**双色球趋势分析开发过程**

1. 需求
   1. 双色球趋势分析
   2. 按照开奖的持续分析；每个号码单独趋势分析
   3. 形成趋势图表
2. 分析
   1. 6个红球 1个篮球按照开奖次序，进行单独趋势分析；
   2. 获取所有双色球开奖历史数据，按原始的出奖次序排列；
3. 开发过程
   1. 设计domain;

|  |
| --- |
| * + 1. Domain代码；有很多自动生成相关 |
| @Entity  @Table(name = "double\_color\_ball")//数据库名称  @org.hibernate.annotations.Entity(dynamicUpdate = **true**, dynamicInsert = **true**)//动态更新  @Cache(usage = CacheConcurrencyStrategy.*READ\_WRITE*)//HQL 查询 缓存数据  // @Proxy(lazy=false)//因为缓存数据所以去掉此项  @Comment(value = "双色球", desc = "开奖数据")//代码生成的时候可以生成多国语言，页面的数据和校验时用到的数据  **public** **class** DoubleColorBall **extends** BaseDomain {//BaseDomain中放入了 主键、日期等公用的属性  // @Field("num\_t")//fulltext 自动solr全文检索  @Comment("期号")//生成页面的多国语言数据和js校验提示信息。  @Length(min = 4, max = 5)//自动生成页面校验  @NotEmpty//自动生成页面校验  @Index(name = "i\_num")//自动生成数据索引  @Column(name = "num\_", nullable = **false**, unique = **true**, length = 5)//nullable=false 允许为空; unique = true 生成数据库约束，必须唯一；  **private** String num;  @Comment("开奖日期")  @Index(name = "i\_lottery\_dates")  @Column(name = "lottery\_dates")  **private** Date lotteryDates;  @Comment("1")  @Length(min = 2)  @NotEmpty  @Index(name = "i\_sequence\_1")  @Column(name = "sequence\_1", nullable = **false**, length = 2)  **private** String sequence1;  @Comment("2")  @Length(min = 2)  @NotEmpty  @Index(name = "i\_sequence\_2")  @Column(name = "sequence\_2", nullable = **false**, length = 2)  **private** String sequence2;  @Comment("3")  @Length(min = 2)  @NotEmpty  @Index(name = "i\_sequence\_3")  @Column(name = "sequence\_3", nullable = **false**, length = 2)  **private** String sequence3;  @Comment("4")  @Length(min = 2)  @NotEmpty  @Index(name = "i\_sequence\_4")  @Column(name = "sequence\_4", nullable = **false**, length = 2)  **private** String sequence4;  @Comment("5")  @Length(min = 2)  @NotEmpty  @Index(name = "i\_sequence\_5")  @Column(name = "sequence\_5", nullable = **false**, length = 2)  **private** String sequence5;  @Comment("6")  @Length(min = 2)  @NotEmpty  @Index(name = "i\_sequence\_6")  @Column(name = "sequence\_6", nullable = **false**, length = 2)  **private** String sequence6;  @Comment("1")  @Length(min = 2)  @NotEmpty  @Index(name = "i\_size\_1")  @Column(name = "size\_1", nullable = **false**, length = 2)  **private** String size1;  @Comment("2")  @Length(min = 2)  @NotEmpty  @Index(name = "i\_size\_2")  @Column(name = "size\_2", nullable = **false**, length = 2)  **private** String size2;  @Comment("3")  @Length(min = 2)  @NotEmpty  @Index(name = "i\_size\_3")  @Column(name = "size\_3", nullable = **false**, length = 2)  **private** String size3;  @Comment("4")  @Length(min = 2)  @NotEmpty  @Index(name = "i\_size\_4")  @Column(name = "size\_4", nullable = **false**, length = 2)  **private** String size4;  @Comment("5")  @Length(min = 2)  @NotEmpty  @Index(name = "i\_size\_5")  @Column(name = "size\_5", nullable = **false**, length = 2)  **private** String size5;  @Comment("6")  @Length(min = 2)  @NotEmpty  @Index(name = "i\_size\_6")  @Column(name = "size\_6", nullable = **false**, length = 2)  **private** String size6;  @Comment("蓝球")  @Length(min = 2)  @NotEmpty  @Index(name = "i\_blue\_ball")  @Column(name = "blue\_ball", nullable = **false**, length = 2)  **private** String blueBall;  @Comment("兑奖号码")  @Length(min = 14)  @Index(name = "i\_red\_blue\_ball")  @Column(name = "red\_blue\_ball", nullable = **true**, length = 14)  **private** String redBlueBall;  /\*\*  \* 设定中奖号码 重构的代码；对象本身的属性和动作，离哪个对象近，就归集到哪个对象上  \*/  **public** **void** setRedBlueBall() {  **this**.redBlueBall = **this**.size1 + **this**.size2 + **this**.size3 + **this**.size4  + **this**.size5 + **this**.size6 + **this**.blueBall;   * + 1. } |

