МИНИСТЕРСТВО ИНФОРМАТИЗАЦИИ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования

<<БЕЛОРУССКАЯ ГОСУДАРСТВЕННАЯ АКАДЕМИЯ СВЯЗИ>>

Отчёт

Лабораторная работа №4

Разработка консольного приложения для взаимодействия с базами данных

Вариант 5

Выполнил студент гр.СП141 Е.К. Дубовик

Проверила О.П. Рябычина

2023

**Задание**

• организацию соединения с базой данных вынести в отдельный класс, метод

которого возвращает соединение;

• создать БД. Привести таблицы к одной из нормированных форм;

• создать класс для выполнения запросов на извлечение информации из БД с

использованием компилированных запросов;

• создать класс на модификацию информации.

**Индивидуальное задание**

Сувениры. В БД хранится информация о сувенирах и их производителях.

Для сувениров необходимо хранить:

– название;

– реквизиты производителя;

– дату выпуска;

– цену.

Для производителей необходимо хранить:

– название;

– страну.

• Вывести информацию о сувенирах заданного производителя.

• Вывести информацию о сувенирах, произведенных в заданной стране.

• Вывести информацию о производителях, чьи цены на сувениры меньше заданной.

• Вывести информацию о производителях заданного сувенира, произведенного в

заданном году.

• Удалить заданного производителя и его сувениры

**Решение**

Исходный код программы можно открыть по [ссылке](https://github.com/Egor-Dubovik/java/tree/master/LR_4)(https://github.com/Egor-Dubovik/java/tree/master/LR\_4)

**Контрольные вопросы**

1. Что такое JDBC?

JDBC (Java Database Connectivity) - это стандартный интерфейс программирования на языке Java, по средствам которого получаем доступ к реляционным базам данных.

Используя JDBC, разработчики могут создавать приложения, которые работают с базами данных различных производителей, используя единый набор API. Благодаря этому, приложения на языке Java могут работать с любой базой данных, для которой доступен JDBC-драйвер (Oracle, MySQL, PostgreSQL, Microsoft SQL Server и др)

2. Этапы работы с базой данных посредством JDBC?  
  
1) Подключение к базе данных: для этого нужно загрузить JDBC-драйвер для используемой СУБД, создать объект Connection и передать ему параметры подключения (например, URL базы данных, имя пользователя и пароль).

2) Создание объекта Statement или PreparedStatement: это объекты, которые используются для отправки SQL-запросов на сервер базы данных. Statement подходит для выполнения статических запросов, а PreparedStatement - для выполнения динамических запросов с параметрами.

3) Выполнение SQL-запроса: для этого нужно вызвать метод execute() у объекта Statement или PreparedStatement. Результаты запроса можно получить с помощью методов getResultSet(), getUpdateCount() и getGeneratedKeys().

4) Обработка результатов запроса: если запрос возвращает данные, то их можно обработать с помощью объекта ResultSet. Для этого нужно вызвать методы next() и getXXX() (например, getString(), getInt() и т.д.), где XXX - это тип данных, который ожидается в столбце.

5) Закрытие ресурсов: после того как запрос выполнен и результаты обработаны, нужно закрыть все открытые ресурсы (Statement, PreparedStatement, ResultSet и Connection) с помощью методов close().

3. Для чего используется шаблон проектирования DAO?

Шаблон проектирования DAO (Data Access Object) используется для управления доступом к данным в приложении. Его цель состоит в том, чтобы создать абстрактный интерфейс для доступа к данным, который скрывает детали реализации и предоставляет единый способ работы с данными в приложении.

DAO помогает разделить логику приложения на две части: бизнес-логику и логику доступа к данным. Благодаря этому, изменения в логике доступа к данным не повлияют на бизнес-логику и наоборот.

4. Как организовать соединение с базой данных (Database connections)?

1) Загрузить драйвер базы данных, используя Class.forName(). Например, для MySQL можно использовать следующий код:

Class.forName("com.mysql.cj.jdbc.Driver");

2) Создать объект Connection с помощью метода DriverManager.getConnection(). Например:

String url = "jdbc:mysql://localhost:3306/mydatabase";  
String user = "username";  
String password = "password";  
Connection connection = DriverManager.getConnection(url, user, password);

3) После завершения работы с базой данных необходимо закрыть соединение с помощью метода Connection.close(). Например:

1. connection.close();