

О ПРОГРАММНОМ ПРОДУКТЕ АС «OPEN TEST» ДЛЯ ОПРЕДЕЛЕНИЯ РЕЙТИНГА СТУДЕНТОВ С ПОМОЩЬЮ JAVA ТЕХНОЛОГИЙ

Java технологиясынын жардамы менен студенттердин рейтингин аныктоо боюнча автоматташтырылган Open Test системасы сунушталат. Бул системанын артыкчылыгы кошумча оңдоо түзүүлөрдү талап кылбастан, билим берүү мекемелерине колдонууга жол ачат.

Предлагается автоматизированная система Open Test для определения рейтинга студентов с помощью Java технологий. Гибкость системы позволяет без дополнительных настроек и адаптаций применять систему в образовательных учреждениях.

This article is devoted to the automate system Open Test for definition of students rate with the helping of Java technologies. Flex systems in the studying without additional tuning.

В настоящее время компьютеры внедряются во все отрасли человеческой деятельности. Они помогают автоматизировать процессы, на выполнение которых требуется много времени. В этой статье мы представили программу для тестирования, которая может использоваться в учебном процессе. Задачей данного проекта является разработка Web-ориентированного приложения для тестового контроля знаний и определения рейтинга студентов. Основной особенностью Web-приложений является использование их большим количеством пользователей, которые могут использовать различные операционные системы. Таким образом, должна обеспечиваться независимость от платформы. Программа создана для автоматизации учебного процесса, получения расширенных возможностей сохранения результатов и ведения статистики тестирования. Процесс создания и администрирования тестами имеет иерархическую (или вложенную) структуру. Работа проектировалась с учетом разделения прав пользователей, что позволило оградить программу от несанкционированных доступов. Созданная программа значительно ускорит производительность труда преподавателей, сократит временные затраты студентов на прохождение контроля, создаст предпосылки для достоверности полученных оценок.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Определение функций и свойств адаптивной системы тестирования.
2. Разработка функциональной схемы с учетом спецификаций IMS QTI.
3. Разработка структуры и состава компонент системы тестирования.
4. Реализация.
5. Внедрение системы в учебный процесс.

Определим требования к функциональным характеристикам подсистемы проведения тестов:

1. Авторизация и аутентификация пользователя.
2. Управление пользователями и группами пользователей.
3. Управление тестами и группами тестов (подразумевает создание, удаление, редактирование, добавление)

Проверка, корректно ли составлен алгоритм (имеет ли логическое начало и конец, всегда ли можно добраться из точки начала в конечную точку алгоритма).

Технология Java Servlet является Java-классом, используемым для расширения функциональных возможностей серверов, на которых размещаются приложения, доступ к которым осуществляется посредством программной модели запрос-ответ. Несмотря на то, что сервлеты могут отвечать на любой тип запроса, они используются для расширения приложений, находящихся на Web-серверах. Для таких приложений технология Java Servlet определяет характерные для HTTP классы сервлетов. В конце данного пункта термин метод службы будет использован для любого метода класса сервлета, обеспечивающего службу клиенту.

Технология JSP-страниц (JavaServer Pages) позволяет без труда создавать web-содержимое, у которого есть как статическая, так и динамическая компоненты. JSP-технология воплощает все динамические возможности Java Servlet-технологии, однако обеспечивает при этом более естественный подход к созданию постоянного содержимого.

JSP-страницей является документ с текстовой основой, содержащий два типа текста: статические шаблонные данные, выражаемые при помощи любого формата на текстовой основе, такого как HTML, SVG, WML, и XML, а также JSP-элементы, которые создают динамическое содержимое.

Технология JavaBeans – это Java-классы многократного использования, которые можно объединять в приложения. Технология страниц JavaServer поддерживает использование JavaBean-компонентов с элементами языка JSP. Вы можете легко создавать и инициализировать компоненты, а также получать и устанавливать значения их свойств.

Реализация свойства при помощи переменной экземпляра не является необходимой. К свойству просто должен осуществляться доступ с помощью методов public, удовлетворяющих определенным условиям.

Элемент `jsp:useBean` объявляет, что в странице будет использован компонент, который хранится и к которому есть доступ из определенной области действия. Такой областью действия может быть приложение, сессия, запрос или страница.

При генерации страниц в Web возникает дилемма, связанная с архитектурой "клиент-сервер". Страницы можно генерировать как на стороне клиента, так и на стороне сервера. Последнее реализуется через механизм подстановок на стороне сервера (Server Site Includes). Компания Netscape распространила в 1995 году механизм управления страницами и на клиента, разработав язык программирования JavaScript. Таким образом, JavaScript – это язык управления сценариями просмотра гипертекстовых страниц Web на стороне клиента. Если быть более точным, то JavaScript – это не только язык программирования на стороне клиента. Liveware, прародитель

JavaScript, является средством подстановок на стороне сервера Netscape. Однако наибольшую популярность JavaScript обеспечило программирование на стороне клиента.

