

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ИМ. Р.Е. АЛЕКСЕЕВА»**

Институт радиоэлектроники и информационных технологий
Кафедра “Прикладная математика”

Отчет по лабораторной работе курса “Базы данных”

Лабораторная работа №1
«Создание базы данных с использованием H2 и JDBC»

Выполнил студент группы 18-ПМ:

Винокуров М.С

Проверил:

Моисеев А.Е.

НИЖНИЙ НОВГОРОД

2021 г.

Оглавление

Введение.....	3
JDBC	3
H2.....	4
Java	5
Задание	6
Выполнение работы.....	7
Вывод	10
Список источников	11
Приложение. Листинг.....	12

Введение

JDBC

JDBC (англ. Java DataBase Connectivity — соединение с базами данных на Java) — платформенно независимый промышленный стандарт взаимодействия Java-приложений с различными СУБД, реализованный в виде пакета `java.sql`, входящего в состав Java SE.

JDBC основан на концепции так называемых драйверов, позволяющих получать соединение с базой данных по специально описанному URL. Драйверы могут загружаться динамически (во время работы программы). Загрузившись, драйвер сам регистрирует себя и вызывается автоматически, когда программа требует URL, содержащий протокол, за который драйвер отвечает.

JDBC API содержит два основных типа интерфейсов: первый – для разработчиков приложений и второй (более низкого уровня) – для разработчиков драйверов.

Соединение с базой данных описывается классом, реализующим интерфейс `java.sql.Connection`. Имея соединение с базой данных, можно создавать объекты типа `Statement`, служащие для исполнения запросов к базе данных на языке SQL.

Существуют следующие виды типов `Statement`, различающихся по назначению:

- `java.sql.Statement` — `Statement` общего назначения;
- `java.sql.PreparedStatement` — `Statement`, служащий для выполнения запросов, содержащих подставляемые параметры (обозначаются символом '?' в теле запроса);
- `java.sql.CallableStatement` — `Statement`, предназначенный для вызова хранимых процедур.

Интерфейс `java.sql.ResultSet` позволяет легко обрабатывать результаты запроса.

H2

H2 — открытая кроссплатформенная СУБД, полностью написанная на языке **Java**.

Несмотря на малый размер (чуть более 1 МБ) H2 поддерживает следующие возможности «из коробки»:

- Два режима работы (клиент-сервер, встроенный)
- Два режима хранения данных (файловая система, память)
- Поддержка планов выполнения запросов
- Поддержка кластеризации и репликации
- Шифрование данных
- Внешние (связанные) таблицы
- Драйвер ODBC
- Полнотекстовый поиск
- Определение доменов
- Мультиверсионный конкурентный доступ
- Поддержка последовательностей
- Поддержка ключевых слов LIMIT и OFFSET в запросах
- Временные таблицы
- Вычисляемые столбцы
- Пользовательские агрегатные функции
- Пользовательские хранимые процедуры
- Сжатие CLOB/BLOB объектов
- Работа с CSV файлами на чтение и запись
- Браузерная консоль управления

Java

Java — строго типизированный объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle). Разработка ведётся сообществом, организованным через Java Community Process, язык и основные реализующие его технологии распространяются по лицензии GPL. Права на торговую марку принадлежат корпорации Oracle.

Приложения Java обычно транслируются в специальный байт-код, поэтому они могут работать на любой компьютерной архитектуре с помощью виртуальной Java-машины. Дата официального выпуска — 23 мая 1995 года. На 2019 год Java — один из самых популярных языков программирования.

Программы, написанные на Java, имеют репутацию более медленных и занимающих больше оперативной памяти, чем написанные на языке C. Тем не менее, скорость выполнения программ, написанных на языке Java, была существенно улучшена с выпуском в 1997—1998 годах так называемого JIT-компилятора в версии 1.1 в дополнение к другим особенностям языка для поддержки лучшего анализа кода (такие, как внутренние классы, класс StringBuffer, упрощённые логические вычисления и так далее). Кроме того, была произведена оптимизация виртуальной машины Java — с 2000 года для этого используется виртуальная машина HotSpot. По состоянию на февраль 2012 года, код Java 7 приблизительно в 1,8 раза медленнее кода, написанного на языке Си.

Некоторые платформы предлагают аппаратную поддержку выполнения для Java. К примеру, микроконтроллеры, выполняющие код Java на аппаратном обеспечении вместо программной JVM, а также основанные на ARM процессоры, которые поддерживают выполнение байткода Java через опцию Jazelle.

