


Рис. 5. Приклад завдання виду "Послідовність"

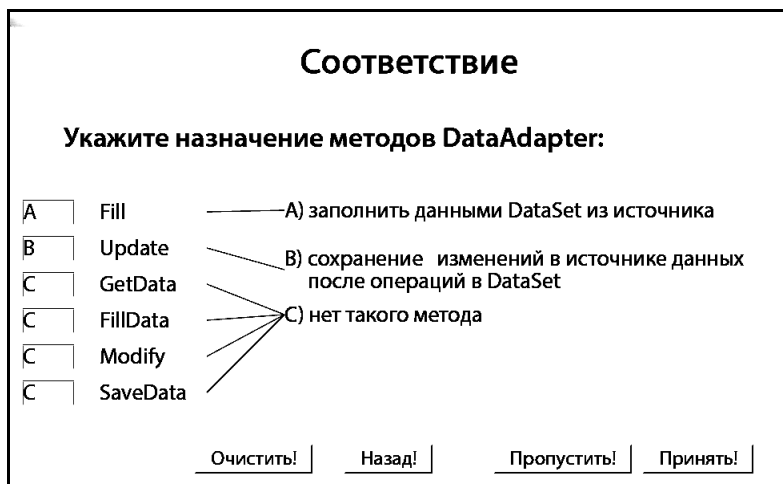


Рис. 6. Приклад завдання виду "Відповідність"

Висновки

В роботі розглянуто можливості інструментальних авторських засобів Adobe Captivate для створення тестів та методичні підходи до використання їх під час вивчення дисципліни "Організація баз даних та знань".

Дослідження показало, що система Adobe Captivate підтримує обидва типи тестових завдань – відкриті і закриті.

До складу відкритих завдань в систему входять Fill-In-The-Blank, ShortAnswer та RatingScale (Likert), а закритих – Multiple, True / False, Matching, HotSpot та Sequence.

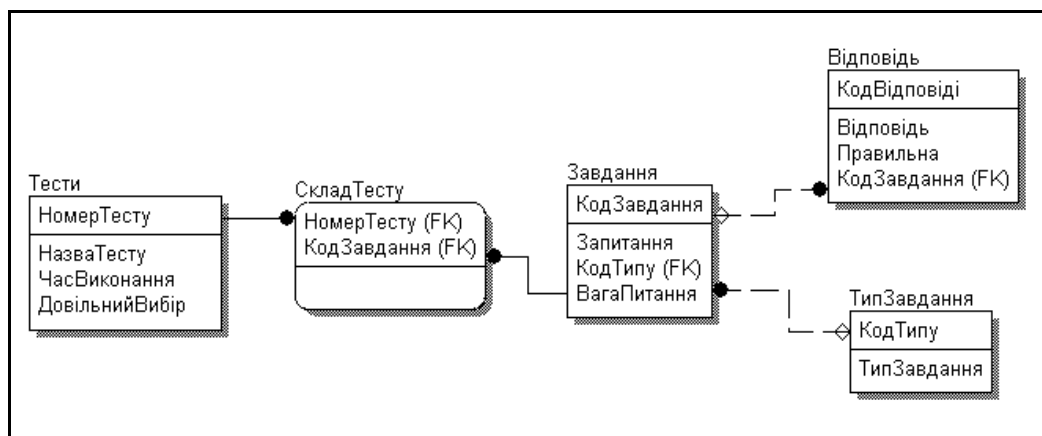


Рис. 7. Схема концептуальної бази даних для реалізації системи тестування

Підготовані в Adobe Captivate тести можна використовувати як складову частину проекту електронного навчального посібника так і автономно. Система має всі необхідні засоби щодо налаштування формату подання тесту та звіту після його проходження, ваги і часу виконання окремих завдань тощо.

На заняттях з дисципліни "Організація баз даних та знань" розглядає зв'язку між поняттями "база даних" та "тести" дозволяє поглибити знання з моделювання даних та проектування інтерфейсу користувача. Поєднання вивчення матеріалу дисципліни "Організація баз даних та знань" з проведенням тестів дає синергетичний ефект.

