

Подключение формы в Apache NetBeans к MySQL

Языков Кирилл Александрович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье рассматриваются основные шаги, необходимые для успешного установления соединения, включая настройку среды, создание базы данных и таблиц, а также выполнение SQL-запросов. Основные инструменты для работы позволяют легко интегрировать базы данных в приложения. Результаты исследования помогут начинающим программистам освоить основные шаги для эффективного взаимодействия с базой данных через графический интерфейс.

Ключевые слова: Базы данных, SQL-запросов, взаимодействие, графический интерфейс.

Connecting a form in Apache NetBeans to MySQL

Yazykov Kirill Alexandrovich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article discusses the basic steps required to successfully establish a connection, including setting up the environment, creating a database and tables, and executing SQL queries. The basic tools for the job make it easy to integrate databases into applications. The research results will help novice programmers to master the basic steps for effective interaction with the database through a graphical interface.

Keywords: Databases, SQL queries, interaction, graphical interface.

1 Введение

1.1 Актуальность

Современные приложения требуют эффективного взаимодействия с базами данных, что делает подключение к MySQL критически важным. Это позволяет разработчикам создавать динамичные и интерактивные интерфейсы, которые могут обрабатывать и отображать данные в реальном времени. Apache NetBeans предоставляет мощные инструменты для разработки, что упрощает процесс интеграции с MySQL. Использование этой среды разработки позволяет значительно ускорить создание приложений, обеспечивая удобный доступ к функционалу баз данных и улучшая производительность разработки. С учетом растущего спроса на веб-приложения, знание подключения форм к MySQL становится необходимым

навыком. Это знание открывает новые возможности для карьерного роста и позволяет разработчикам создавать более сложные и функциональные системы, соответствующие современным требованиям рынка.

1.2 Обзор исследований

В статье Я. В. Мищенко рассматривается понятие баз данных в контексте компьютерных систем и их значимость для различных сфер деятельности. Описываются основные виды баз данных, их структура и принципы работы. Анализируются преимущества использования баз данных, такие как эффективность хранения, обеспечение целостности данных и повышение безопасности. Особое внимание уделено современным тенденциям в области баз данных, включая облачные технологии и большие данные [1]. В своей работе «Базы данных. Системы управления базами данных» А. К. Иванова и А. Л. Куленцан рассмотрели такие понятия как «информационная система», «база данных», «система управления базами данных»; представили классификации информационных систем, баз данных и СУБД; привели преимущества и недостатки использования СУБД; а также рассмотрели этапы проектирования базы данных [2]. С. А. Митришин, В. Л. Симонов, М. В. Храпченко в учебном пособии полностью описали весь процесс проектирования, реализации и администрирования базы данных в СУБД MySQL отдельно и с использованием MySQL Workbench [3]. Рассмотрели технологию интеграции базы данных в веб-приложение после проектирования и наполнения А. А. Абдымомунов, М. Т. Жапаров [4]. А. С. Григоров в своей статье рассмотрел способы и методы обнаружения аномалий в SQL запросах к базе данных [6].

1.3 Цель исследования

Цель исследования заключается в разработке эффективного метода подключения форм в Apache NetBeans к MySQL. Это позволит улучшить взаимодействие между пользовательским интерфейсом и базой данных, обеспечивая более высокую производительность приложений. Также исследование направлено на изучение лучших практик и инструментов, способствующих упрощению процесса интеграции и повышению качества разработки.

2 Материалы и методы

Для создания базы данных будет использовано программное обеспечение MySQL Workbench. Для создания формы потребуется Apache NetBeans, а также драйвер для подключения к базе данных mysql-connector-j-8.1.0.

3 Результаты исследования

Для начала нужно создать базу данных в среде разработки MySQL Workbench (Рис. 1).

