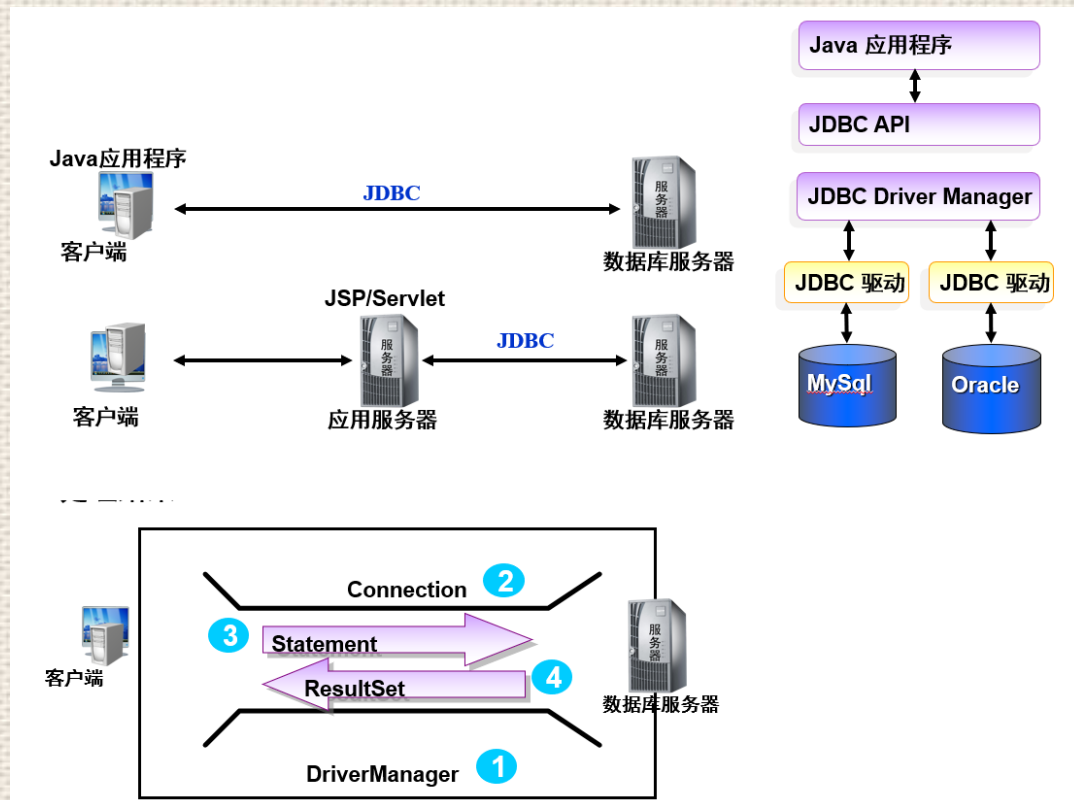


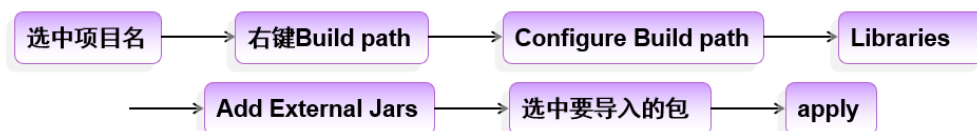
1.jdbc:java程序和数据库的连接技术.

2.jdbc工作原理

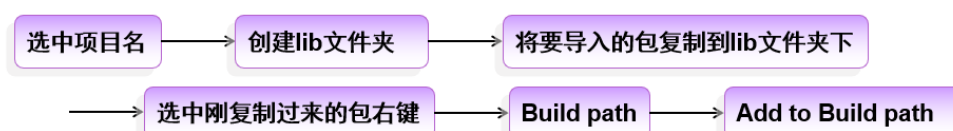


3.导入Jar包的方式

1. java工程中导包 (包还在原来的地方, 只是当前项目中运行时可用) :



2. java工程中导入包 (包就在当前项目中) :



#### 4.jdbc常用接口和类

##### DriverManager类的常用方法

方法名	说明
<i>Connection</i> <u>getConnection</u> (url, user, password)	获得连接对象

##### Connection的常用方法

方法名	说明
<i>Statement</i> <u>createStatement</u> ()	得到Statement执行对象
<i>PreparedStatement</i> <u>prepareStatement</u> (String sql)	创建一个 <i>PreparedStatement</i> 对象
void <u>close</u> ()	关闭 Connection对象
void <u>commit</u> ()	提交事务
void <u>rollback</u> ()	回滚事务
void <u>setAutoCommit</u> (boolean autoCommit)	是否手动开启事务，默认true自动开启事务