Список літератури

1. Абрамов И.А. Программная оболочка для автоматизированного создания теста / И.А. Абрамов, К.И. Юдин // Известия Пензенского государственного педагогического университета им. В.Г. Белинского. – 2008. – Вып. № 12. – С. 125-126.
2. Аванесов В.С. Композиция тестовых заданий / В.С. Аванесов. – М.: Центр тестирования, 2002. – 240 с.
3. Березовский В.С. Создание электронных учебных ресурсов и онлайн-обучение: учебн. пособ. / В.С. Березовский, И.В. Стеценко. – К.: Изд. группа ВНУ, 2013. – 176 с.
4. Ким В.С. Тестирование учебных достижений: монография / В.С. Ким. – Уссурийск: Издательство УГ-ПИ, 2007. – 214 с.

5. Нардюжнев В.И. Модели и алгоритмы информационно-вычислительной системы компьютерного тестирования: монография / В.И. Нардюжнев, И.В. Нардюжнев. – М.: Прометей, 2000. – 148 с.

6. Методичні рекомендації до проведення комп'ютерного тестового контролю / Укл. А.В. Бізюк, О.К. Пандорін, В.В. Федько. – Х.: Вид. ХНЕУ, 2006. – 32 с.

7. Программные оболочки, позволяющие создавать тесты на проверку знаний обучающихся. [Электронный ресурс]. – Режим доступа до ресурсу: <https://docs.google.com/document/d/1t9kUYKuW8h6cFPPKyR7awMXU4A-H6zPQQYxRmib8q1o/edit>.

8. Михаил-Тест. Оболочка для тестов. [Электронный ресурс]. – Режим доступа до ресурсу: <http://www.twirpx.com/file/9388/>.

9. Тестовые оболочки – Форум учителей об образовании в России и мире. [Электронный ресурс]. – Режим доступа до ресурсу: <http://pedsovet.ru/forum/78-204-1>.

10. INDIGO. Программа для создания тестов и онлайн тестирования. [Электронный ресурс]. – Режим доступа до ресурсу: <http://indigotech.ru/>.

11. Adobe Captivate 9 [Electronic resource]. – Access mode: <http://www.adobe.com/ru/products/captivate.html>.

12. Adobe Captivate. [Электронный ресурс]. – Режим доступа до ресурсу: https://ru.wikipedia.org/wiki/Adobe_Captivate.

Надійшла до редколегії 21.03.2016

Рецензент: д-р екон. наук, проф. О.І. Пушкар, Харківський національний економічний університет ім. С. Кузнеця, Харків.

САРТИВАТЕ-РЕАЛИЗАЦИЯ ТЕСТОВ ПО ДИСЦИПЛИНЕ "ОРГАНИЗАЦИЯ БАЗ ДАННЫХ И ЗНАНИЙ"

В.В. Федько, А.В. Тарасов

Исследованы возможности инструментальных авторских средств AdobeCaptivate для создания тестов и методические подходы к использованию их при изучении дисциплины "Организация баз данных и знаний". Предложено как методический прием изучение связи между понятиями "база данных" и "тесты" на занятиях по дисциплине "Организация баз данных и знаний", что позволяет углубить знания по моделированию данных и проектированию интерфейса пользователя.

Ключевые слова: педагогический тест, AdobeCaptivate, база данных, интерфейс пользователя.

CAPTIVATE-IMPLEMENTATION TEST ON THE DISCIPLINE "ORGANIZATION OF DATABASES AND KNOWLEDGE"

V.V. Fedko, O.V. Tarasov

The possibilities of Adobe Captivate authoring tools for creating test and methodical approaches to their use in the study of the discipline "Organization of databases and knowledge". It is proposed as a methodological procedure to study the connection between the concepts of "database" and "tests" in the classroom for discipline "Organization of databases and knowledge" that allows deepening the knowledge of data modeling and designing the user interface.

Keywords: pedagogical tests, Adobe Captivate, database, user interface.