

开始

10.3 实现 JDBC 程序

通过 10.1 节和 10.2 节的学习,读者对 JDBC 及常用 API 已经有了大致的了解,下面将讲解如何使用 JDBC 的常用 API 实现一个 JDBC 程序。使用 JDBC 的常用 API 实现 JDBC 程序的步骤如图 10-2 所示。

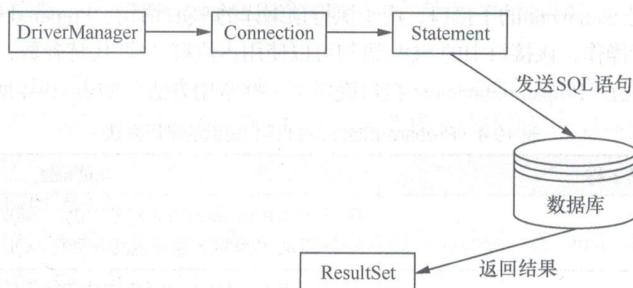


图 10-2 使用 JDBC 的常用 API 实现 JDBC 程序的步骤

下面结合图 10-2,分步骤讲解使用 JDBC 的 API 实现 JDBC 程序的过程。

1. 加载并注册数据库驱动

在连接数据库之前,要加载数据库的驱动程序到 JVM (Java 虚拟机)。加载操作可以通过 `java.lang.Class` 类的静态方法 `forName (String className)` 或 `DriverManager` 类的静态方法 `register Driver (Driver driver)` 实现。具体示例如下:

```

DriverManager.registerDriver (Driver driver);
或
Class.forName ("DriverName");

```

在实际开发中,常用第 2 种方式注册数据库驱动程序, `DriverName` 表示数据库的驱动类。以 MySQL 数据库为例,MySQL 驱动类在 6.0.2 版本之前为 `com.mysql.jdbc.Driver`,而在 6.0.2 版本之后为 `com.mysql.cj.jdbc.Driver`。要根据自己数据库版本选择相应的驱动类。

2. 通过 DriverManager 获取数据库连接

获取数据库连接的具体方式如下:

```

Connection conn = DriverManager.getConnection (String url, String user, String pwd);

```

从上述代码可以看出, `getConnection ()` 方法有 3 个参数,分别表示连接数据库的地址、登录数据库的用户名和密码。以 MySQL 数据库为例,MySQL 数据库地址的格式如下:

```

jdbc:mysql://hostname:port/databasename

```

在上面的代码中, `jdbc:mysql:` 是固定的写法; `mysql` 是指 MySQL 数据库; `hostname` 是指主机的名称 (如果数据库在本机中, `hostname` 可以为 `localhost` 或 `127.0.0.1`;如果要连接的数据库在其他计算机上, `hostname` 为所要连接计算机的 IP); `port` 是指连接数据库的端口号 (MySQL 端口号默认为 3306); `databasename` 是指 MySQL 中相应数据库的名称。

3. 通过 Connection 对象获取 Statement 对象

`Connection` 创建 `Statement` 对象的方法有以下 3 个。

- `createStatement ()`: 创建基本的 `Statement` 对象。
- `prepareStatement ()`: 创建 `PreparedStatement` 对象。
- `prepareCall ()`: 创建 `CallableStatement` 对象。

以创建基本的 `Statement` 对象为例,创建方式如下:

```

Statement stmt = conn.createStatement ();

```

4. 使用 Statement 执行 SQL 语句

所有的 `Statement` 都有以下 3 种执行 SQL 语句的方法。

- `execute ()`: 可以执行任何 SQL 语句。

- `executeQuery()`: 通常执行查询语句, 执行后返回代表结果集的 `ResultSet` 对象。
- `executeUpdate()`: 主要用于执行 DML 和 DDL 语句。执行 DML 语句, 如 INSERT、UPDATE 或 DELETE 时, 返回受 SQL 语句影响的行数; 执行 DDL 语句返回 0。

以 `executeQuery()` 方法为例, `executeQuery()` 方法调用形式如下:

```
// 执行 SQL 语句, 获取结果集 ResultSet
ResultSet rs = stmt.executeQuery(sql);
```

