1. Crawler模块

1.1 概述

**承担的需求（主要秘密）：**

该模块主要承担的任务是从其他网站上抓（爬）取商品价格信息，抓（爬）取策略的选择和配置以及周期性抓取新数据以保证数据有效性。

模块需要实现自动抓（爬）取指定网页的功能，初步过滤掉无效信息，然后将商品相关的信息交给数据处理模块处理。对一个网页进行数据抓（爬）取时，将页面中的有效链接放入到一个URL抓（爬）取队列中，在当前网页抓（爬）取完毕后，从URL抓（爬）取队列中选取下一个URL作为抓（爬）取对象。以此操作，直到URL抓（爬）取队列为空。

模块需要实现抓（爬）取策略可变更的功能，能够根据爬虫管理员的选择来采取相应的网页抓（爬）取策略。抓（爬）取策略存储和配置存储在配置文件中。当系统初始化或者抓（爬）取策略变更时，爬虫类从配置文件中读取选定的抓（爬）取策略，然后开始网站抓（爬）取工作。

模块需要实现一个可配置的定时器来完成周期性抓（爬）取的功能。定时器的各项配置保存在配置文件中。爬虫管理员需要提前设定好定时器的周期。在开启爬虫程序后，一旦定时器达到指定周期，就会触发爬虫去抓（爬）取网站信息。在爬虫程序运行过程中，爬虫管理员可以随时改变定时器的周期，然后定时器的配置文件会做出相应的更改，然后在适当的时间（即爬虫程序空闲时）重启爬虫程序来使配置生效。

