

УДК 004.91

**ОПЫТ РАЗРАБОТКИ ИНФОРМАЦИОННОЙ СИСТЕМЫ  
ДЛЯ НЕСКОЛЬКИХ ТИПОВ ПОЛЬЗОВАТЕЛЕЙ****Койшыбекова Айжан Кумарбековна**

старший преподаватель  
Жетысуский государственный университет им. И. Жансугурова  
Талдыкорган (Казахстан)  
author@apriori-journal.ru

**Аннотация.** В работе рассматривается практическая часть создания информационной системы, которая позволила бы оптимально организовывать педагогический процесс. Проект представлен web-приложением, рассчитанным на использование тремя группами пользователей: администраторы, преподаватели и студенты.

**Ключевые слова:** информационная система; web-приложение; база данных; библиотека Hibernate.

---

**EXPERIENCE OF DEVELOPMENT OF INFORMATION SYSTEM  
FOR SEVERAL TYPES OF USERS****Koishybekova Ayzhan Kumarbekovna**

senior lecturer  
Zhetysusky state university of I. Zhansugurov, Taldikorgan (Kazakhstan)

**Abstract.** The practical part of creation of information system which would allow to organize pedagogical process optimum is considered. The project is submitted by the web application calculated on use by three groups of users: administrators, teachers and students.

**Key words:** information system; web application; database; Hibernate library.

Реализованный нами проект развивает опыт работы, практическая часть которой была выполнена посредством ASP.NET (язык C#) и СУБД MS SQL Server. Иной подход в реализации системы представлен web-приложением и базой данных под управлением MySql. Web-приложение написано на языке Java. Выбор данного языка в качестве инструмента разработки обусловлен тем, что для написания приложения требовался объектно-ориентированный язык, для которого созданы и поддерживаются библиотеки (каркасы), предназначенные для разработки web-приложений.

Целью работ являлось создание информационной системы, которая позволила бы получить данные об успеваемости студентов, расписании занятий, проводимых спецкурсах и важных объявлениях. В обеих реализациях проект представлен web-приложением, рассчитанным на использование тремя группами пользователей: администраторы, преподаватели и студенты.

Администраторы обладают возможностью редактировать, удалять и дополнять данные. Преподаватели могут изменить информацию, предоставляемую системой о преподаваемом предмете, а также разослать уведомления студентам. Студенты имеют доступ к персональной электронной зачетной книжке и прочим данным в режиме чтения. Поведение web-страниц зависит от роли пользователя: некоторые компоненты и функции могут быть доступны только определенной группе пользователей.

Функционально систему можно разделить на несколько независимых частей:

- обращение к базе данных и обработка результатов запросов;
- преобразование данных из базы в вид, удобный для отображения конечному пользователю, а так же организация реакции на действия пользователя;
- интерфейс доступа пользователя к данным.

Традиционно взаимодействие с базой данных производится посредством JDBC (Java Database Connectivity – соединение с базой данных на языке Java), реализация которого входит в стандартный набор инструментов Java. Преимуществами JDBC считаются:

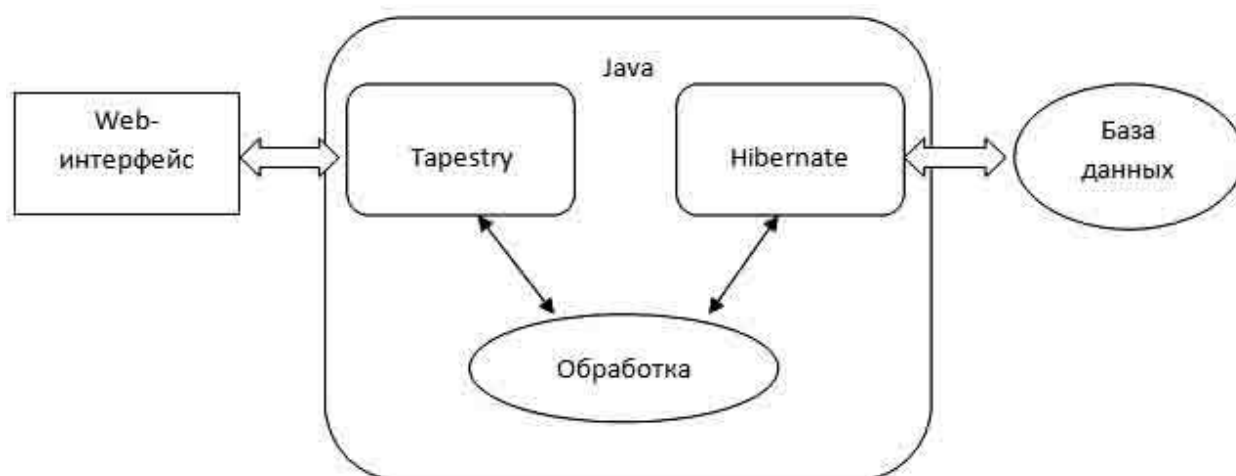
Отсутствие необходимости знать специфику базы данных, с которой производится работа. Более того, при переходе на другую БД нет нужды изменять приложение.

