

МЕТОДОЛОГИЯ ВЕБ-ТЕСТИРОВАНИЯ

В. А. Новиков¹⁾, Н. Н. Буснюк²⁾

¹⁾ *Белорусский государственный университет, пр. Независимости 4, 220030, г. Минск, Республика Беларусь, vanovikov@tut.by*

²⁾ *Белорусский государственный технологический университет, ул. Свердлова 13а, 220006, г. Минск, Республика Беларусь, n_bus@tut.by*

В связи с интенсивным развитием технологий компьютерного обеспечения учебного процесса встает задача создания компьютерного теста с учетом альтернативных вариантов простоты и сервисных возможностей. В работе предлагается методика компьютерного тестирования в среде Internet. Предлагается в оболочке теста практически не использовать сервер, что позволяет создавать программный продукт легко переносимый с компьютера на компьютер. Средства динамического HTML позволяют создавать программы с зашифрованным набором данных и фрагментами программного кода. Настройку теста предлагается осуществлять на языке высокого уровня C#. Рассмотрены также аспекты оценивания результатов тестирования с учетом сложности вопроса.

Ключевые слова: дистанционное обучение; тестирование; методология OLTP; вопросы; ответы; программирование; язык HTML; язык C#; язык JavaScript; среднее значение; среднеквадратичное отклонение.

Введение. Дистанционное обучение как нетрадиционная и перспективная форма преподавания получает все большее развитие в системе современного образования. Неотъемлемой частью дистанционного обучения являются компьютерные тесты. Компьютерное тестирование это основная составляющая не только дистанционного обучения, но и необходимым элементом традиционной формы обучения [2].

Процесс активного внедрения компьютерных тестов в учебный процесс сопровождается не только организационными проблемами, но и чисто техническими. Последнее связано с непростым процессом подготовки теста и дальнейшего его обновления.

Основная часть. При разработке программной оболочки теста необходимо учитывать современные тенденции и рекомендации. Таким базовым руководством может быть активно пропагандируемая в настоящее время методология OLTP (Online Transaction Processing). Применительно к информационным системам OLTP-системы предназначены для ввода, структурированного хранения и обработки информации (операций, документов) в режиме реального времени. Основными компонентами OLTP являются:

- механизмы транзакционного обслуживания клиентов;
- механизмы сбора статистической информации;
- механизмы подготовки и обновления базы данных.

Механизмы транзакционного обслуживания предусматривают применительно к компьютерному тестированию отсутствие задержек в процессе тестирования и простую и понятную систему интерфейсного обеспечения.

С позиций сбора статистической информации необходимо учитывать только самые необходимые для ведения учебного процесса данные и не усложнять программную оболочку чрезмерным ее администрированием. На наш взгляд сбор статистической информации можно ограничить для каждого вопроса теста числом его вывода и числом правильных ответов на этот вопрос. Такая статистика позволяет преподавателю делать периодически коррекцию весового коэффициента вопроса. Если учитывать программную оболочку теста, как независимую составляющую, то нет смысла вести статистику по фамилиям студентов, как при обучении, так и при тестировании. Подобная статистика привлекательна, но слишком усложняет администрирование и в связи с этим обновление теста.

Самым существенным механизмом организации компьютерного теста является механизм подготовки теста и его оперативное обновление. Наиболее полно методология OLTP обеспечивается комплексным использованием технологии Internet с применением в качестве связующего элемента языков высокого уровня (например С#).

При создании компьютерного теста надо учитывать возможность применения в нем помимо текста еще и формул, диаграмм и картинок. Наиболее полно эта возможность обеспечивается средствами Word. Оптимальным вариантом разметки является использование таблиц для формирования каждого вопроса. Необходимо предусмотреть три категории вопросов: вопрос перебор, вопрос-эталон и вопрос-список.

