# 1.GoodsSearch模块设计

## 1.1概述

### 1.1.1承担的需求（主要秘密）

根据用户输入的关键字与数据库中的商品信息进行快速匹配，返回搜索的结果

本模块需要对输入的关键字进行预处理，将一些无用的词汇删除，如英语的a、an、the，中文的了，啊，呢，吧，哦等等

本模块需要将输入的关键词在同义词表中找出相对应的同义词，并根据同义词进行搜索，得到检索列表

涉及需求：

R2 系统高可靠性 在用户输入有误的时候能够及时反馈，并且仍然能够正常运行

R11 浏览器兼容性 兼容Firefox版本号>=3.5;兼容Chrome版本号>=3;兼容IE版本号>=6;兼容Opera版本号>=10.5;兼容Safari版本号>=3.1

R25 高效性搜索 搜索响应时间不超过2s

R27 搜索结果排序 允许使用竞价规则定制排序策略

### 1.1.2可能会修改的实现（次要秘密）

同义词匹配算法

检索匹配算法

输入词汇预处理算法

### 1.1.3角色

检索功能

### 1.1.4对外接口

## 1.2类图

## 1.3类描述

（类方法（重要方法要给出前置与后置条件）与重要数据结构描述）

## 1.4重要协作

（顺序图与协作描述）

## 1.5使用的设计模式

（使用场景；所用的设计模式；要达到的效果）

# 2.GoodsSort模块设计

## 2.1概述

### 2.1.1承担的需求（主要秘密）

用户输入关键字并向系统提交查询请求后，系统在查询结果页面根据如下的排序规则对结果进行排序并展示给用户：

根据用户输入的关键词的相关度进行排序，关键字越吻合，相关度越大，排名越靠前

根据来源的商城的可靠度进行排序，商城知名度越大，关键字相关度一致的条件下排名越靠前

允许使用竞价排序规则调整排序

涉及需求：

R27 搜索结果排序 允许使用竞价规则定制排序策略

商品排序用例图：



用例描述：

1. 计算相关度

GoodsSort模块根据传入的关键词和商品列表计算商品列表的每一项与关键词的匹配程度

1. 计算可靠来源

GoodsSort模块根据每一项商品信息的来源计算其来源的可靠程度

1. 竞价规则

GoodsSort模块根据竞价规则对商品信息列表进行排序

### 2.1.2可能会修改的实现（次要秘密）

排序算法

竞价规则排序策略

### 2.1.3角色

计算相关度：根据所获得的关键词与关键词列表进行匹配，通过特定的算法计算获得的关键词与关键词列表中每一项的信息的相关度

计算来源可靠度：对与满足相关度要求的商品信息，根据网站来源等信息来计算可靠度

竞价排序：根据竞价规则对得分的各部分进行加权，获得最后的排序列表

### 2.1.4对外接口

与GoodsSearch模块之间的接口:

Public List<Goods> getSort(List<Goods> goodslist,String keyword);

与数据库之间的接口:

Public List fetchSourceList();

## 2.2类图



## 2.3类描述

### 2.3.1 SortFilter类

该类为GoodsSort模块和其他模块交互的主入口，调用各个排序规则以实现搜索结果的排序

类方法

|  |  |
| --- | --- |
| 类方法 | 描述 |
| Public List<Goods> getSort(List<Goods> goodslist,String keyword); | 概述：本方法为外界交互的接口，用于获取排序后的商品信息  前置条件： keyword合法,goodslist不为空  后置条件： 获得排序后的商品信息列表 |

数据结构描述

|  |  |
| --- | --- |
| 数据结构 | 描述 |
| List sortRules | SortRules集合 |

### 2.3.2 SortRules接口

本接口用于封装进行排序的计算规则，具体算法由子类实现，包括Relevancy，ReliableSource和Bidsort

### 2.3.3 Relevancy类

该类计算商品信息列表与关键字的匹配程度并进行排序

类方法

|  |  |
| --- | --- |
| 类方法 | 描述 |
| Public List<Goods> sort(List<Goods> goodslist,String keyword); | 概述：根据getRelevancy(Goods good) 计算出的相关度进行排序  前置条件： keyword合法，goodslist不为空  后置条件： 获得排序后的商品信息列表 |
| Private double getRelevancy(Goods god) | 将商品信息与关键字进行比较，计算出相关度 |

数据结构描述

|  |  |
| --- | --- |
| 数据结构 | 描述 |
| String keywords | 用户输入的搜索关键字 |

### 2.3.4 ReliableSource类

该类用于计算商品来源的可靠度

类方法

|  |  |
| --- | --- |
| 类方法 | 描述 |
| Public List<Goods> sort(List<Goods> goodslist,String keyword); | 概述：根据getSourceScore(String url)计算得到的来源可靠度对商品信息列表进行排序并返回  前置条件： goodslist不为空  后置条件： 获得排序后的商品信息列表 |
| Private double getSourceScore(String url) | 根据url的来源计算商品的可靠度 |
| Public List fetchSourceList() | 从数据库中获取sourceList |

数据结构描述

|  |  |
| --- | --- |
| 数据结构 | 描述 |
| List sourcetList | 来源网站及可靠度信息列表 |

### 2.3.5 BidSort类

该类用于根据竞价规则对商品列表进行排序

类方法

|  |  |
| --- | --- |
| 类方法 | 描述 |
| Public List<Goods> sort(List<Goods> goodslist,String keyword); | 概述：根据竞价规则bidRules对商品列表进行排序  前置条件： goodslist不为空  后置条件： 获得排序后的商品信息列表 |

数据结构描述

|  |  |
| --- | --- |
| 数据结构 | 描述 |
| BidRules bidRules | 所选择的竞价规则 |

### 2.3.6 BidRules接口

本接口用于封装竞价规则，具体算法由子类实现，目前有ConcreteRules

### 2.3.7 ConcreteRules类

该类为一个具体的竞价规则的实现类，根据具体的竞价规则对商品列表进行竞价排序

类方法

|  |  |
| --- | --- |
| 类方法 | 描述 |
| Public List<Goods> sortByBid(List<Goods> goodslist); | 概述：根据制定的竞价规则商品列表进行排序  前置条件： goodslist不为空  后置条件： 返回根据竞价规则排序后的商品信息列表 |

## 2.4重要协作



协作描述

商品搜索模块从数据库获取到商品信息列表后，将该列表传入本模块，有sortFilter对商品列表进行排序。SortFileter先通过Relevancy对商品和关键词进行相关度的匹配并根据相关度大小进行排序；再调用ReliableSource对商品信息的来源进行进行可靠性排序；后通过BidSort对商品信息列表进行根据竞价规则的排序。

## 2.5使用的设计模式

### 2.5.1策略模式

**使用场景**

1) 商品排序使用的策略可能发生变化

2) 商品的竞价规则可能发生变化

**在类图中的体现**

****

****

**要达到的效果**

策略模式可以在不修改原有排序策略和竞价规则的前提下增加新的策略（规则），并使得这些策略可以相互替换