# 2．系统运行环境

## 2.1 硬件环境

要求至少为 Windows XP 系统，内存至少2G。

## 2.2 软件环境

操作系统 Windows XP、Windows 10 、Windows 10 皆可。若硬件允许，建议使用Windows 7，其稳定性更优。数据库为Oracle数据库，采用Oracle据库库来存储程序运行过程中需要的数据，Oracle 数据库的版本为Oracle 11g。成效分析的系统采用的是JAVA语言，算法采用的是Matlab。编译软件采用的是MyEclipse 10，服务器采用的是Tomcat，采用的框架为Spring + Spring MVC + Mabatis。使用maven 管理项目中使用到的jar包，使用Git进行版本控制。

# 3．系统运行及初始化

## 3.1 数据库

本系统中涉及到的数据表主要有：

## 3.2 数据导入

界面上显示的数据，除了公式表的数据，其他的数据都存储在T\_CXFX\_MAIN\_CQU 中。

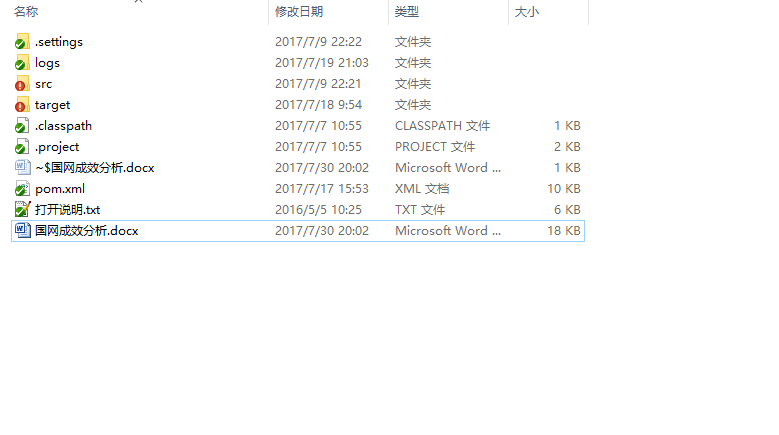
此处写数据库的建表语句，以及格式，数据来源等。

## 3.3 系统初始化

将项目导入到eclipse 或 Myeclipse 等IDE平台，修改build Path。

## 3.4 项目发布

项目的发布依赖maven ，在项目的pom.xml文件夹下，按住shift键并点击鼠标右键，点击“在此处打开命令窗口”。



图

# 4. 框架介绍

## 4.1 Spring

Spring是应用非常广泛的轻量级JAVA EE框架之一，他以Ioc、AOP为主要思想，能够协同Struts、Hibernate、JSF、MyBatis等众多的框架。

Spring 框架是一个分层架构，由 7 个定义良好的模块组成。Spring 模块构建在核心容器之上，核心容器定义了创建、配置和管理 bean 的方式，如下图所示。

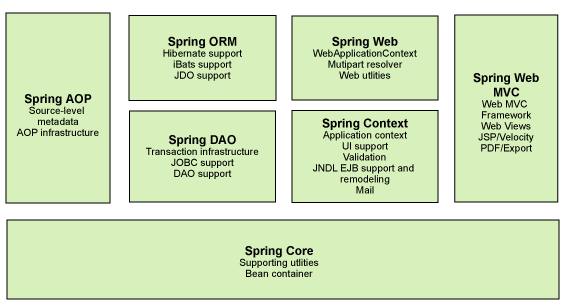


图 Spring 框架的 7 个模块

组成 Spring 框架的每个模块（或组件）都可以单独存在，或者与其他一个或多个模块联合实现。主要的模块如下：核心容器

、Spring Context、Spring AOP、Spring DAO、Spring ORM、Spring Web 模块和Spring MVC 框架。

## 4.2 Spring MVC

MVC 思想是所有面向对象设计语言都应该遵守的规范。MVC思想是将一个应用分成三个基本部分：Model（模型）、View（视图）和Controller（控制器），这三个部分以最少的耦合协同工作，从而提高应用的可维护性和可扩展性。

Spring 提供了一套自己的MVC 框架，它使用了MVC架构模式的思想，将web层进行职责解耦，基于请求驱动指的就是使用请求-响应模型， Spring Web MVC简化了Web开发。