Взаимодействие приложения Java и расширения сервера Web

Методика организации взаимодействия приложений Java и расширений сервера Web основана на применении классов URL и URLConnection. Далее приложение получает ссылку на канал передачи данных с этим расширением как объекта класса URLConnection. Затем, пользуясь методами `getOutputStream` и `getInputStream` из класса URLConnection, приложение создает с расширением сервера Web выходной и входной канал передачи данных. Когда данные передаются приложением в выходной канал, созданный подобным образом, он попадает в стандартный поток ввода приложения CGI, как будто бы данные пришли методом POST из формы, определенной в документе HTML. На рис.1 показаны потоки данных для описанной выше схемы взаимодействия приложения Java и расширения сервера Web с интерфейсом CGI.

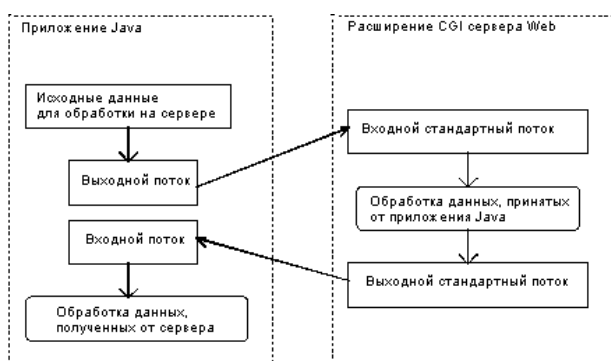


Рис. 1. Взаимодействие приложения Java с расширением сервера Web на базе интерфейса CGI

Вместо стандартного потока вывода также применяется специальная функция.

Общая функциональность программного продукта

При разработке данного программного продукта была избрана концепция визуального моделирование проекта, реализованная при помощи языка моделирования UML. В качестве среды проектирования была выбрана Ration Rose. Сначала была создана диаграмма вариантов использования, описывающая общую функциональность данного программного продукта.

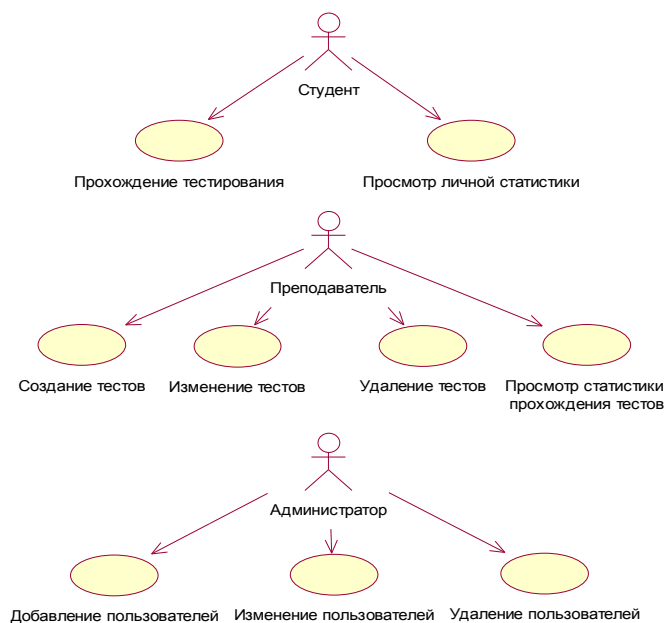


Рис. 2. Диаграмма вариантов использования

На данной диаграмме отображено взаимодействие программы с различными пользователями. Существует три типа пользователей: администратор, преподаватель, студент. Администратору доступны такие функции, как создание, изменение и удаление пользователей разных уровней. Преподавателю не доступны функции администратора, но доступны функции администрирования тестов и просмотра статистики их прохождения студентами. Студенту доступны лишь функции прохождения тестов.

Структура прикладной программы

Перед непосредственным созданием JSP страниц необходимо уяснить общую структуру программы и взаимодействие между страницами.

