回顾重点内容:

会话管理:管理浏览器与服务器之间会话过程产生的会话数据

Cookie技术:会话数据会保存在浏览器端

Cookie技术API:

1)创建Cookie对象，保存会话数据

Cookie cookie = new Cookie("name","value");

2)服务器发送cookie给浏览器保存(原理:响应头:set-cookie)

response.addCookie(cookie)

3)浏览器携带cookie数据访问服务器(原理:响应头:set-cookie)

4)服务器接收到浏览器发送的cookie数据

Cookie[] cookie = request.getCookie();

Session技术:

Session API:

1)创建HttpSession对象

HttpSession session = request.getSession(true/false);

(原理:发送一个JSESSION的cookie)

2)使用HttpSession对象保存会话数据

session.setAttribute("name","Object");

3)使用HttpSession对象获取UI化数据

session.getAttribute("name")

4)使用HttpSession对象删除会话数据

session.removeAttribute("name")

### 编程实战(上)

通讯录程序

需求:

添加联系人

修改联系人

删除联系人

查询联系人

职业发展:

技术线:初级开发工程师->高级开发工程师->核心开发人员(系统分析师助理)->系统分析师

非技术线:初级开发工程师->高级开发攻城狮->项目组长(项目经理助理)->项目经理

1.1需求分析(需求分析师)

a）功能需求：

需求:

添加联系人

修改联系人

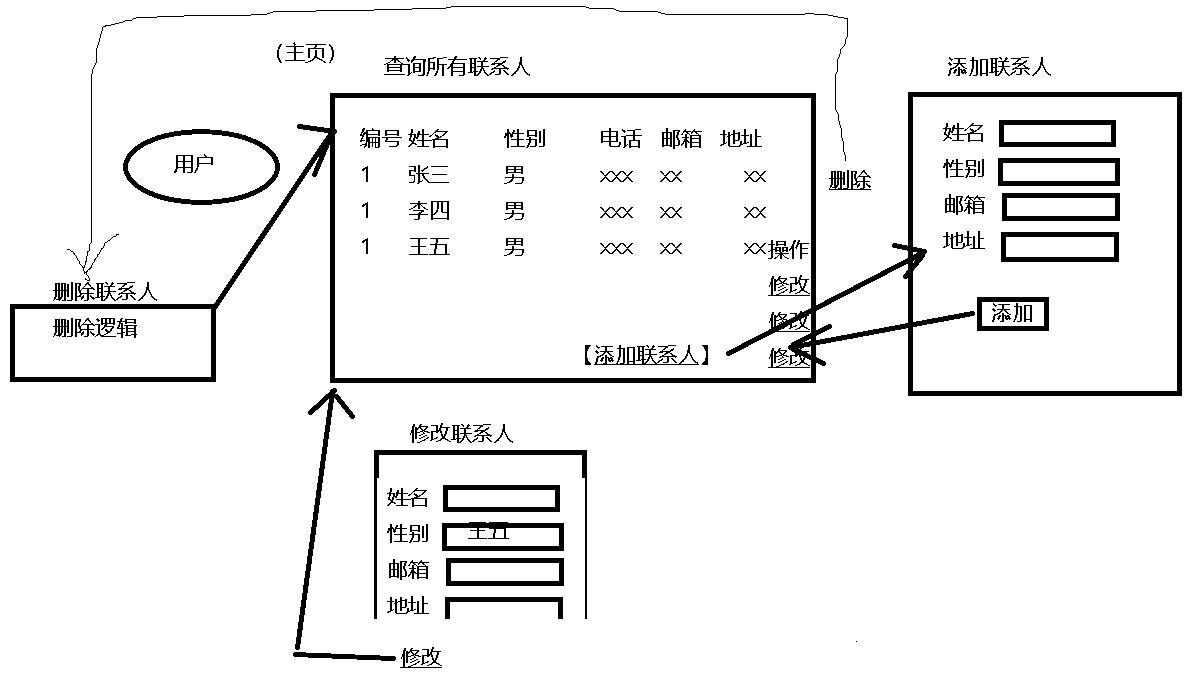
删除联系人

查询所有联系人

b）功能流程

抽取业务模型：图书模型：名称、页码、出版社

功能流程图



1.2需求设计（系统分析师、架构师）

1. 美工设计(设置静态原型)
2. 系统设计
3. 设计实体

class Contact{

String ig;

String name;

String gender;

String phone;

String email;

String address;

}

1. 设计“数据库”

Xml文件代替数据库

Contact.xml



1. 设计接口

面向接口编程:

联系人DAO类：集中编写联系人实体对象的CURD方法。（数据库访问对象）

规则：一个实体对象对应一个dao接口和方法

Interface ContactDao{

public void addContact(Contact contact);

public void updateContact(Contact contact);

public void deleteContact(String id);

public List<Contact> findAll();

public Contact findById(String id);

}

1. 设计系统目录（包）规范

|-contactSys\_web 根目录

|-src

|-gz.itcast.contactSys\_web.entity 存放实体对象类

|-gz.itcast.contactSys\_web.dao 存放dao接口

|-gz.itcast.contactSys\_web.web 存放servlet类

|-gz.itcast.contactSys\_web.util 存放工具类

|-gz.itcast.contactSys\_web.exception 自定义异常类

|-images 存放图片

|-css 存放css

|-js 存放js文件

1.3编码实现（开发攻城狮）

1. 搭建项目环境
2. 项目建立
3. 导入第三方包
4. 开发步骤:从底层往上层
5. “数据库”（设计xml）
6. 实体对象（建立实体对象）
7. DAO接口和实现类
8. 页面+servlet(跟用户直接相关)

1.4功能测试（测试攻城狮）

1.5性能测试（测试攻城狮）

1.6部署上线（实施攻城狮）

1.7系统维护（实施工程师）

补充：

1 区块选择方式---多行拼接字符串shift+alt+a 拉出一个条