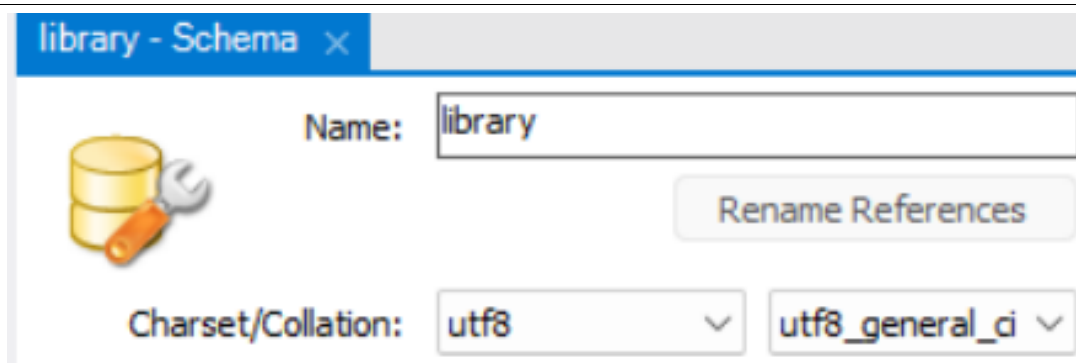


Рисунок 1 – Создание базы данных в MySQL Workbench

В результате в составе проекта должны быть следующие компоненты, как на рисунке. (Рис. 2).

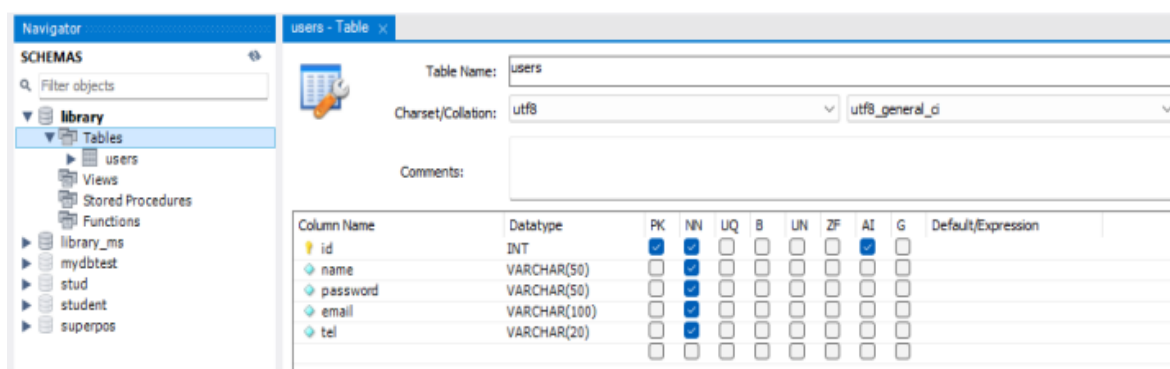


Рисунок 2 – Подключение компонентов в таблице

Так же для работы с БД потребуется подключение к проекту специального драйвера «mysql-connector-j-8.1.0» (Рис. 3).

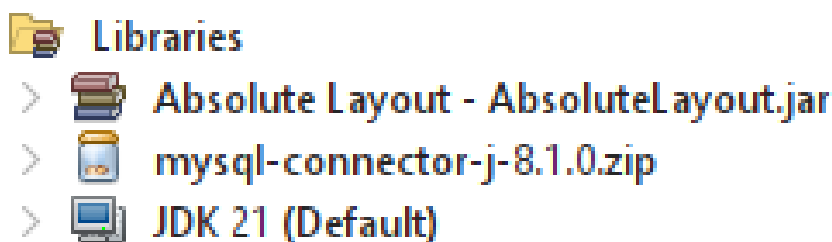


Рисунок 3 – Подключение драйвер

Наконец можно приступить на программном уровне к подключению БД к графической оболочке ИС. Для этого в проекте следует выбрать класс «SignupPage.java» и перейти в листинг кода программы, выбрав вкладку «Source». В листинге кода программы нужно создать новый метод. Название метода «insertSignupDetails». Данный метод будет отвечать за подключение графической формы к БД. (Рис. 4).

```
public class SignupPage extends javax.swing.JFrame {

    public SignupPage() {
        initComponents();
    }

    public void insertSignupDetails() {
```

