|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 姓名 | 学号 | Github账号 | 分工 | 成绩 |
| 李文韬 | 201592417 | liwentao201592417 | 组长 |  |
| 魏振伟 | 201592165 | Damugeisme | 组员 |  |
| 胡正超 | 201592382 | HZCcarrot | 组员 |  |
| 李宏达 | 201592181 | tonylihongda | 组员 |  |
| 王佳华 | 201592110 | 18742011168 | 组员 |  |

**开源软件基础课程报告**

|  |  |
| --- | --- |
| 报告题目 | 京东全站数据爬虫 |
| 项目网址 | https://github.com/Damugeisme/JDSpider |
| 完成日期 | 2018年1月11日 |

大连理工大学软件学院

目 录

[1 生命周期模型 1](#_Toc420487560)

[1.1 模型选择 1](#_Toc420487561)

[1.2 模型介绍 1](#_Toc420487563)

[1.3 选择原因 2](#_Toc420487563)

[2 需求分析 2](#_Toc420487560)

[2.1 从京东网站爬取商品信息 2](#_Toc420487561)

[2.2 将爬取信息进行处理 2](#_Toc420487561)

[2.3 大概功能结构图 2](#_Toc420487561)

3 设计

3.1 总体设计 3

3.2 详细设计 [5](#_Toc420487563)

4 实现与测试7

# 生命周期模型

1. 模型选择

经过分析我们最终决定使用瀑布模型。

1. 模型介绍

1. 阶段间具有顺序性和依赖性

•前一阶段完成后才能开始后一阶段的工作

•前一阶段输出作为后一阶段输入

•只有前一阶段有正确输出时，后一阶段才有可能正确结果

2. 推迟实现的观点

•可避免急于求成而导致的返工

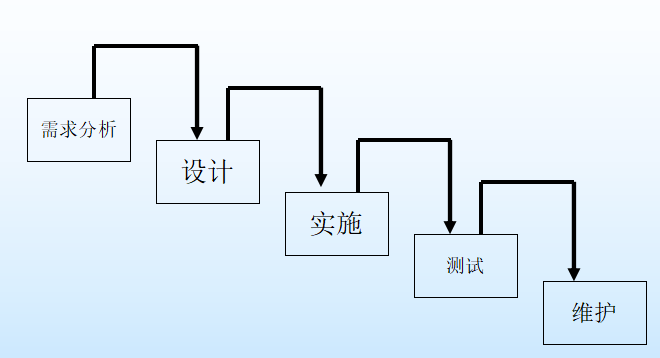
•将系统分析与逻辑设计同物理实现清楚分开开来

•尽可能推迟程序的物理实现

3. 质量保证的观点

•每个阶段都必须完成和提交规定的文档，否则则认为该阶段任务未完成

•每个阶段结束前都要对所完成的文档进行严格评审，以尽早发现问题，纠正错误



1. 选择原因

因为在项目开始之前有明确的需求（参照生活经历以及网上查询）；解决实施方案方法明确；是一种短期项目；且不存在太大变动、不确定性；风险较小；对系统安全性要求不高；有成功的案例可以参考。瀑布模型提供了一种规范，使得分析、设计、编码、测试与维护工作可以在该规范的指导下有序地展开，避免了软件开发、维护过程中的随意状态。所以最终选定使用瀑布模型。

# 需求分析

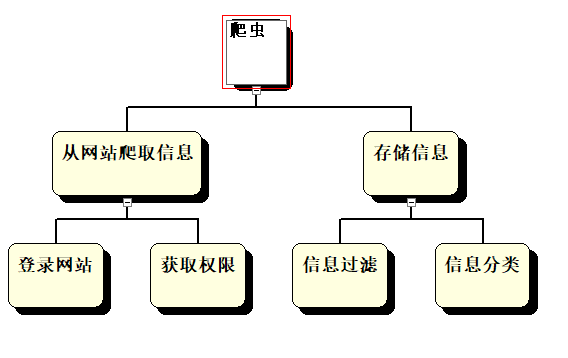
1. 从京东网站爬取商品信息

本次项目是应用python知识实现一个简单的爬虫，实现从网上爬取数据，并保存到本地，然后应用数据分析，过滤的方法，过滤分析数据，尽可能使数据清晰，容易分辨。程序应能进入京东网站，获取网站权限，了解哪些是可爬取信息，哪些是不可爬取信息。获取商品分类，按照分类获取商品基本信息，包括：商品名称，商品价格，优惠，简单介绍等等。爬取过程中，注意网页跳转情况，进行网址转换继续爬取信息，以及账号信息，维护网页Cookie，避免多次登录。考虑各种可能发生情况，如:cookie失效，账号被限制，等可能导致登录或爬取失败的事件。注意爬取过程不要违法相关法律法规，不能盗用他人隐私信息。

