

В. А. Тарасов,

*канд. физ.-мат. наук, профессор кафедры информатики
Карельского государственного педагогического университета,*

А. С. Кюршунов,

*аспирант кафедры информатики
Карельского государственного педагогического университета*

ИНСТРУМЕНТАЛЬНОЕ ПРОГРАММНОЕ СРЕДСТВО ДЛЯ ГЕНЕРАЦИИ ТЕСТОВ С ВЫБОРОЧНЫМИ ОТВЕТАМИ НА ЯЗЫКЕ JAVASCRIPT

Наблюдаемый сегодня рост интереса к дистанционному обучению стимулирует разработку соответствующих средств и методов автоматизированного обучения и контроля знаний (что, в частности, нашло отражение в материалах федеральной целевой программы «Развитие единой образовательной информационной среды (2001—2005 годы)»). В настоящее время подавляющее большинство дистанционных курсов, проводимых на базе телекоммуникационной сети Интернет, включают обязательное тестирование слушателей для контроля за их учебной деятельностью. Однако создание компьютерных тестов для поддержки сетевого тестирования — это трудоемкий и длительный процесс, так как кроме подготовки собственно тестов нужно также создать необходимую программную поддержку диагностики результатов тестирования, защиты данных и т. д.

В данной статье речь пойдет о тестах для самоконтроля с выборочными ответами, реализованных в виде web-документов с использованием языка JavaScript.

Инструментальные программные средства для подготовки тестов

При разработке компьютерных тестов в большинстве случаев прибегают к их непосредственному программированию на языках JavaScript, Java и др. [1], что требует привлечения высококвалифицированных программистов и связано со значительными затратами труда и времени. Более производительным является использование инструментальных программных средств (ИПС), позволяющих сократить трудоемкость и сроки разработки компьютерных программ, а также дающих возможность лично участвовать в этом процессе преподавателям, не являющимся квалифицированными программистами.

Под ИПС в педагогическом смысле мы будем понимать совокупность программ, используемых в процессе разработки тестов для автоматизации обучения и контроля, вклю-

чающую в себя функциональные модули, обеспечивающие достаточно удобный интерфейс с пользователем. С их помощью создается адаптированная инструментальная среда для учителя-предметника, позволяющая реализовать формализованную методику контроля в виде компьютерного теста.

Одним из важнейших направлений разработки таких программных средств является создание ИПС контролирующего типа, генерирующих компьютерные тесты с выборочными ответами, что позволяет автоматизировать наиболее трудоемкую и рутинную часть работы учителя по контролю усвоения учебного материала, повторяющуюся в каждом очередном цикле обучения. Авторами статьи создано ИПС «TEST», позволяющее учителю-предметнику, не умеющему программировать, но владеющему навыками пользователя, создать базу данных по контролируемой теме в своей предметной области. После создания такой базы данных система «TEST» генерирует контролирующие тесты с выборочными ответами* (в виде web-документа).

Описываемое ИПС прошло успешную апробацию в локальной сети Intranet при тестировании по некоторым разделам информатики на первом курсе ФМФ КГПУ. Демонстрационная версия ИПС размещена по адресу <http://aks.dem.ru/resources.htm>.

Структура системы TEST

Структура системы «TEST» показана на рис. 1.

Редактор тестов позволяет создавать и редактировать файлы базы данных для теста с выборочными ответами и генерировать тест в виде web-документа. Этот редактор представляет собой оконное Java-приложение с именем TestEditor.jar, для запуска которого требуется *виртуальная машина*. Виртуальную машину Java (JVM — Java Virtual Machine) можно переписать с сайта фирмы SUN по адресу <http://java.sun.com>. После установки JVM (на момент написания статьи

* Рассматриваемая ниже программа TestEditor.jar также может быть адаптирована для генерации тестов с открытыми вопросами [2, 3].



Рис. 1. Структура ИПС «TEST»

текущей является версия 1.4) файл с расширением jar воспринимается операционной системой как стандартное приложение.

Заметим, что файл TestEditor.jar фактически представляет собой архив в формате zip и содержит требуемые для работы редактора Java-классы, шаблоны-заготовки генерируемых web-документов и модуль справки HelpContext.

Генерируемый тест в виде web-документа состоит из нескольких модулей (файлов). Модуль main.htm выводит в окне браузера инструктивный кадр, поясняющий последовательность действий обучаемого при ответе на предъявляемые вопросы, и передает управление модулю test.htm, открываемому в отдельном окне. Последний выполняет основную работу по тестированию и содержит формы с вопросами и вариантами ответов, а также коды вариантов ответов, генерирует в случайном порядке номера вопросов и извлекает сами соответствующие вопросы из базы данных baza.js. Модуль func.js содержит Java-скрипты, используемые в документе test.htm, и выводит результаты тестирования на экран. (Структура модулей main.htm, test.htm и func.js и выполняемые

ими функции описаны в [1], но в данном случае база данных baza.js выделена из модуля func.js в самостоятельный модуль. Это сделано для удобства формирования базы данных, если учитель-предметник не использует редактор тестов.)

Методика создания пользователем базы данных

Редактор TestEditor.jar позволяет создавать файлы баз данных, содержащие вопросы и варианты ответов по контролируемым темам. Прежде всего необходимо создать новую базу данных теста (пункт **Файл, Создать**). Затем в поле **Тема** (рис. 2) вводится название контролируемой темы (например, «Основные понятия информатики»), а в поле **Количество вопросов** указывается общее количество предъявляемых тестируемому вопросов. Потом в поле **Введите текст вопроса** вводится первый вопрос, а в поле **Введите ответ** — первый вариант ответа. Затем, после выбора в меню пункта **Правка, Добавить ответ**, вводится второй вариант ответа и т. д. При этом для правильного варианта ответа необходимо пометить флажок **Правильный ответ**.