##### statement的常用方法

方法名	说 明
<i>ResultSet</i> <u>executeQuery</u> (String sql)	执行SQL查询并获取到ResultSet对象
int <u>executeUpdate</u> (String sql)	可以执行插入、删除、更新等操作，返回值是执行该操作所影响的行数
boolean <u>execute</u> (String sql)	可以执行任意SQL语句，然后获得一个布尔值，表示是否返回ResultSet
void <u>close</u> ()	关闭Statement对象
int[ ] <u>executeBatch</u> ()	执行批处理,返回影响的行数
void <u>addBatch</u> (String sql)	将sql语句添加批处理中
void <u>clearBatch</u> ()	清空批处理中Sql语句

## ResultSet的常用方法

方法名	说 明
<b>boolean next()</b>	将光标从当前位置向下移动一行
<b>boolean previous()</b>	游标从当前位置向上移动一行
<b>void close()</b>	关闭ResultSet 对象
<b>int getInt(int colIndex)</b>	以int形式获取结果集当前行指定列号值
<b>int getInt(String colLabel)</b>	以int形式获取结果集当前行指定列名值
<b>float getFloat(int colIndex)</b>	以float形式获取结果集当前行指定列号值
<b>float getFloat(String colLabel)</b>	以float形式获取结果集当前行指定列名值
<b>String getString(int colIndex)</b>	以String 形式获取结果集当前行指定列号值
<b>String getString(String colLabel)</b>	以String形式获取结果集当前行指定列名值

5.用jdbc对数据库中数据作增加,修改,删除操作

```
eg:public static void main(String[] args) throws Exception {  
    //声明连接对象和执行对象  
    Connection conn=null;  
    Statement state=null;  
  
    try {  
        //1.加载驱动  
        Class.forName("com.mysql.jdbc.Driver");  
        //2.创建连接对象  
  
        conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/myschool",  
        "root", "root");  
        //3.准备sql  
  
        //执行插入操作  
        //String sql="insert into t_student(sname,sage,sex,address,cid)  
        values('司永康',20,'男','千锋',1)";  
  
        //执行修改操作
```

```

        //String sql="update t_student set sex='女' where sid=20";

        //执行删除操作
        String sql="delete from t_student where sid=20";

        //4.创建执行对象
        state=conn.createStatement();
        //5.用执行对象调用相应的方法将Sql语句传到数据库中去执行,并
接收执行结果
        int result=state.executeUpdate(sql);
        //6.处理结果
        if (result>0) {
            System.out.println("操作成功!");
        }

    } catch (Exception e) {
        e.printStackTrace();
    }finally {
        //7.关闭连接对象,释放资源,先开后关
        if (state!=null) {
            state.close();
        }
        if (conn!=null) {
            conn.close();
        }
    }
}
}

```

#### 6.用jdbc对数据库中数据进行查询

```

eg:public static void main(String[] args) throws Exception {
    //声明连接对象和执行对象和结果集对象
    Connection conn=null;
    Statement state=null;
    ResultSet rs=null;

```



```

        try {
            //1.加载驱动
            Class.forName("com.mysql.jdbc.Driver");
            //2.创建连接对象

            conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/myschool",
            "root", "root");

            //3.准备sql
            String sql="select sid,sname,sage,sex,address,cid from
t_student";

            //4.创建执行对象
            state=conn.createStatement();
            //5.用执行对象调用相应的方法将Sql语句传到数据库中去执行,并
            接收执行结果
            rs=state.executeQuery(sql);
            //6.处理结果
            System.out.println("编号\t姓名\t年龄\t性别\t地址\t班级编号");
            while (rs.next()) {
                //读取当前行的数据,用结果集对象调用方法获得当前行指定
                列的值并存在变量
                int sid=rs.getInt("sid");
                String sname=rs.getString("sname");
                int sage=rs.getInt("sage");
                String sex=rs.getString("sex");
                String address=rs.getString("address");
                int cid=rs.getInt("cid");

                System.out.println(sid+"\t"+sname+"\t"+sage+"\t"+sex+"\t"+address+"\t"+ci
                d);
            }

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

```
    }finally {  
        //7.关闭连接对象,释放资源,先开后关  
        if (state!=null) {  
            state.close();  
        }  
        if (conn!=null) {  
            conn.close();  
        }  
    }  
}
```