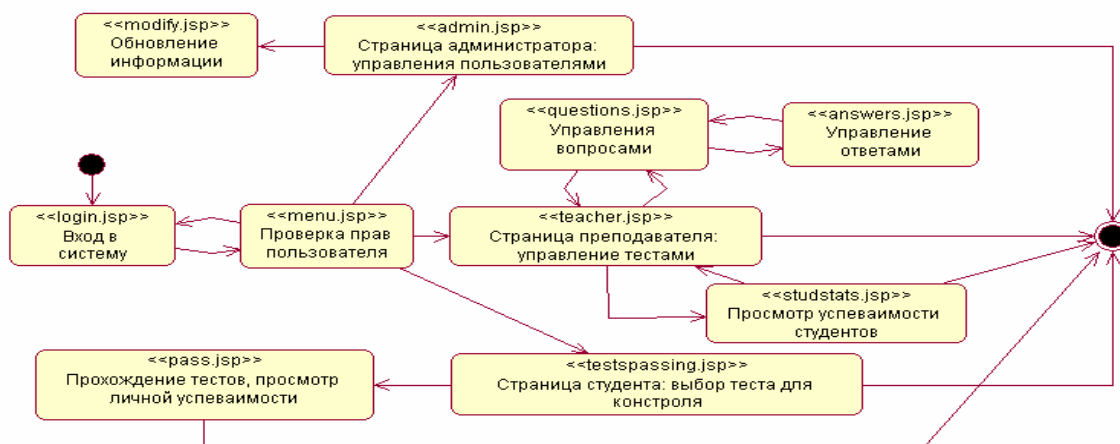


Рис. 3. Диаграмма навигаций

Работа с программой начинается со страницы login.jsp.

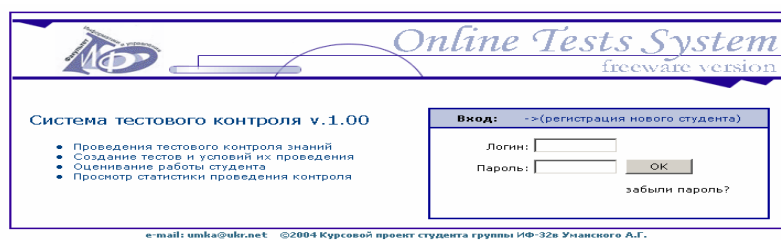


Рис. 4. Начальная страница программы login.jsp

При правильном вводе логина и пароля пользователь попадает на страницу main.jsp, которая направляет в зависимости от его прав на страницы admin.jsp, teacher.jsp и testspassing.jsp. Страница администратора admin.jsp позволяет создавать, редактировать и удалять пользователей. Страница преподавателя teacher.jsp позволяет создавать, редактировать и удалять тесты, вопросы и ответы к ним, просматривать статистику проведения тестового контроля. Страница студента testspassing.jsp позволяет студенту выбрать тест для прохождения и просматривать информацию о пройденных им тестах.

Проектирование интерфейса пользователя

Интерфейс проектируется исходя из диаграммы навигаций. На странице login.jsp необходимо реализовать форму с полями для ввода логина и пароля. Так как с главной страницы возможен переход на все остальные, а со всех страниц возможен возврат на главную страницу, то целесообразно реализовать меню навигации по сайту на каждой странице (см. рис. 5). На страницах admin.jsp, teacher.jsp и testspassing.jsp необходимо реализовать таблицы просмотра информации и формы для добавления, редактирования и удаления различных данных.

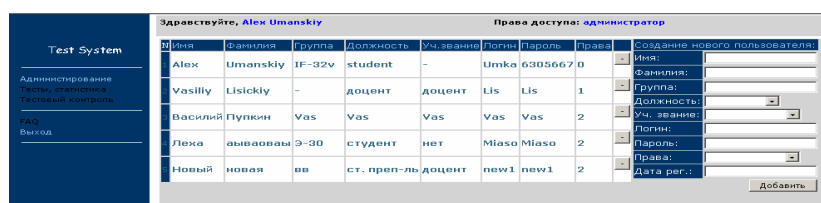


Рис. 5. Страница admin.jsp (меню – слева, таблица и форма – справа) и получение личной статистики

В рамках данной работы было спроектировано и реализовано Web-ориентированное приложение, предназначенное для проведения тестового контроля знаний студентов. При разработке использовались технологии сервлетов и JSP. Основной особенностью Web-службы является способность участников взаимодействовать друг с другом даже при использовании ими различных информационных систем.

При разработке была создана двухуровневая структура программного обеспечения. На верхнем уровне находится программное обеспечение, реализованное в виде сервлета или JSP-страницы. Нижний уровень представляет собой ПК клиента, на котором установлен Web-браузер.

СПИСОК ЛИТЕРАТУРЫ

1. Эккель Б. Философия Java. Библиотека программиста.– СПб.: Питер, 2001.
2. Хабибуллин И.Ш. Создание распределенных приложений на Java 2. – СПб.: БХВ-Петербург, 2001.
3. Дейтел Х.М., Дейтел П.Дж., Нието Т.Р., Лин Т.М., Садху П. Как программировать на ХМ./ Пер. с англ. – М.: ЗАО «Издательство БИНОМ», 2001.
4. Джейсон Мейнджер. Java: основы программирования /Пер. с англ. – К.: Издательская группа ВНУ, 1997. – 320 с.
5. Симкин Стив, Бартлет Нейл, Лесли Алекс. Программирование на Java. Путеводитель /Пер. с англ. – К. НИПФ «ДиаСофт Лтд», 1996. – 736 с.