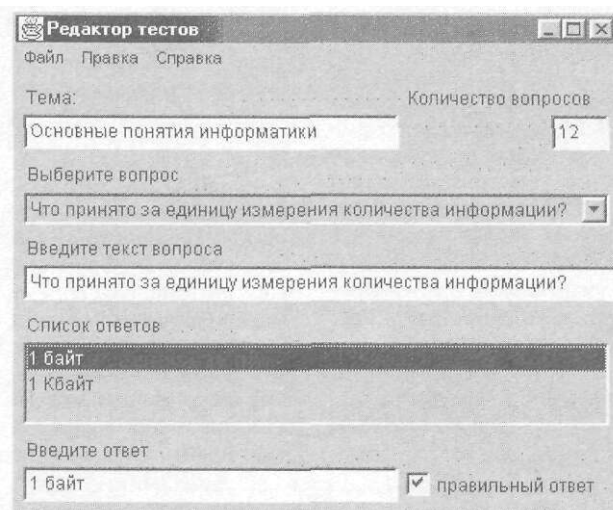


Рис. 2. Пример ввода вопроса и двух вариантов ответа

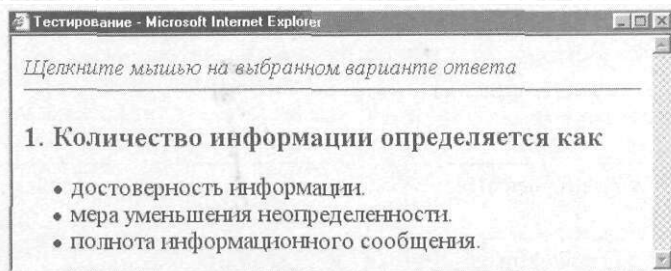


Рис. 3. Пример вопроса с тремя вариантами ответа

Далее, после выбора пункта меню Правка, Добавить вопрос, аналогичным образом вводятся все остальные вопросы и варианты ответов. По окончании заполнения базы данных тестового задания она записывается в отдельный файл (пункт меню Файл, Сохранить как...), который позже можно загрузить в редактор тестов для внесения изменений (пункт меню Файл, Открыть).

Когда база данных будет готова к использованию, на ее основе генерируется web-документ, реализующий тест с выборочными ответами для самоконтроля. Для этого нужно выбрать пункт меню Файл, Экспорт и в появившемся диалоговом окне ввести имя папки (каталога на диске), куда программа TestEditor.jar поместит четыре созданных ею файла: main.htm, test.htm, func.js и baza.js.

Методика проведения тестирования

Полученные файлы компьютерного теста main.htm, test.htm, func.js и baza.js могут быть интегрированы в дистанционный учебный курс (web-сайт) или использоваться самостоятельно. В последнем случае вышеперечисленные файлы загружаются на учебнические компьютеры по локальной сети либо с дискета.

После запуска на исполнение файла main.htm на экран выводится краткая инструкция для тестируемого, после чего сам процесс тестирования запускается щелчком мыши на расположенной здесь же кнопке Начать тест. При этом на экран выводится первый контрольный вопрос и варианты ответа на него (пример см. на рис. 3).

Выбор нужного варианта ответа осуществляется щелчком мыши, после чего выда-

ется соответствующий комментарий («Правильно»/»Неправильно»). В случае неправильного ответа учащемуся предоставляется еще одна попытка, а затем компьютер переходит к следующему вопросу. После завершения работы на экран выводятся результаты тестирования (рис. 4): общее количество предложенных вопросов, число верных ответов, словесная оценка (по четырехбалльной шкале), а также текст вопросов, на которые были даны неправильные ответы. При получении неудовлетворительной оценки можно повторить тестирование (так как вопросы генерируются в случайном порядке, а объем базы больше количества предъявляемых вопросов, новый вариант теста будет отличаться от предыдущего). Особо заметим, что использование вышеописанных тестов для итогового контроля не предполагается, поскольку обучаемый имеет возможность «вскрыть» базу данных и подсмотреть правильные ответы. Однако если выполнение теста проходит под контролем преподавателя, то результаты его выполнения могут быть засчитаны как итоговые.

Литература

1. Тарасов В. А., Тарасов В. В. Разработка контролирующих НТМБ-документов//Информатика и образование. 2001. № 3.
2. Тарасов В. А. Проектирование компьютерных тестов с открытыми вопросами//Информатика и образование. 2003. № 1.
3. Тарасов В. А., Кюршун А. С. Организация тестирования обучаемых в условиях локальной сети //Информатика и образование. 2003. № 2.

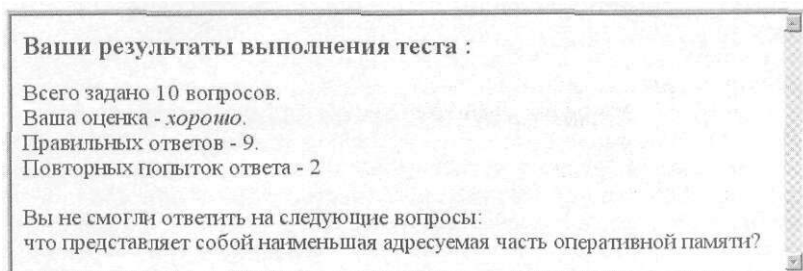


Рис. 4. Пример результатов тестирования