# 1.  JDBC

JDBC Java Database Connection

是一套api规范。

各个数据库厂商实现，提供自己的驱动包

mysql-connector-java-5.1.39-bin.jar；

步骤：

1. 加载驱动

2. 获取连接

DriverManager.getConnection(url,username,password)

3. 获取PreparedStatement/Statement

4. 保存结果，ResultSet

核心类：

DriverManager

Connection

Statement

PreparedStatement

ResultSet

核心方法：

public static synchronized void registerDriver(java.sql.Driver driver)

一般不这么写，硬编码，耦合了，并且实际注册了2次驱动

DriverManager.registerDriver(new com.mysql.jdbc.Driver());

JDBC规定，在实现java.sql.Driver时，必须在静态部分完成Driver注册到DriverManager的动作。

mysql的实现：

public class Driver extends NonRegisteringDriver implements java.sql.Driver {

    static {

        try {

            java.sql.DriverManager.registerDriver(new Driver());

        } catch (SQLException E) {

            throw new RuntimeException("Can't register driver!");

        }

    }

}

所以，应该这么写：

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

JDBC规定url由3部分组成：

jdbc:${dbType}://${host}:${port}/#{dbName}

jdbc:mysql://localhost:3306/test

可以带上参数：

jdbc:mysql://localhost:3306/web08?useUnicode=true&characterEncoding=UTF8

规范的写法：

```java

@Test

public void query() {

    Connection conn = null;

    Statement stmt = null;

    ResultSet rs = null;

    try {

        conn = getConnection();

        stmt = conn.createStatement();

        String sql = "select \* from user";

        rs = stmt.executeQuery(sql);

        while(rs.next()) {

            String username = rs.getString(1);

            String password = rs.getString(2);

            System.out.println(username + ", " + password);

        }

    } catch(Exception e) {

        throw new RuntimeException(e);

    } finally {

        try {

            if(rs != null) rs.close();

            if(stmt != null) stmt.close();

            if(con != null) conn.close();

        } catch(SQLException e) {}

    }

}

```

注意关闭的顺序。

PreparedStatement

防止sql注入

性能更好

使用示例：

```java

String sql = “select \* from tab\_student where s\_number=?”;

PreparedStatement pstmt = conn.prepareStatement(sql);

pstmt.setString(1, “S\_1001”);

ResultSet rs = pstmt.executeQuery();

rs.close();

pstmt.clearParameters();

pstmt.setString(1, “S\_1002”);

rs = pstmt.executeQuery();

```

参考：

https://www.cnblogs.com/qlqwjy/p/8227665.html

# 2. 基础知识

Java注释。

1.单行注释 //

2.多行注释 /\* \*/。不可以嵌套

3.文档注释 /\*\* \*/。java特有。可以通过javadoc工具解析成html文档。不可以嵌套。 @version @author @since

Java源文件

Java源文件里可以有多个class类。

只能有一个类可以被public修饰，并且这个类名称要和源文件名称一致。不然，编译报错。

Java源文件名称和类名可以不一致，编译后的class文件名称和类名保持一致。

Java类的package包名可以和文件目录不一致（包名错误） 。编译不报错，但是字节码执行会报错，一般是找不到类。