Для вопроса-перебора табличка представляется, как это показано на рис. 1. Символом @ обозначается строка параметров: весовой коэффициент вопроса, категория сложности вопроса и число колонок при выводе ответов. Символом \$ обозначается строка вопроса. Строки ответов вопроса-перебора идентифицируются символом "+" для правильного ответа и символом "—" для неправильного ответа. В вопросе, как и в ответах, может быть любой текст, кроме таблиц (в приведенном примере это формула). В компьютерном тесте вопрос-перебор выглядит, как это представлено на рис. 1

+	1.5
—	6
—	2
—	4
\$	Чему равно значение интеграла $\int_1^2 \frac{x}{x+5} dx$
@	Вес=1; Категория=2; Колонки ответов=4;

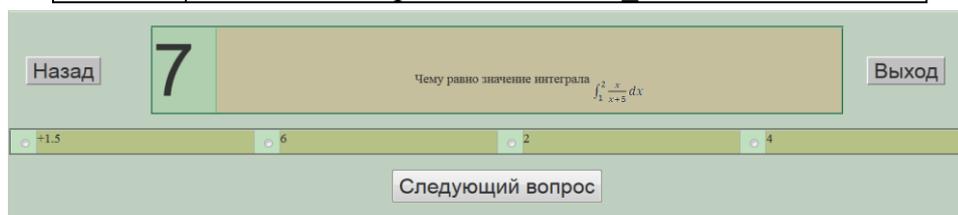


Рис. 1. Табличка для вопроса-перебора и ее отображение в тексте

Для вопроса-эталона табличка Word и ее отображение представлены на рис. 2. Как видно признак вопроса-эталона определяется по вариантам правильных ответов, идентифицируемых символом "*".

Для вопроса-списка табличка Word и ее отображение представлены на рис. 3. Как видно признак вопроса-списка определяется по вариантам правильного ответа в виде нумерации начиная с цифры 1.

Для дальнейшей обработки размеченного документа его необходимо перекодировать, записав из Word в формате HTML с фильтрацией. Дальнейшая обработка документа HTML зависит от средств организации интерфейсной части программной оболочки.

С позиций доступности компьютерного теста и обеспечения транзакционного обслуживания клиентов наиболее подходят средства организации интерактивных оболочек в Internet. Действительно, средства Internet позволяют организовывать доступный и понятный интерфейс средствами HTML и JavaScript. С другой стороны, программирование на стороне сервера позволяет обеспечить любой сложности сбор статистических данных.

@	Вес=1; Категория=1; Колонки ответов=1;
\$	Кто открыл теорию относительности
*	Эйнштейн
*	Альберт Эйнштейн
*	Эйнштейн Альберт

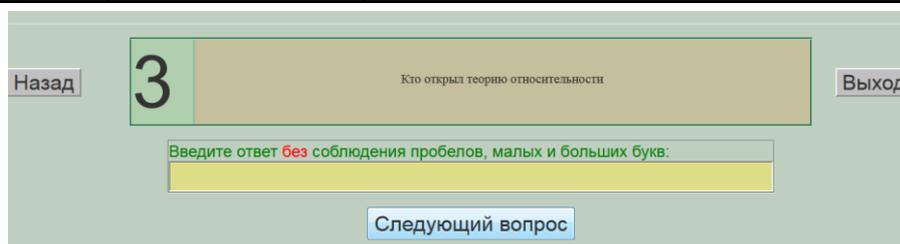


Рис. 2. Табличка для вопроса-эталона и ее отображение в тексте

С позиций методологии OLTP компьютерный тест должен в отсутствие статистики работать на любом локальном компьютере, чем обеспечивается принцип доступности теста. Таким образом, программная оболочка теста должна обслуживаться только средствами обработки на клиентской стороне с полным исключением из этого процесса серверной обработки данных.

Изложенная концепция компьютерного теста частично реализована в [1]. Тест предусматривает два режима: тестирование и обучение. Режим обучения отличается от тестирования только возможностью вывода после регламентированного числа попыток правильной комбинации ответов. В режиме обучения в качестве ответа засчитывается первая попытка. Кроме режима обучения тест предусматривает два уровня тестирования: уровень 1 облегченного варианта и уровень 2 полной версии.