Рисунок 4 – Создание нового метода

Следующим шагом будет являться объявление переменных. Количество переменных будет 4, имена переменных должны соответствовать именам столбцов таблицы «users» в БД «library» для облегчения.

```
public void insertSignupDetails() {
    String name = txt_username.getText();
    String password = txt_password.getText();
    String email = txt_email.getText();
    String tel = txt_tel.getText();
```

Рисунок 5 – Объявление переменных

Далее в классе «jframe» проекта требуется создать новый класс с названием «DBConnection» (Рис. 7, Рис. 6).

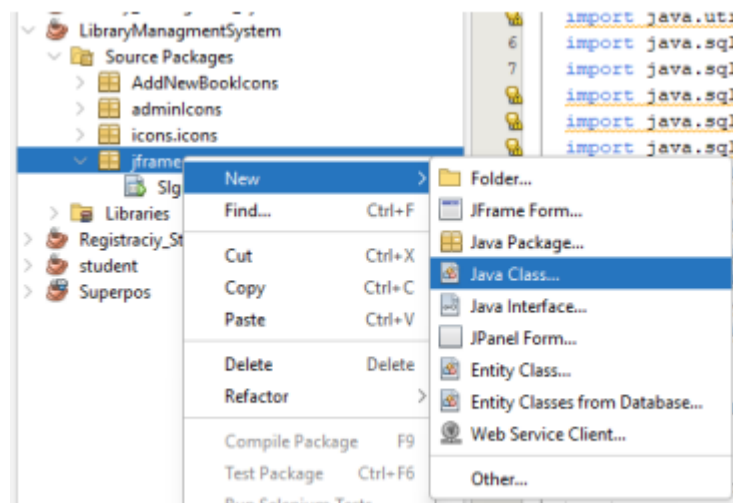


Рисунок 6 – Создание нового класса

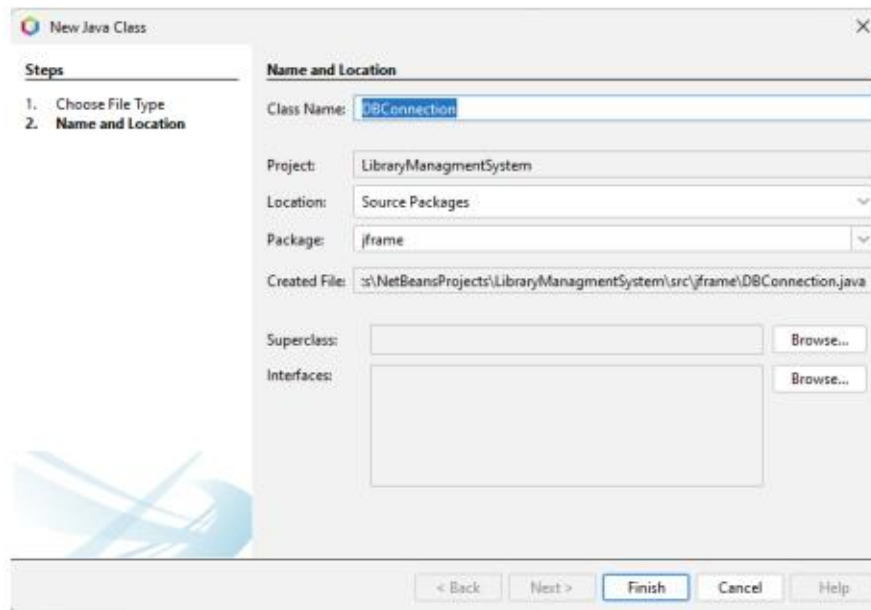


Рисунок 7 – Название нового класса

Следующим шагом следует открыть листинг кода нового класса «DBConnection» и прописать следующую конструкцию. Сначала необходимо подключить две библиотеки (Рис. 8).

```

package jframe;

import java.sql.Connection;
import java.sql.DriverManager;

/**
 *
 * @author ADMIN
 */
public class DBConnection {