Задание

- Скачать драйвер JDBC для встраиваемой базы данных H2
- Подключить драйвер к проекту Java
- Подключиться к базе из программы Java
- Создать таблицу и загрузить данные из Лабораторной работы №0
- Выполнить несколько запросов SELECT

Выполнение работы

Был создан проект Java, состоящий из одного класса GameDB. В классе был реализован один метод: `public static void main(String[] args)`, являющийся точкой входа в программу. База данных работает с «Играми»
В этом методе выполняются следующие действия:

Загружается драйвер для работы с базой данных(пути в переменных):

```
public class GameDB {  
    public static final String DB_URL = "jdbc:h2:/c:/Users/super/IdeaProjects/GameDB/db/GameDB";  
    public static final String DB_Driver = "org.h2.Driver";  
  
    public static void main(String[] args){  
        try{  
            Class.forName(DB_Driver);
```

Выполняется подключение к базе данных:

```
Connection connection = DriverManager.getConnection(DB_URL);
```

Создается таблица:

```
try{  
    st.execute("CREATE TABLE GameDB( " +  
        "id BIGINT AUTO_INCREMENT PRIMARY KEY," +  
        "title VARCHAR(255) NOT NULL," +  
        "price INT NOT NULL," +  
        "year VARCHAR(255) NOT NULL," +  
        "primaryStore VARCHAR(255) NOT NULL," +  
        ");");  
    System.out.println("Таблица успешно создана.");  
}
```

Добавляются данные в таблицу:

```
if(!st.executeQuery("SELECT * from GameDB").next()){  
    st.execute("INSERT into GameDB values(1, 'Fallout', '10', '1994',  
    'Bethesda Launcher');");  
    st.execute("INSERT into GameDB values(2, 'Fallout 2', '10', '1998',  
'Bethesda Launcher');");  
    st.execute("INSERT into GameDB values(3, 'Fallout Nevada', '20',  
'2007', 'non');");  
    st.execute("INSERT into GameDB values(4, 'Fallout New Vegas', '20',  
'2009', 'Steam');");  
    st.execute("INSERT into GameDB values(5, 'Halo Combat Evolved', '35',  
'2015', 'Xbox Live');");  
    st.execute("INSERT into GameDB values(6, 'Halo 2', '40', '1994',  
'Xbox Live');");  
    st.execute("INSERT into GameDB values(7, 'Dota 2', '25', '2001',  
'Steam');");
```

```

        st.execute("INSERT into GameDB values(8, 'League of Legends', '25', '2003', 'RIOT Launcher');");
        st.execute("INSERT into GameDB values(9, 'Counter-Strike Condition Zero', '25', '2005', 'Steam');");
        st.execute("INSERT into GameDB values(10, 'Call of Duty 3', '25', '2007', 'Xbox Live');");
        st.execute("INSERT into GameDB values(11, 'Horizon Zero Dawn', '30', '2008', 'PS Network');");
        st.execute("INSERT into GameDB values(12, 'The last of Us', '10', '2004', 'PS Network');");
        st.execute("INSERT into GameDB values(13, 'Fallout 76', '20', '2010', 'Bethesda Launcher');");
        st.execute("INSERT into GameDB values(14, 'Halo 5', '40', '2015', 'Xbox Live');");
        st.execute("INSERT into GameDB values(15, 'Stubbs the Zombie', '35', '2013', 'Xbox Live');");
        st.execute("INSERT into GameDB values(16, 'Battlefield 1942', '40', '2003', 'Retail');");
        st.execute("INSERT into GameDB values(17, 'Team Fortress 2', '40', '2014', 'Steam');");
        st.execute("INSERT into GameDB values(18, 'Half-Life', '40', '2014', 'Steam');");
        st.execute("INSERT into GameDB values(19, 'Half-Life 2', '40', '2005', 'Steam');");
        st.execute("INSERT into GameDB values(20, 'Half-Life 3', '40', '2077', 'Steam');");
    }

```

Делается запрос к базе и его результат выводится в консоль:

```

ResultSet rs = st.executeQuery("SELECT * from GameDB WHERE ((year < 2001)or(year > 2004))");

    while (rs.next()){
        System.out.println(
            "Игра: " + rs.getString("title") + " | Цена: " + rs.getString("price") + "$, Год выпуска: " + rs.getString("year") + ", Магазин: " + rs.getString("primaryStore")
        );
    }

    connection.close();
    System.out.println("Соединение было закрыто.");

```


Результатом этого запроса является вся таблица в базе данных.