Все параметры соединения к БД указываются в URL.

Однако при работе с JDBC требуется самостоятельно открывать, поддерживать и закрывать соединение с базой, осуществлять обработку результата выполнения запроса. Намного удобнее организовывать работу с объектами из БД как с объектами Java. Решением задачи объектно-реляционного отображения является объектно-ориентированная библиотека Hibernate для языка Java.

С использованием Hibernate появилась возможность оперировать данными как объектами через методы-аксессоры (get и set), а также посредством именованных запросов на языке HQL (Hibernate Query Language). Конфигурация соединения с БД прописывается в отдельном xml-файле.

Структура проекта представлена на рисунке 1.



**Рис. 1. Структура потоков информации проекта**

Изначально для доступа к данным использовался модуль, содержащий статические методы, которые и производили операции с информацией. Под каждый запрос или операцию описывалась отдельная функция, что закономерно привело к большому количеству одинакового кода.

В новом варианте каждая таблица БД имела представление в виде java-класса (в терминах Hibernate – сущность), а чтение и запись ее полей производились через get- и set-методы. Для управления данными использовались специальные объекты – объекты доступа к данным (или DAO – Data Access Objects). В проекте каждая сущность имела отдельный DAO класс. Все специализированные DAO реализовывали интерфейс, предоставляющий 5 операций:

- добавление новой записи;
- удаление записи;
- изменение записи;
- получение записи по идентификатору;
- получение списка всех записей указанного типа.

Каждая операция производилась в общей реализации объектов доступа к данным готовыми средствами библиотеки Hibernate. Таким образом, количество программного кода свелось к минимуму.

Схема для сравнения обоих способов доступа к данным изображена на рисунке 2.

При разработке пользовательского интерфейса важно верно организовать ввод пользовательских данных, их проверку (валидацию) и реакцию приложения:

- 1) действие пользователя возбуждает некоторое событие;
- 2) программа обрабатывает событие и посылает запрос в базу данных;
- 3) результат запроса выводится через интерфейс.



**Рис. 2. Схемы доступа к данным**

Подобную событийную модель приложения помогает реализовать Tapestry-объектно-ориентированная библиотека для создания web-приложений, основанных на MVC.

Любая страница или компонент web-приложения Tapestry состоит из нескольких частей:

- Java-класс реализует логику работы страницы (компонента) на стороне сервера;
- JavaScript библиотеки и сценарии реагируют на действия пользователя;
- Файл шаблона компонента (TML – Tapestry Markup Language) задает визуальную составляющую.

Такое устройство компонент и страниц позволяет полностью отделить непосредственно интерфейс от механизмов работы приложения.

Упомянутые программные средства и библиотеки облегчают создание web-приложений: наборы готовых компонентов ускоряют разработку интерфейса, инструменты для работы с БД упростили работу с данными. Созданная с их помощью инфраструктура приложения позволяет расширять функционал системы.

## Список использованных источников

1. Боковиков А.М. Модус контроля как фактор стрессоустойчивости при компьютеризации профессиональной деятельности // Психол. журн. 2012. № 1.
2. Васильева И.А., Осипова Е.М., Петрова Н.Н. Психологические аспекты применения информационных технологий // Вопр. психол. 2009. № 3.
3. Осин А.В. Мультимедиа в образовании: контекст информатизации. М., 2005.
4. URL:<http://msdn.microsoft.com>
5. URL:<http://tapestry.apache.org>
6. URL:<http://www.w3schools.com/aspnet>