Spring MVC主要由DispatcherServlet、处理器映射、处理器(控制器)、视图解析器、视图组成。他的两个核心是两个核心：

1. 处理器映射：选择使用哪个控制器来处理请求
2. 视图解析器：选择结果应该如何渲染

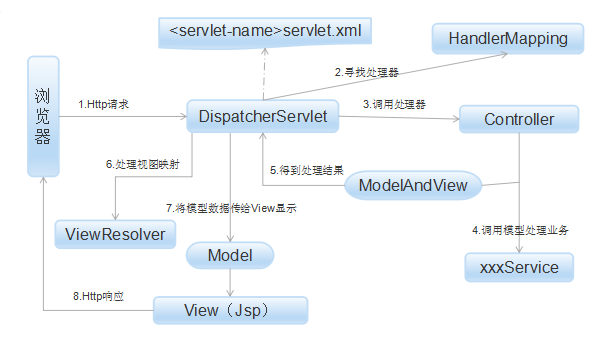


图 Spring MVC 运行原理

对于浏览器客户端的请求，Spring的解析过程如下：

1. Http请求：客户端请求提交到DispatcherServlet。
2. 寻找处理器：由DispatcherServlet控制器查询一个或多个HandlerMapping，找到处理请求的Controller。
3. 调用处理器：DispatcherServlet将请求提交到Controller。
4. 调用业务处理和返回结果：Controller调用业务逻辑处理后，返回ModelAndView。
5. 处理视图映射并返回模型：DispatcherServlet查询一个或多个ViewResoler视图解析器，找到ModelAndView指定的视图。
6. Http响应：视图负责将结果显示到客户端。

## 4.3 Mybatis

目前流行的编程语言，Java、Python等，都是面向对象语言；而目前主流的数据库依然是关系数据库。编程语言和底层数据库的发展不协调，催生了ORM框架，ORM框架可作为面向对象编程语言和数据库之间的桥梁。

ORM的全称是Object/Relation Mapping ，即对象/关系数据库映射。当使用一种面向对象的编程语言来进行开发时，从项目一开始就采用的是面向对象分析、面向对象设计、面向对象编程，但到了持久层数据库数据库访问时，又必须重回关系数据库的访问方式。于是我们需要一种工具，他可以把关系数据库包装成面向对象的模型，这个工具就是ORM。

采用ORM之后，应用程序不再直接访问底层数据库，而是以面向对象的方式来操作持久化对象（例如增删改查等），而ORM框架则将这些面向对象的操作转换成底层的SQL操作。ORM工具工作示意图如下图所示。

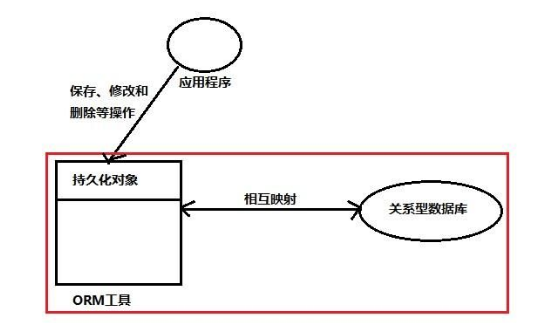


图 ORM工具工作示意图

目前流行的ORM 框架主要有：JPA、Hibernate和MyBatis。本文采用的持久层框架为MyBatis。MyBatis 是支持定制化 SQL、存储过程以及高级映射的优秀的持久层框架。MyBatis 避免了几乎所有的 JDBC 代码和手动设置参数以及获取结果集。MyBatis 可以对配置和原生Map使用简单的 XML 或注解，将接口和 Java 的 POJOs(Plain Old Java Objects,普通的 Java对象)映射成数据库中的记录。

MyBatis作为持久层框架，其主要思想是将程序中的大量SQL语句剥离出来，配置在XML文件中，以实现SQL的灵活配置。这样做的好处是将SQL与程序代码分离，做到可以在不修改程序代码的情况下，直接在配置文件中修改SQL。相对Hibernate 等“全自动”的ORM机制，MyBatis以SQL开发的工作量和数据库移植性上的让步，为系统的设计提供了更大的自由空间。

## 4.4 其他

其他包括Maven 和Git。Maven 是用于构建程序和管理项目，在本程序中，项目的主要功能是用于管理Jar包。Git是一个分布式版本控制工具，用于在项目管理项目源代码。

# 5. 界面展示

## 5.1 建设模式

## 5.2 投入成本

## 5.3 应用效益

## 5.4 计算公式

## 5.5 建设模式可视化

## 5.6 综合结论