Вывод в консоли:

```
Таблица успешно создана.  
Игра: Fallout | Цена: 10$, Год выпуска: 1994, Магазин: Bethesda Launcher  
Игра: Fallout 2 | Цена: 10$, Год выпуска: 1998, Магазин: Bethesda Launcher  
Игра: Fallout Nevada | Цена: 20$, Год выпуска: 2007, Магазин: non  
Игра: Fallout New Vegas | Цена: 20$, Год выпуска: 2009, Магазин: Steam  
Игра: Halo Combat Evolved | Цена: 35$, Год выпуска: 2015, Магазин: Xbox Live  
Игра: Halo 2 | Цена: 40$, Год выпуска: 1994, Магазин: Xbox Live  
Игра: Dota 2 | Цена: 25$, Год выпуска: 2001, Магазин: Steam  
Игра: League of Legends | Цена: 25$, Год выпуска: 2003, Магазин: RIOT Launcher  
Игра: Counter-Strike Condition Zero | Цена: 25$, Год выпуска: 2005, Магазин: Steam  
Игра: Call of Duty 3 | Цена: 25$, Год выпуска: 2007, Магазин: Xbox Live  
Игра: Horizon Zero Dawn | Цена: 30$, Год выпуска: 2008, Магазин: PS Network  
Игра: The last of Us | Цена: 10$, Год выпуска: 2004, Магазин: PS Network  
Игра: Fallout 76 | Цена: 20$, Год выпуска: 2010, Магазин: Bethesda Launcher  
Игра: Halo 5 | Цена: 40$, Год выпуска: 2015, Магазин: Xbox Live  
Игра: Stubbs the Zombie | Цена: 35$, Год выпуска: 2013, Магазин: Xbox Live  
Игра: Battlefield 1942 | Цена: 40$, Год выпуска: 2003, Магазин: Retail  
Игра: Team Fortress 2 | Цена: 40$, Год выпуска: 2014, Магазин: Steam  
Игра: Half-Life | Цена: 40$, Год выпуска: 2014, Магазин: Steam  
Игра: Half-Life 2 | Цена: 40$, Год выпуска: 2005, Магазин: Steam  
Игра: Half-Life 3 | Цена: 40$, Год выпуска: 2077, Магазин: Steam  
Соединение было закрыто.
```

И наложение дополнительных ограничений на вывод
(строго младше 2004 года):

```
Игра: Fallout Nevada | Цена: 20$, Год выпуска: 2007, Магазин: non  
Игра: Fallout New Vegas | Цена: 20$, Год выпуска: 2009, Магазин: Steam  
Игра: Halo Combat Evolved | Цена: 35$, Год выпуска: 2015, Магазин: Xbox Live  
Игра: Counter-Strike Condition Zero | Цена: 25$, Год выпуска: 2005, Магазин: Steam  
Игра: Call of Duty 3 | Цена: 25$, Год выпуска: 2007, Магазин: Xbox Live  
Игра: Horizon Zero Dawn | Цена: 30$, Год выпуска: 2008, Магазин: PS Network  
Игра: Fallout 76 | Цена: 20$, Год выпуска: 2010, Магазин: Bethesda Launcher  
Игра: Halo 5 | Цена: 40$, Год выпуска: 2015, Магазин: Xbox Live  
Игра: Stubbs the Zombie | Цена: 35$, Год выпуска: 2013, Магазин: Xbox Live  
Игра: Team Fortress 2 | Цена: 40$, Год выпуска: 2014, Магазин: Steam  
Игра: Half-Life | Цена: 40$, Год выпуска: 2014, Магазин: Steam  
Игра: Half-Life 2 | Цена: 40$, Год выпуска: 2005, Магазин: Steam  
Игра: Half-Life 3 | Цена: 40$, Год выпуска: 2077, Магазин: Steam  
Соединение было закрыто.
```

Вывод

В ходе выполнения работы была задана база данных H2, выполнено к ней подключение из кода на Java, создана и заполнена таблица в этой базе данных, сделаны несколько запросов с условиями к базе данных, результаты которых были выведены в консоль.