5. 操作 ResultSet 结果集

如果执行的 SQL 语句是查询语句, 执行结果将返回一个 `ResultSet` 对象, 该对象保存了 SQL 语句查询的结果。程序可以通过操作该 `ResultSet` 对象取出查询结果。

6. 关闭连接, 释放资源

每次操作数据库结束后都要关闭数据库连接, 释放资源, 关闭顺序和声明顺序相反。需要关闭的资源包括 `ResultSet`、`Statement` 和 `Connection` 等。

至此, JDBC 程序的大致实现步骤已经讲解完了。下面按照所讲解的步骤编写 Java 程序演示 JDBC 的使用, 该程序从 `users` 表中读取数据, 并将结果打印在控制台, 具体步骤如下所示。

(1) 搭建数据库环境

在 MySQL 中创建一个名称为 `jdbc` 的数据库, 然后在 `jdbc` 数据库中创建一个 `users` 表, 创建数据库和表的 SQL 语句如下:

```
CREATE DATABASE jdbc;
USE jdbc;
CREATE TABLE users (
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(40),
    password VARCHAR(40),
    email VARCHAR(60),
    birthday DATE
);
```

`jdbc` 数据库和 `users` 表创建成功后, 再向 `users` 表中插入 3 条数据, 插入的 SQL 语句如下:

```
INSERT INTO users (NAME,PASSWORD,email,birthday)
VALUES ('zhangs','123456','zs@sina.com','1980-12-04'),
('lisi','123456','lisi@sina.com','1981-12-04'),
('wangwu','123456','wangwu@sina.com','1979-12-04');
```

为了查看数据是否添加成功, 使用 `SELECT` 语句查询 `users` 表中的数据, 执行结果如图 10-3 所示。

(2) 创建项目环境, 导入数据库驱动

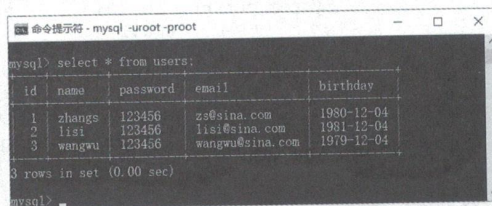
在 IDEA 中新建一个名称为 `chapter10` 的 Java 项目, 使用鼠标右键单击项目名称, 选择 “New” → “Directory”, 在弹出的窗口中将该文件夹命名为 `lib`, 项目根目录中就会出现一个名称为 `lib` 的文件夹。

将下载好的 MySQL 数据库驱动文件 `mysql-connector-java-8.0.1.jar` 复制到项目的 `lib` 目录中, 并把 jar 包添加到项目里。使用鼠标单击 `File` 菜单栏, 选择 “Project Structure” → “Modules” → “Dependencies”, 单击最右侧加号后选择第一项 “JARs or directories”, 在弹出的对话框中选择下载好的 Jar 包确认。最后可以看到 `mysql-connector-java-8.0.1.jar` 包添加到 IDEA 的依赖项中, 添加成功界面如图 10-4 所示。

在图 10-4 中, `mysql-connector-java-8.0.1.jar` 包添加到依赖项之后, 单击【Apply】按钮后再单击【OK】按钮, 可以看到在 `External Libraries` 下已经出现刚刚添加的 jar 包。至此, jar 包添加成功。加入数据库驱动后的项目结构如图 10-5 所示。

(3) 编写 JDBC 程序

在项目 `chapter10` 的 `src` 目录下, 新建一个名称为 `cn.itcast.jdbc.example` 的包, 在该包中创建类 `Example01`, `Example01` 类用于读取数据库中的 `users` 表, 并将结果输出到控制台。`Example01` 类的实现如文件 10-1 所示。



id	name	password	email	birthday
1	zhangs	123456	zs@sina.com	1980-12-04
2	lisi	123456	lisi@sina.com	1981-12-04
3	wangwu	123456	wangwu@sina.com	1979-12-04

3 rows in set (0.00 sec)

图10-3 users表中的数据