```

Рисунок 8 – Подключение библиотек

Затем в классе «DBConnection» объявить переменную (Рис. 9).

```

1 package jframe;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5
6
7 public class DBConnection {
8     static Connection con = null;
9 }

```

Рисунок 9 – Объявление переменной

И наконец, в классе «DBConnection» следует прописать метод, в котором указывается само подключение БД к ИС. Единственное, что следует пояснить, так это то, почему метод подключения прописан не в основном коде программы, а в созданном для этого отдельном классе «DBConnection» (Рис. 10).

```
public class DBConnection {
    static Connection con = null;

    public static Connection getConnection(){
        try{
            Class.forName("com.mysql.jdbc.Driver");
            con = DriverManager.getConnection("jdbc:mysql://localhost:3306/library", "root","root");
        }catch (Exception e){
            e.printStackTrace();
        }
        return con;
    }
}
```

Рисунок 10 – Подключение к БД

После возвращения к графической форме, в листинге кода следует прописать следующую конструкцию.

```
try {

    //выполнили подключения к БД через созданный класс DBConnection
    Connection con = DBConnection.getConnection();
    //создаем MySQL запрос на добавление данных в таблицу
    String sql = "insert into users (name, password, email, tel) value (?, ?, ?, ?)";
    //поместим этот запрос в pst
    PreparedStatement pst = con.prepareStatement(sql);

    pst.setString(1, name);
    pst.setString(2, password);
    pst.setString(3, email);
    pst.setString(4, tel);

    //Объявляем int переменную.
    int updatedRowCount = pst.executeUpdate();
    //Если переменная имеет значение > 0 то выполняется условие
    if (updatedRowCount > 0){
        //прописываем условие если данные были добавлены то система сообщит пользователю об этом
        JOptionPane.showMessageDialog(this, "Пользователь успешно добавлен", "Внимание!", HEIGHT);
        //После добавления данных поля формы очищаются а курсор ввода переносится в первое поле
        txt_username.setText("");
        txt_password.setText("");
        txt_email.setText("");
        txt_tel.setText("");
        txt_username.requestFocus();

        LoginPage page = new LoginPage();
        page.setVisible(true);
        dispose();
    }else{
        //если данные не были добавлены то система так же сообщит пользователям об этом
        JOptionPane.showMessageDialog(this, "Произошла ошибка при регистрации", "Внимание", HEIGHT);
    }
} catch (Exception e){
    e.printStackTrace();
}
```

Рисунок 11 – Подключение к базе данных

Теперь запуск и проверка кода. Для этого вводятся данные в поля для ввода и обновляется база данных, чтобы увидеть результат записи (Рис. 12, Рис. 13).