СПИСОК ИСТОЧНИКОВ

1. Основная страница H2 -
<https://www.h2database.com/html/main.html>
2. Статья о JDBC в Википедии -
https://ru.wikipedia.org/wiki/Java_Database_Connectivity
3. Статья о H2 в Википедии - <https://ru.wikipedia.org/wiki/H2>
4. Статья о Java в Википедии -
<https://ru.wikipedia.org/wiki/Java>

Приложение. Листинг.

```
package sql.game.db;

import java.sql.*;

public class GameDB {
    public static final String DB_URL = "jdbc:h2:/c:/Users/super/IdeaProjects/GameDB/db/GameDB";
    public static final String DB_Driver = "org.h2.Driver";

    public static void main(String[] args){
        try{
            Class.forName(DB_Driver);
            Connection connection = DriverManager.getConnection(DB_URL);
            System.out.println("Соединение выполнено.");
            Statement st = connection.createStatement();

            try{
                st.execute("CREATE TABLE GameDB( " +
                    "id BIGINT AUTO_INCREMENT PRIMARY KEY," +
                    "title VARCHAR(255) NOT NULL," +
                    "price INT NOT NULL," +
                    "year VARCHAR(255) NOT NULL," +
                    "primaryStore VARCHAR(255) NOT NULL," +
                    ");");
                System.out.println("Таблица успешно создана.");
            }
            catch (SQLException e){
                e.printStackTrace();
                System.out.println("Проблема при создании таблицы.");
            }
            if(!st.executeQuery("SELECT * from GameDB").next()){
                st.execute("INSERT into GameDB values(1, 'Fallout', '10', '1994', 'Bethesda Launcher');");
                st.execute("INSERT into GameDB values(2, 'Fallout 2', '10', '1998', 'Bethesda Launcher');");
                st.execute("INSERT into GameDB values(3, 'Fallout Nevada', '20', '2007', 'non');");
                st.execute("INSERT into GameDB values(4, 'Fallout New Vegas', '20', '2009', 'Steam');");
                st.execute("INSERT into GameDB values(5, 'Halo Combat Evolved', '35', '2015', 'Xbox Live');");
                st.execute("INSERT into GameDB values(6, 'Halo 2', '40', '1994', 'Xbox Live');");
                st.execute("INSERT into GameDB values(7, 'Dota 2', '25', '2001', 'Steam');");
                st.execute("INSERT into GameDB values(8, 'League of Legends', '25', '2003', 'RIOT Launcher');");
            }
        }
    }
}
```

```

        st.execute("INSERT into GameDB values(9, 'Counter-Strike Condition Zero', '25', '2005', 'Steam');");
        st.execute("INSERT into GameDB values(10, 'Call of Duty 3', '25', '2007', 'Xbox Live');");
        st.execute("INSERT into GameDB values(11, 'Horizon Zero Dawn', '30', '2008', 'PS Network');");
        st.execute("INSERT into GameDB values(12, 'The last of Us', '10', '2004', 'PS Network');");
        st.execute("INSERT into GameDB values(13, 'Fallout 76', '20', '2010', 'Bethesda Launcher');");
        st.execute("INSERT into GameDB values(14, 'Halo 5', '40', '2015', 'Xbox Live');");
        st.execute("INSERT into GameDB values(15, 'Stubbs the Zombie', '35', '2013', 'Xbox Live');");
        st.execute("INSERT into GameDB values(16, 'Battlefield 1942', '40', '2003', 'Retail');");
        st.execute("INSERT into GameDB values(17, 'Team Fortress 2', '40', '2014', 'Steam');");
        st.execute("INSERT into GameDB values(18, 'Half-Life', '40', '2014', 'Steam');");
        st.execute("INSERT into GameDB values(19, 'Half-Life 2', '40', '2005', 'Steam');");
        st.execute("INSERT into GameDB values(20, 'Half-Life 3', '40', '2077', 'Steam');");

    }

    //ResultSet rs = st.executeQuery("SELECT * from GameDB");
    ResultSet rs = st.executeQuery("SELECT * from GameDB WHERE (year > 2004)");

    while (rs.next()){
        System.out.println(
            "Игра: " + rs.getString("title") + " | Цена: " + rs.getString("price") + "$, Год выпуска: " + rs.getString("year") + ", Магазин: " + rs.getString("primaryStore")
        );
    }

    connection.close();
    System.out.println("Соединение было закрыто.");
}
catch (ClassNotFoundException e){
    e.printStackTrace();
    System.out.println("JDBC драйвер не найден!");
}
catch (SQLException e) {
    e.printStackTrace();
    System.out.println("Ошибка SQL!");
}
}}

```