**可能会修改的实现（次要秘密）：**

增加新的网站抓（爬）取源时，需要增加新的网页内容抓（爬）取方式。

增加新的网站抓（爬）取策略时，需要增加新的抓（爬）取策略实现。

**角色：**

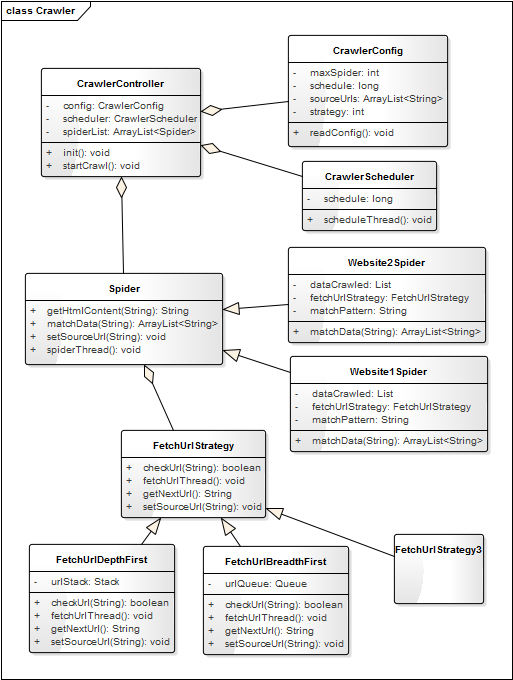
在整个系统中扮演商品数据信息获取的角色，获取的数据交给商品兼容处理模块。

**对外接口：**

1）void analyseData();（参数为从网站爬取的原始商品信息）

提供给商品数据兼容处理模块，目的是让其能够获取从网站上爬取的原始商品信息。

1.2 类图



* 1. 类描述

**CrawlerController类：**爬虫模块的集中控制类，包括爬虫配置文件读取、爬虫定时器的启动、初始化爬虫和开始爬虫程序。类中维护了一个爬虫列表。

**Init():void----**爬虫程序初始化方法，通过CrawlerConfig读取配置文件来获取一些运行时的基本参数。

**startCrawl():void----**爬虫程序的启动方法，里面根据config里的基本参数来新建爬虫对象，开启爬虫线程。

**CrawlerConfig类：**爬虫模块配置类，负责读取配置文件中的爬虫配置，从而获得爬虫程序的启动参数。具体参数包括爬取网站源、爬取周期、爬取策略、最大爬虫数量等。

**readConfig():void----**读取配置文件，获得基本的爬虫参数。

**CrawlerScheduler类：**爬虫模块的定时器类，负责周期性启动爬虫程序。

**scheduleThread():void----**开启定时器的计时线程，当一个周期到时，能够重新开启爬虫程序来爬取数据。

**Spider类：**爬虫模块的抓取数据类，负责根据给定的URL抓取网页的信息，然后从中提取需要的商品信息。它是具体网站信息抓取类Website1Spider、Website2Spider类的父类，包含了爬虫类的统一属性和方法。

**setSourceUrl(String):void----**设置待抓取网站的源URL，给URL爬取类使用。

**spiderThread():void----**启动爬虫页面爬取类的线程。

**getHtmlContent(String):String----**根据URL获取网页的html内容，作为返回值返回，供后面提取商品信息用。

**matchData(String):ArrayList<String>----**根据匹配规则筛选出抓取页面中的商品信息。

**WebSite1Spider类：**是适用于具体网站Website1的爬虫类。

**matchData(String):ArrayList<String>----**根据适用于Website1的信息匹配规则，提取出页面中的商品信息。

其他方法与父类中相同。

**Website2Spider类：**是适用于具体网站Website2的爬虫类。

**matchData(String):ArrayList<String>----**据适用于Website2的信息匹配规则，提取出页面中的商品信息。

其他方法与父类中相同。

**FetchUrlStrategy类：**爬虫模块的URL爬取类，负责爬取给定网站所包含的所有URL，并且对URL进行去重。爬取到的URL提供给爬虫类进行网页数据的抓取。

其中的方法为抽象方法，供具体策略类实现。

**FetchUrlDepthFirst类：**URL爬取类的具体实现，运用了深度优先遍历的URL爬取策略。

**setSourceUrl(String)----**设置待抓取网站的源URL，作为初始URL爬取对象。

**fetchUrlThread():void----**开始URL爬取线程，爬取的URL放入URL栈中。

**checkUrl(String):Boolean----**检查爬取的URL是否重复，不重复则加入URL栈中。

**getNextUrl():String----**爬虫类从URL栈中获取下一条可抓取信息的URL。

**FetchUrlBreadthFirst类：**URL爬取类的具体实现，运用了广度优先遍历的URL爬取策略。

**setSourceUrl(String)----** 设置待抓取网站的源URL，作为初始URL爬取对象。

**fetchUrlThread():void----**开始URL爬取线程，爬取的URL放入URL队列中。

**checkUrl(String):Boolean----**检查爬取的URL是否重复，不重复则加入URL队列中。

**getNextUrl():String----**爬虫类从URL队列中获取下一条可抓取信息的URL。

**FetchUrlStrategy3类：**URL爬取类的具体实现，运用了其他的URL爬取策略（Strategy3）。

* 1. 重要协作

**顺序图**

**协作描述**

* 1. 使用的设计模式

**使用场景**

**所用设计模式**

**要达到的效果**

1. CrawlerDataUpdate模块

2.1 概述

**承担的需求（主要秘密）：**

该模块主要承担的任务是将规范化后的商品数据增量更新到数据库中。由于商品信息中不同的数据改变频率不同（比如商品描述变化少，商品价格、数量变化频繁），该模块会对不同的部分采取不同的更新策略，以此来提高数据更新的效率。

**可能会修改的实现（次要秘密）：**

可能会新增不同的数据更新方式和策略。

可能会修改数据库表结构的设计。

**角色：**

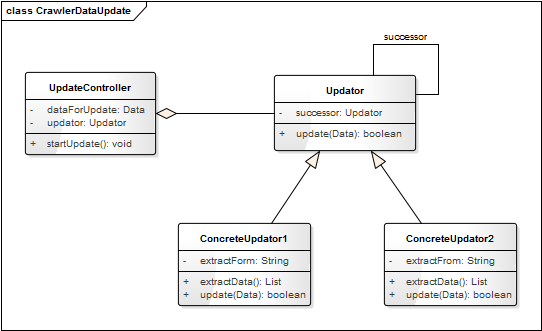
在爬虫部分中扮演数据更新的角色，负责将商品兼容模块处理后的数据更新进数据库。

**对外接口：**

1）void giveFormattedData();（参数为分析处理后的商品数据）

提供给商品数据兼容处理模块，目的是获取处理后的商品数据，从而更新到数据库中。

2.2 类图



* 1. 类描述

**UpdateController类：**爬虫数据更新模块的集中控制类，其中包括分析处理过后的规范化数据和一个更新器。

**startUpdate():void----**开始数据的更新。

**Updator类：**爬虫数据更新模块的更新器，负责对数据进行更新。里面维护了一个Updator对象，如果还有指向的更新器，则继续向下传递数据进行更新。

**update(Data):boolean**----数据更新方法。

**ConcreteUpdator1类：**更新器类的具体实现，从接收到的数据中提取自己负责更新的部分，然后按照给定策略进行更新。

**extractData():List**----按照extractForm里的规则，从接收到的数据中提取自身类负责更新的部分。

**update(Data):boolean----**重写Updator类里的方法，按照给定的策略进行更新。

**ConcreteUpdator2类：**同ConcreteUpdator1类，只是其负责更新的数据部分不同。即其extractForm里提取数据的规则不同，update(Data):boolean里的数据更新策略有不一样的实现。

* 1. 重要协作

**顺序图**

**协作描述**

* 1. 使用的设计模式

**使用场景**

**所用设计模式**

**要达到的效果**