* 1. 自动生成代码

|  |
| --- |
| * + 1. org.supermy.autogen/hibernate.cfg.xml   增加<mapping class=*"com.lottery.ssq.domain.DoubleColorBall"* /> |
| * + 1. 运行 ant build.xml 中的 gen |
|  |
| Build中的gen;windows 环境下实现设定编码是utf-8，在eclipse下执行，确保编码正确；直接在dos命令窗口执行回出现乱码问题 |
| * + 1. 生成的逻辑层代码service，已经封装了常用的crud 等方法，可以直接应用；同时生成了逻辑层测试类。 |
|  |
| 生成的页面 |
|  |
|  |

* 1. 测试domain crud;

|  |
| --- |
| * + 1. 在app-springs-service.xml文件中加入   <context:component-scan base-package=*"com.lottery"*>  <context:include-filter type=*"regex"*  expression=*".service.\*"* />  <context:exclude-filter type=*"annotation"*  expression=*"org.springframework.stereotype.Repository"* />  </context:component-scan> |
| * + 1. 测试运行通过 |

* 1. 导入数据；

|  |
| --- |
| * + 1. 获取双色球的历史数据，并且整理为csv文件：src\com\lottery\ssq\domain\ source\_data.csv (用excel整理数据并且转存) |
| * + 1. 命令行文件导入 LOAD DATA INFILE ' /src/com/lottery/ssq/domain/source\_data.csv'  INTO TABLE double\_color\_ball  指定分隔符号为 , FIELDS TERMINATED BY ','  制定文本文件中字段的出现次序 (num\_,sequence\_1, sequence\_2, sequence\_3, sequence\_4, sequence\_5, sequence\_6,size\_1, size\_2, size\_3, size\_4, size\_5, size\_6, blue\_ball,lottery\_dates);     2. 更改默认数据生效 update double\_color\_ball set enabled\_=1;     3. 初始化某些数据字段 update double\_color\_ball set red\_blue\_ball=concat(size\_1,size\_2,size\_3,size\_4,size\_5,size\_6,blue\_ball); |
| * + 1. 测试导入数据 DoubleColorBallServiceTest |

* 1. 编辑逻辑，并且进行测试

|  |
| --- |
| * + 1. DoubleColorBallService |
| **public** List<Map<String, String>> trendline(Integer location, Integer limit) {  String columnname = "";  **if** (location.intValue() == 7) {// 蓝色球  columnname = "blueBall";  } **else** {  columnname = "Sequence" + location;  }  String ballquery = "select obj." + columnname  + " from DoubleColorBall obj order by obj.id asc";  String countquery = "select count(obj.id) from DoubleColorBall obj";  Long total = (Long) doubleColorBallUtil.findUnique(countquery);  List<String> findForProperty = doubleColorBallUtil.findForProperty(  ballquery, total.intValue(), limit);  List<Map<String, String>> result = **new** ArrayList<Map<String, String>>(  limit);  **for** (String obj : findForProperty) {  Map<String, String> line = **new** TreeMap<String, String>();  **for** (**int** i = 0; i <= 32; i++) {  **int** jj = i + 1;  String jkey = jj <= 9 ? "0" + jj : jj + "";  **if** (Integer.*parseInt*(obj) == jj) {  line.put("ball" + jkey, obj + "");  } **else** {  line.put("ball" + jkey, "");  }  }  result.add(line);  }  **return** result;  } |
| * + 1. DoubleColorBallServiceTest 增加方法进行测试 |
| * + 1. 测试通过进行web层面的开发 |