\$	Расположите в хронологическом порядке по дате рождения писателей
1	Нарежный
2	Лев Толстой
3	Есенин
4	Булгаков

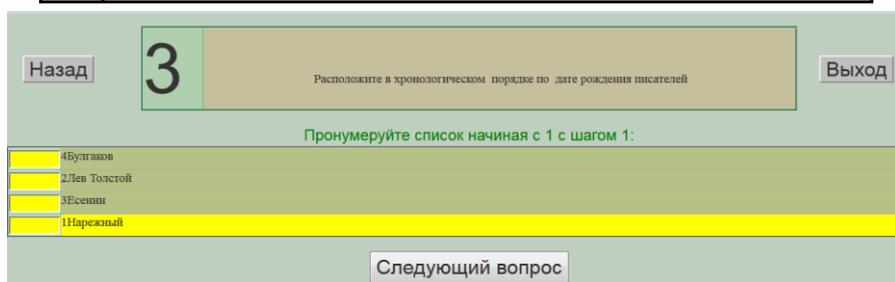


Рис. 3. Табличка для вопроса-списка и ее отображение в тексте

В конце тестирования выводится стандартный протокол тестирования рис. 1. В режиме сбора статистики протокол тестирования сопровождается фиксацией заданных вопросов и правильных ответов на них.

Структурировано тест предусматривает наличие нескольких разделов по предмету с возможностью тестирования по каждому разделу. Каждый раздел может содержать несколько тем с возможностью обучения по каждой теме. При тестировании по разделу число вопросов берется равномерным по каждой теме. Итоговое тестирование сводится к равномерному выбору числа вопросов по разделу, а внутри раздела и по темам.

Обеспечение сервиса и настройку теста наиболее оптимально организовывать на языке высокого уровня, которым является Visual Studio C#. В первую очередь необходимо

переконвертировать документ HTML каждой темы, полученный при подготовке теста в Word, в спецификацию данных на JavaScript. Для вопросов и ответов данные хранятся в ступенчатом массиве JavaScript. Строкового типа элемент массива представляет собой HTML-фрагмент, который и отображается в окне браузера.

Кроме конвертации на языке C# несложно создать и настройку теста с использованием механизма создания на C# соответствующих фрагментов JavaScript-кода. В этот сервис входит фиксация разделов, привязка темы к разделу, формирование критериев выставления оценок и необходимых режимов типа проверки теста или режима вывода зависимых переключателей для ответов. Протокол результата тестирования представлен на рис. 4.

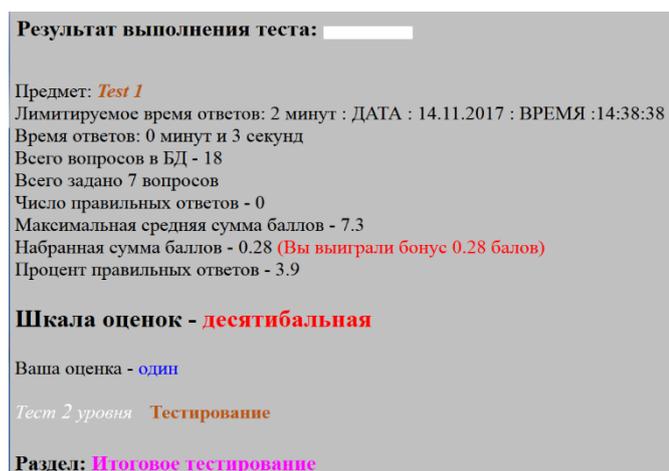


Рис. 4. Протокол тестирования

Непростой задачей при тестировании является определение критериев оценивания по каждому вопросу и затем по всем вопросам [3]. Анализ имеющихся точек зрения показал, что наиболее оптимальным является зачет правильного ответа на вопрос, если выставлены правильно все флажки ответов. В противном случае пришлось бы учитывать все возможные комбинации ответов и выставлять по каждой из них баллы, что практически невозможно. Для учета комбинаторной сложности ответов в вопросе-переборе предлагается, например для числа вариантов C_8^4 удваивать вес вопроса. Для меньших комбинаций C_n^m при $m < 4$ и $n < 8$ множительный коэффициент берется прямопропорционально.

