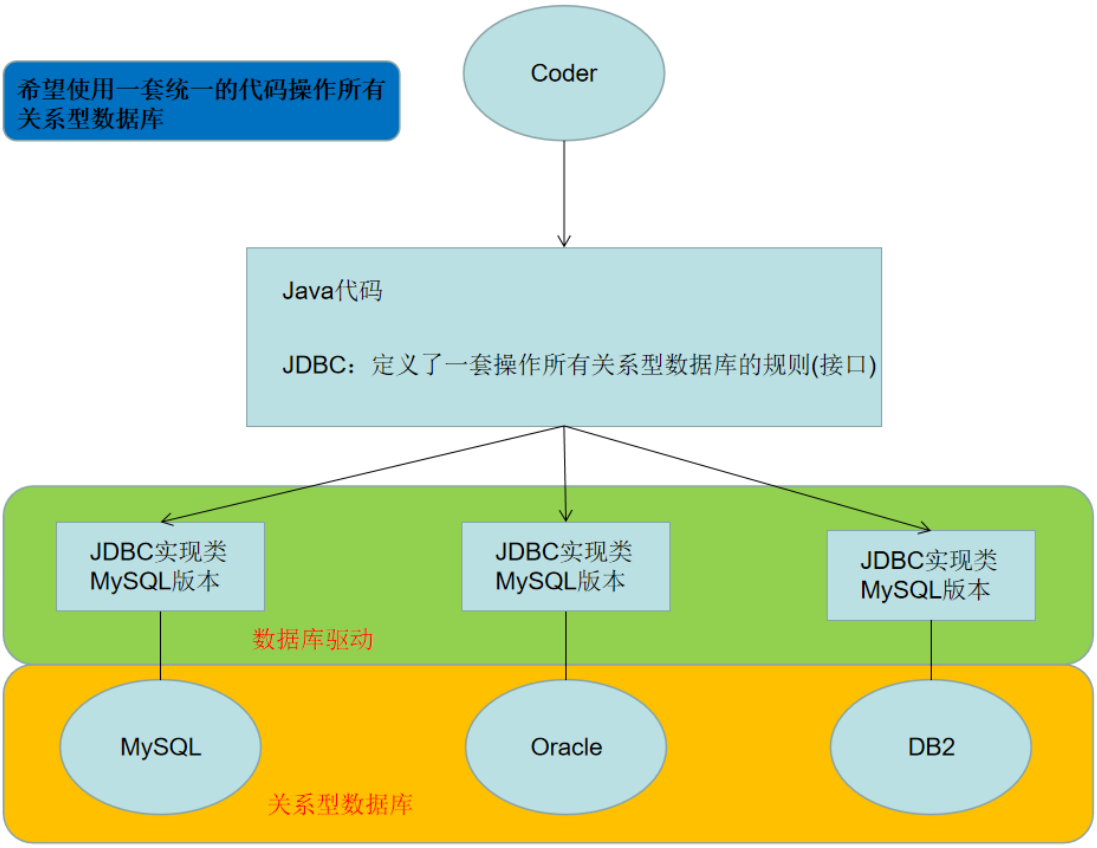
T02 JBDC

笔记本: JavaWeb阶段
创建时间: 2019/11/23 1:32 **更新时间:** 2019/11/28 21:14
作者: kwh361@gmail.com

1.概念：Java DataBase Connectivity Java数据库连接，Java语言操作数据库

- JDBC本质：其实是官方(sun公司)定义的一套操作所有关系型数据库的规则，即接口。各个数据库厂商去实现这套接口，提供数据库驱动jar包。我们可以使用这套接口(JDBC)编程，真正执行的代码是驱动jar包中的实现类。



2.快速入门:

- 步骤:
  - 1.导入驱动jar包
  - 2.注册驱动
  - 3.获取数据库连接对象Connection
  - 4.定义sql
  - 5.获取执行sql语句的对象Statement
  - 6.执行sql，接收返回的对象
  - 7.处理结果
  - 8.释放资源

练习1：JdbcDemo1.java

3.详解各个对象：

1.DriverManger：驱动管理对象

- 功能：
  - 1.注册驱动：告诉程序该使用哪一个数据库驱动jar

    static void registerDriver(Driver driver)：注册与给定的驱动程序

    写代码时使用：Class.forName("com.mysql.jdbc.Driver");
    通过查看源码发现：在com.mysql.jdbc.Driver类中存在静态代码块

```
static {
    try {
        java.sql.DriverManger.registerDriver(new Driver());
    }catch (SQLException E) {
        throw new RuntimeException("Can't register driver!");
    }
}
```