* 1. WEB层面

|  |
| --- |
| * + 1. 拷贝Action代码到制定的目录 com.lottery.ssq.web.ssq |
| * + 1. 拷贝生成的页面到指的的目录 web\WEB-INF\content |
|  |
|  |
| * + 1. 启动tomcat;打开浏览器，进入：<http://127.0.0.1:8080/fastweb/ssq/double-color-ball.action>；已经能够访问了。 |
| * + 1. 根据需要调整页面，同步增加逻辑和action方法； |

1. 定制

|  |
| --- |
| * 1. 引入amcharts; |
| * 1. 需要的amcharts数据拷贝到\web\flash\amline\_1.6.4.1目录下 |
| 建立文件\web\flash\amline\_1.6.4.1\examples\chart\_with\_scroller\index.jsp 需要增加数据源：so.addVariable("data\_file", encodeURIComponent("/fastweb/ssq/double-color-ball!showtrendlinexml.action?location=${param.location}&limit=${param.limit}"));   * 1. DoubleColorBallAction类中增加方法showtrendlinexml ;详见以下代码 |
| **public** **static** **final** String *TRENDLINEXML* = "trendlinexml";  **public** List<Map<String, Object>> trendlinexmls = **new** ArrayList<Map<String, Object>>(  30);  **public** List<Map<String, Object>> getTrendlinexml() {  **return** trendlinexmls;  }  /\*\*  \* 趋势线分析 xml  \*/  **public** String showtrendlinexml() **throws** Exception {  // double-color-ball!showtrendlinexml.action?lcation=1&limit=20  String location = Struts2Utils.*getRequest*().getParameter("location");  String limit = Struts2Utils.*getRequest*().getParameter("limit");  trendlinexmls = doubleColorBallService.trendlinexml(  Integer.*valueOf*(location), Integer.*valueOf*(limit));  **return** DoubleColorBallAction.*TRENDLINEXML*;  } |
| * 1. DoubleColorBallServiceTest 中增加测试方法，测试以上方法； |
| * 1. 增加页面 \web\WEB-INF\content\ssq\double-color-ball-trendlinexml.jsp,为flash chart提供数据 |
| <%@ page contentType="text/html;charset=UTF-8"%>  <%@ page import="org.supermy.core.security.SecurityUtils" %>  <%@ taglib prefix="s" uri="/struts-tags" %>  <?xml version="1.0" encoding="UTF-8"?>  <chart>  <series>  <s:iterator value="trendlinexml" status='st'>  <value xid="${st.index}"><s:date name="year" format="yyyy-MM-dd"/></value>  </s:iterator>  </series>  <graphs>  <graph gid="1">  <s:iterator value="trendlinexml" status='st'>  <value xid="${st.index}">${ball}</value>  </s:iterator>  </graphs>  </chart> |
| * 1. 浏览器中测试以上数据页面 |
| * 1. 测试web\flash\amline\_1.6.4.1\examples\chart\_with\_scroller\index.jsp页面 |
| * 1. 将以上页面整合到\web\WEB-INF\content\ssq\double-color-ball.jsp列表页面中 |
|  |

1. 数据维护功能调试

|  |
| --- |
| * 1. 双色球管理权限数据初始化 DbServiceTest 中增加 |
| //双色球数据维护  auth2Role(manager, authManager, guestRole, DoubleColorBall.**class**.getSimpleName()  .toUpperCase(), "/ssq/double-color-ball"); |
| * 1. 运行 DbServiceTest 方法 |
| 3.启动tomcat ,进行页面crud的测试与修改； |
|  |

1. 完成

|  |
| --- |
| * 1. 默认列表页面 |
|  |
| * 1. 默认编辑页面 |
|  |
| * 1. 定制趋势分析页面（此种方法可以应用到报表） |
|  |
|  |

1. other