

В случае, если необходимо отменить применение  $F_1$ , необходимо взять  $I_1$  и затем далее применить все не применённые, но необходимые  $F_2, F_3, F_4$ .

В случае, если необходимо отменить применение  $F_2$ , необходимо также взять  $I_1$  и затем далее применить все не применённые фильтры, включая  $F_1$ .

В случае, если необходимо отменить применение  $F_3$ , вместо исходного  $I_1$  можно взять хранящееся промежуточное  $I_3$  и затем применить лишь  $F_4$ . Если же взять  $I_1$ , то необходимо также применять  $F_1, F_2$ . Таким образом видно, что хранение некоторых промежуточных версий может ускорение производства операций.

Применение новых фильтров аналогично удалению.

Для обработки HTTP-запросов к серверу использовались специальные классы-контроллеры с аннотацией `@Controller`. Методы, отвечающие за различные по смыслу, части приложения были вынесены в различные контроллеры. Были созданы следующие классы-контроллеры: `ImageController` для методов, отвечающий за запросы связанные с изображениями; `AuthController` для методов, отвечающий за запросы связанные с аутентификацией и авторизацией пользователей; `UploadController`, для методов, отвечающий за загрузку изображений на сервер; `MainController` для методов, отвечающий за главную страницу.

## Литература

1 Spring Documentation [Электронный ресурс] // URL: <https://spring.io/docs> (дата обращения: 29.04.2016).

УДК 004.4

*Д. И. Тимошенко, М. И. Жадан*

### РАЗРАБОТКА КЛИЕНТ-СЕРВЕРНОГО ПРИЛОЖЕНИЯ ДЛЯ ТЕСТИРОВАНИЯ ПОЛЬЗОВАТЕЛЯ НА ЗНАНИЕ JSE7

*Статья посвящена созданию клиент-серверного приложения на языке Java. Приложение предназначено для тестирования пользователей на знание основ языка Java, по таким темам как Strings, Variables, Operators and Expressions, Exceptions, Threads и др. При разработке использовались следующие технологии: язык Java и SQL, система управления базами данных MySQL и среда разработки IntelliJ Idea. Подробное изучение всех особенностей языка Java позволило составить списки наиболее полезных для ознакомления тестов. Приложение будет полезно как для новичков, так и для опытных Java-программистов, с целью приобретения или освежения знаний особенностей языка Java.*

Java является основой практически для всех типов сетевых приложений и всеобщим стандартом для разработки и распространения встроенных и мобильных приложений, игр, веб-контента и корпоративного программного обеспечения. В мире насчитывается более 9 миллионов специалистов, разрабатывающих приложения на Java, которая позволяет эффективно разрабатывать, внедрять и использовать превосходные приложения и услуги.

Реализованное приложение, осуществляет тестирование пользователей на знание базовых основ языка Java, представляет собой клиент-серверное приложение, написанное на языке Java, запрашивающее от пользователя регистрацию и авторизацию [1, 2]. После авторизации пользователю предлагается выбрать тему, по которой он хотел бы пройти тест. Тестирование представляет собой 10 вопросов с вариантами ответов.

После прохождения теста пользователю доступен его результат в виде таблицы верных и неверных ответов, а так же есть возможность отправки результата к себе на почту, указанную при регистрации. Так же приложение поддерживает хранение информации о прохождении тестов и другой статистики.

Создание графического приложения начинается с проектирования дизайна для форм, составляющих приложение. Создание формы начинается с создания класса, наследующего класс JFrame и реализующего интерфейс ActionListener из библиотеки Swing. Определяя конструктор для нового класса, создаем дизайн формы, размещаем на ней компоненты, а с помощью реализации интерфейса определяем события компонентов формы.

В приложении были созданы следующие формы:

- 1 AuthorizationForm – стартовая форма приложения, содержащая поля для ввода логина, пароля и кнопку для регистрации нового пользователя;
- 2 RegistrationForm – форма для регистрации нового пользователя;
- 3 MenuForm – форма, следующая за авторизацией пользователя, представляющая возможность выбора дальнейших действий;
- 4 TestingForm – форма представляющая процесс тестирования;
- 5 ResultForm – форма представляющая результаты тестирования;
- 6 CheckingForm – форма для просмотра отдельного вопроса теста.

Для создания форм использовались компоненты библиотеки Swing, такие как: JLabel, JPasswordField, JTextField, JButton, JRadioButton и др.

Регистрация пользователей и хранение статистики реализованы с помощью базы данных MySQL [3]. Для создания и работы с базой данных использовался MySQL Workbench инструмент.

Для хранения информации о пользователях и статистики в базе данных было создано множество таблиц, представлений и SQL-запросов.

Были созданы следующие таблицы:

- 1 Users – таблица для хранения информации о пользователях;
- 2 Sessions – хранение информации о завершенных и незавершенных сессиях;
- 3 Results – информация о прохождении тестов.