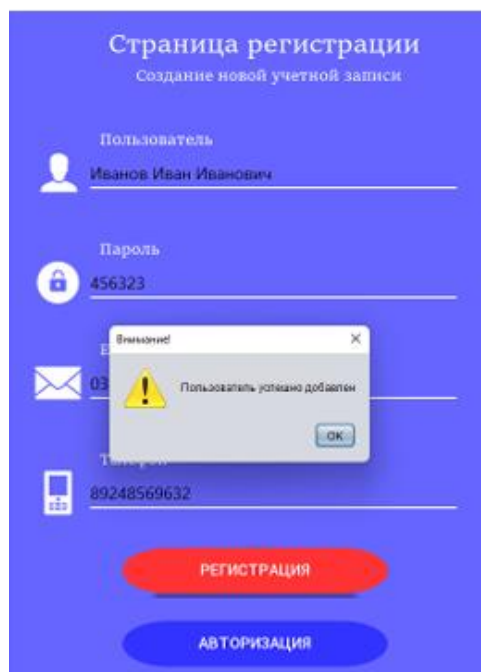
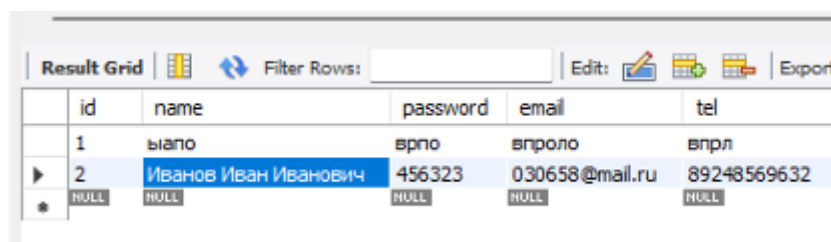


Рисунок 12 – Ввод данных в форме



	id	name	password	email	tel
	1	ивано	врпо	врполо	врпл
▶	2	Иванов Иван Иванович	456323	030658@mail.ru	89248569632
*	NULL	NULL	NULL	NULL	NULL

Рисунок 13 – Результат добавления записи в базе данных

Выводы

В данной статье подробно рассмотрен процесс подключения графической оболочки информационной системы к базе данных MySQL с использованием Apache NetBeans. Были изучены ключевые этапы, включая создание необходимых классов, объявление переменных и подключение драйвера «mysql-connector-j-8.1.0», что является основой для успешной интеграции. В результате, обеспечивая надежное взаимодействие между пользовательским интерфейсом и базой данных, что значительно улучшает функциональность приложения и пользовательский опыт.

Процесс подключения графической оболочки к базе данных MySQL в Apache NetBeans демонстрирует важность интеграции различных компонентов для создания функциональных приложений. Успешное выполнение всех этапов, от создания классов до обработки данных, позволяет разработчикам не только улучшить пользовательский интерфейс, но и обеспечить надежное хранение и управление данными. Это, в свою очередь,

способствует созданию более сложных и интерактивных систем, которые могут удовлетворять потребности пользователей.

Кроме того, использование «mysql-connector-j-8.1.0», значительно упрощает процесс подключения и взаимодействия с базами данных. Разделение логики подключения в отдельный класс, как в случае с «DBConnection», способствует лучшей организации кода и его поддерживаемости. В конечном итоге, освоение этих технологий и методов является необходимым шагом для разработчиков, стремящихся создавать современные и эффективные программные решения.

Библиографический список

1. Мищенко Я. В. Базы данных: современные тенденции в области баз данных и их значимость для различных сфер деятельности // Вестник науки. 2024. № 7. С 252 – 256.
2. Иванова А. К., Куленцан А. Л. Базы данных. Системы управления базами данных // Сборник научных трудов вузов России «Проблемы экономики, финансов и управления производством». 2023. № 52. С. 110-114.
3. Мартишин С. А., Симонов В. Л., Храпченко М. В. Проектирование и реализация баз данных в субд MySQL с использованием MySQL Workbench // ИД «Форум». 2012.
4. Абдымомунов А. А., Жапаров М. Т. Технология интеграции базы данных с WEB-приложением // Вестник Кыргызстана. 2019. 72-78 с.
5. Бегунов В. Н. Интеграция Веб-приложения ASP.net MVC с базой данных MySQL // Научное сообщество студентов столетия. Технические науки. 2018. С. 30-35.
6. Григоров А. С. О способе интеграции системы обнаружения аномалий в SQL запросах к базе данных на основе результатов выполнения запроса с приложениями, использующими субд в качестве хранилища данных // Молодой ученый. 2011. № 12-1. С. 21-24.