    注意:mysql5驱动jar包可以省略注册驱动的步骤

  - 2.获取数据库连接：

    *方法：static Connection getConnection(String url,String user,String password)

    *参数：

    *url：指定连接的路径

    *语法：jdbc:mysql://ip地址(域名):端口号/数据库名称

    *例子：jdbc:mysql://localhost:3306/db3

    *细节：如果连接的是本机的mysql服务器，并且mysql服务器默认端口是3306,则url可以简写为：jdbc:mysql:///数据库名称

    *user：用户名

    *password：密码

2.Connection：数据库连接对象

- 1.功能
  - 1.获取执行sql的对象
    - Statement createStatement()
    - PrepearedStatment prepareStatement(String sql)
  - 2.管理事务
    - 开启事务：setAutoCommit(boolean autoCommit):调用该方法设置参数为false，即开启事务
    - 提交事务：commit()
    - 回滚事务：rollback()

3.Statement:执行sql的对象

- 1.执行sql
  - 1.boolean execute(String sql)：可以执行任意的sql了解
  - 2.int executeUpdate(String sql)：执行DML（insert、update、delete）语句、DDL（create,alter、drop）语句
    - 返回值：影响的行数，可以通过这个影响的行数判断DML语句是否执行成功 返回值>0的则执行成功，反之，则失败。
  - 3.Result executeQuery(String sql)：执行DQL（select）语句
- 2.练习：

        - 1.添加一条记录 JdbcDemo2.java
        - 2.修改记录　　JdbcDemo3.java
        - 3.删除一条记录 JdbcDemo4.java

4.ResultSet:结果集对象，封装查询结果

- boolean next()：游标向下移动一行，判断当前行是否是最后一行末尾(是否有数据)，如果是，则返回false，如果不是则返回true;
- getXxx(参数)：获取数据
    - Xxx：代表数据类型　如：int getInt(),String getString(),Double getDouble,Date getDate()
    - 参数：
        - 1.int：代表列的编号，从1开始　如:getString(1)
        - 2.String:代表列名称。如：getDouble("balance")

- 注意：
    - 使用步骤:
        - 1.游标向下移动一行
        - 2.判断是否有数据
        - 3.获取数据

练习：JdbcDemo8.java

定义一个方法，查询emp表的数据将其封装为对象，然后装载集合，返回。

1.定义Emp类

2.定义方法public List<Emp> findAll(){}

3.实现方法select * from emp;

5.PreparedStatement：执行sql的对象

- 1.SQL注入问题：在拼接sql时，有一些sql的特殊关键字参与字符串的拼接。会造成安全性问题
    - 1.输入用户随便,输入密码：a' or 'a'='a
    - 2.sql：select * from user where username='fhsjf' and password='a' or 'a' = 'a'
- 2.解决sql注入问题：使用PreparedStatement对象类解决
- 3.预编译的SQL:参数使用?作为占位符

抽取JDBC工具类： JDBCUtils

* 目的：简化书写
* 分析：
  1. 注册驱动也抽取
  2. 抽取一个方法获取连接对象
     * 需求：不想传递参数（麻烦），还得保证工具类的通用性。
     * 解决：配置文件
       jdbc.properties
       url=
       user=
       password=
  3.抽取一个方法释放资源

* 练习：JdbcDemo9.java
  * 需求：

1. 通过键盘录入用户名和密码
2. 判断用户是否登录成功
   * select * from user where username = "" and password = "";
   * 如果这个sql有查询结果，则成功，反之，则失败

JDBC控制事务：

1. 事务：一个包含多个步骤的业务操作。如果这个业务操作被事务管理，则这多个步骤要么同时成功，要么同时失败。
2. 操作：
    1. 开启事务
    2. 提交事务
    3. 回滚事务
3. 使用Connection对象来管理事务
    * 开启事务：setAutoCommit(boolean autoCommit)：调用该方法设置参数为false，即开启事务
        * 在执行sql之前开启事务
    * 提交事务：commit()
        * 当所有sql都执行完提交事务
    * 回滚事务：rollback()
        * 在catch中回滚事务