Другой задачей оценивания результатов при случайном выборе вопросов темы является учет возможности появления комбинации легких вопросов, с одной стороны, или тяжелых вопросов, с другой стороны. Можно с позиций защиты прав потребителя предложить следующий алгоритм учета этой ситуации. Оценка выставляется по процентному отношению баллов правильных ответов к суммарному числу баллов. Фиксируется по всей теме средний вес вопроса, а при отображении среднее число баллов по числу выводимых вопросов. Если для тестирования выведены вопросы с числом баллов менее среднего, то оценивание выполняется по процентному соотношению правильных ответов именно среди этих выведенных вопросов. Если выведены сложные вопросы с суммой баллов больше средней, то берется отношение баллов правильных ответов к среднему числу баллов по всему тесту.

С другой стороны, можно предложить алгоритм тестирования при заданном количестве баллов и плавающим числом вопросов.

Прежде всего вычисляется среднее значение весового коэффициента по формуле:

$$S = \frac{1}{N} \sum_i w e s_i$$

Далее вычисляется среднее квадратичное отклонение, т. е. среднее отклонение от среднего:

$$\sigma = \sqrt{\frac{1}{N-1} \sum_i (wes_i - S)^2}$$

По заданному преподавателем числу выводимых для тестирования вопросов K вычисляется среднее значение суммы баллов, которую тестируемый должен набрать:

$$\Sigma = K \cdot S$$

Теперь при тестировании число задаваемых вопросов оказывается плавающим и определяется величиной Σ . Но если ограничиться только этим алгоритмом, то могут быть ситуации задания слишком малого K_{\min} или слишком большого K_{\max} числа вопросов. В связи с этим величины K_{\min} и K_{\max} ограничиваются по значению σ :

$$K_{\min} = \frac{\Sigma}{S + \sigma}; \quad K_{\max} = \frac{\Sigma}{S - \sigma}$$

Таким образом, при тестировании число задаваемых вопросов зависит от величины Σ , но не может быть меньше K_{\min} и больше K_{\max} .

Так алгоритм позволяет исключить необъективность оценивания при случайном выборе сложных вопросов.

Предлагаемая методология организации программной оболочки теста, таким образом, позволяет полностью автоматизировать процесс его подготовки. Кроме этого самая сложная задача разметки теста сводится к набору текста в Word с простой системой разметки вопросов и ответов. Использование в качестве буфера программной оболочки на C# дает возможность любой сложности предварительную настройку теста и его сервиса. Средства Internet используются только для организации удобного интерфейса и сбора необходимой статистики.

С позиций организации всего комплекса дистанционного обучения тест легко интегрируется в систему выдачи дидактического материала. Более того, при создании на основе предлагаемой методологии электронных учебников тест несложно монтируется в систему обеспечения сервиса электронного учебника. Тест из электронного учебника позволяет войти в оболочку минуя систему обычного доступа к теме. Тест дает возможность выдавать на сервер через элементы формы фамилию тестируемого и его оценку, причем фамилия поступает на тест из электронного учебника через командную строку. Подобный механизм обеспечивает синхронизацию многопользовательского режима при обслуживании электронного учебника.

Заключение. В заключение отметим, что учет рекомендаций, основанных на методологии OLTP, дает возможность избежать слишком запутанной системы навигации по программному продукту и обеспечить простую технологию наполнения базы данных данными. Учитывая, что компьютерный тест тоже специфическая база данных, использование прогрессивной технологии структурирования данных дает возможность создания самой простой процедуры подготовки теста средствами, непосредственно предназначенными для такого структурирования и отображения [4]. Средства динамического HTML позволяют шифровать данные и фрагменты программного кода JavaScript с использованием языка высокого уровня типа C#.

Библиографические ссылки

1. Новиков В.А., Шипулина Л.Г. Универсальный тест на базе Internet // Образовательно-инновационные технологии: теория и практика, книга 9. – Воронеж, 2009, с.43-54.
2. Никифоров О.Ю., Кокшарова Е.И. Комплекс признаков классификация систем компьютерного тестирования // Современные научные исследования и инновации. 2013. № 6

3. Звонников В.И.Современные средства оценивания результатов обучения: учеб. пособие для студ. высш. учеб. заведений/ В.И. Звонников, М.Б. Чельшкова.- М.: Издательский центр «Академия», 2007.-224 с.

4. Сиренко С.Н. Компьютерное тестирование в вузе: преимущества, особенности, опыт внедрения / С.Н. Сиренко // Педагогические измерения. – 2007. – № 4. – С. 67–74.