В ходе написания приложения были созданы следующие классы:

- 1 DBConnect – класс с методами для соединения приложения с базой данных;
- 2 User – класс, описывающий пользователя приложения. Содержит методы получения пользователя из базы данных, получение имени пользователя, а так же создания нового пользователя в базе данных;
- 3 Quiz – класс-фабрика, хранящий все темы, вопросы, а так же варианты ответов и списки верных ответов, хранит методы для получения списков тестов по определенным темам.

На рисунке 1 представлена форма авторизации. Форма включает два поля для ввода: имя пользователя и пароль. При вводе неверных имени пользователя или пароля будет выведено соответствующее сообщение. Зарегистрировать нового пользователя можно нажатием кнопки Registration. При этом будет выведена форма регистрации (рисунок 2). Новые пользователи сохраняются в базе данных в таблице Users.

При верном заполнении полей и нажатии кнопки Login on пользователь перейдет в главное меню программы, представленное на рисунке 3. Форма содержит MainMenu со списком тестов и статистикой приложения, а так же имя текущего пользователя и кнопки выбора тем тестирования.

При выборе темы тестирования в меню или нажатии на кнопку с соответствующим тестом будет показана форма тестирования, представленная на рисунке 4. Форма включает номер вопроса, вопрос, варианты ответов на него и кнопки ответа. Если пользователь закрывает форму тестирования, приложение сохранит текущий результат теста

в базу данных в таблицу Sessions. При последующем выборе этого теста, пользователь сможет продолжить тест с вопроса, на котором он остановился в прошлый раз.

Рисунок 1 – Форма авторизации

Рисунок 2 – Форма регистрации

Рисунок 3 – Меню

Рисунок 4 – Форма тестирования

После ответа на последний вопрос теста, результат теста будет сохранен в базе данных в таблице Results, будет выведена форма результата, показанная на рисунке 5.

Рисунок 5 – Форма результатов

Форма содержит информацию о пользователе, теме теста, и результате в виде текста и в графическом изображении (в виде иконок показывающих верные и неверные ответы).

При нажатии на кнопку под иконкой можно увидеть вопрос, ваш ответ на него и верный ответ. Результат на рисунке 6.

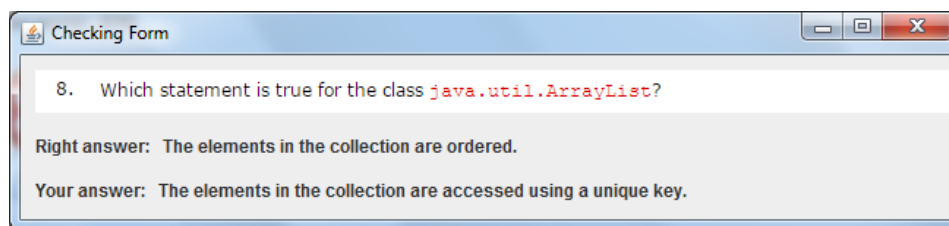


Рисунок 6 – Результат отдельного теста

Так же в приложении предусмотрены хранение общей статистики по прохождению тестов и статистики по каждому пользователю. Статистика доступна каждому пользователю из главного меню приложения.

В приложении реализована функция отправки результата тестирования на почту пользователя. Адрес электронной почты указывается при регистрации в базе данных приложения. Для отправки результата на электронный ящик необходимо после прохождения тестирования, на форме результата нажать на кнопку «Send me to email».

Все компоненты приложения были протестированы с использованием технологии JUnit. Функционал реализованного приложения может быть расширен и усовершенствован в перспективе до более высокого уровня, с целью дальнейшего повышения информативности, привлекательности и удобства, в зависимости от потребностей пользователей.

### Литература

- 1 Седжвик, Р. Алгоритмы на Java, 4-е издание / Р. Седжвик, К. Уэйн. – М.: Вильямс, 2013. – 848 с.
- 2 Gosling, J. The Java language Specification. Java SE 7 Edition / J. Goslong, B. Joy. – Oracle America Inc., 2013. – 644 с.
- 3 Васвани, В. MySQL: использование и администрирование / В. Васвани. – М.: Питер, 2014. – 368 с.

УДК 681.3

**И. И. Франков**

### ПРОГРАММНЫЕ СРЕДСТВА РЕАЛИЗАЦИИ РАСЧЁТА НАДЁЖНОСТИ СЛОЖНЫХ СИСТЕМ С ИСПОЛЬЗОВАНИЕМ ДЕРЕВЬЕВ ОТКАЗА

*В статье излагается один из подходов для анализа надёжности сложных систем с использованием аппарата деревьев отказа. Описываются средства автоматизации. Приводится пример, демонстрирующий расчёт оценки надёжности функционирования заправочной операции.*

**Введение.** Структуру сложной системы образуют основные элементы и связи между ними. К сбою в работе сложной системы могут приводить сбои в работе их элементов, отказы при функционировании системы управления, недостаточное количество и состав