1. 将爬取信息进行处理

首先将数据保存到本地数据库，注意分析数据信息种类，如商铺，产品等等，分类存取。尽量能存储成text文件最好。利用数据分析，过滤实现。首先创建一个总类，用于存储各个类的信息，包括：分类名称，分类url，分类id，分类的index等等。然后创建一系列得类，存储不同信息，如，产品类（ProductsItem），包含信息：产品名称，产品url，产品id，产品价格，产品描述，shop id等等；店铺类（ShopItem），包括信息：店铺名称，shop id，等等；管理类，将商品与店铺链接，同一店铺的商品与相同店铺建立联系；店铺汇总类，将店铺汇信息总到一起，一同展示。

1. 大概功能结构图



三、 总体设计

1 对象模型

CategoriesItem

Name

url

\_id

index

JDSpider

Name

Allowed-domains

Start\_urls

Parse\_category()

Parse\_list()

Parse\_product()

Parse\_comments()

ProdeucsItem

Name

url

\_id

Category

reallyPrice

originalPrice

description

shopId

commentCount

goodComment

generalComment

poolComment

favourableDesc1

favourableDesc2

CommentImageItem

\_id

associateId

productId

imgUrl

available

pin

dealt

imgTitle

isMain

2 功能模型

使用

使用

扩展

使用

使用者

使用

使用

四、 详细设计

1 获取分类页

F

F

T

T

F

T

获取下一组

获取下一组

获取下一组

新增分类页

添加

item.split 是关键词

item in items

获取items

text in texts

开始

与列表中重复

初始化

2 获得商品地址和下一页地址

开始

初始化

获取下一组

获取地址

text in texts

T

F

3 商品页获取

开始

Category初始化

获取ids

T

if not ids

F

vender\_id和shop\_id获取

shop信息获取

结束

4 获取商品

开始

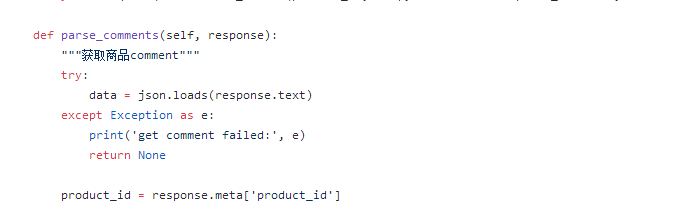
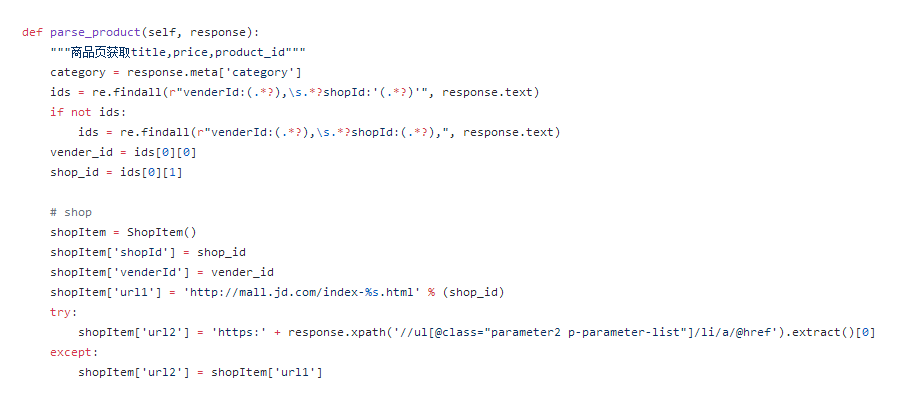
product\_id获取

结束

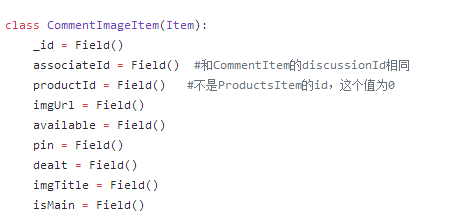
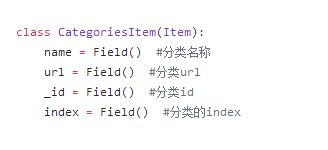
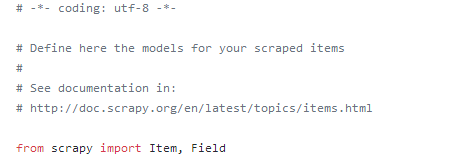
# 实现与测试

1. 实现代码

爬取信息部分：



信息分类存储部分代码：



1. 测试运行

运行部分截图：

