# 1、系统概述

**1.1、系统简介**

软件系统的名称：网络爬虫软件

背景：“爬虫”多用于从网页中爬取大量数据，为数据分析等操作提供了便利。但对于零编程基础的人来说，选择一款好用的爬虫软件，可以提高工作效率，达到事半功倍的效果。本软件实现功能就是如此。本软件产品是一项独立的软件，而且全部内容自含。

简介：本爬虫软件是一个自动提取网页的程序。用户可以自行输入想要爬取的关键词、下载图片数量、下载地址，点击搜索即自行下载相应数量的关键词图片到指定文件夹。

**1.2、系统目标**

总的目标是编写一个快捷，稳定的爬虫软件，编写并改善简易爬虫代码并在其基础上进行创新与增添。分开来讲有代码规范的编写，BUG的修复，任意指定爬取对象功能的添加与完善，任意指定下载数量功能的添加与完善，任意指定下载路径功能的添加与完善。

**1.3、系统运行环境**

操作系统：WINDOWS XP，WINDOWS 7，WINDOWS 8

编程平台：Spyder、Anaconda

**1.4、开发环境**

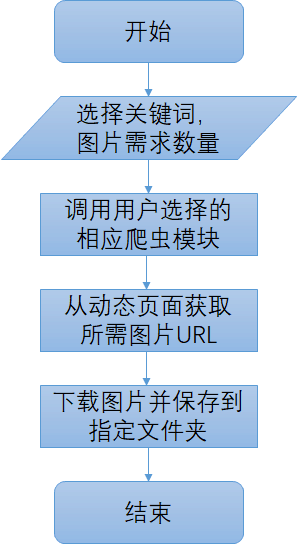
开发语言：Python

工具软件：Spyder3.3.3、Anaconda3-4.4.0

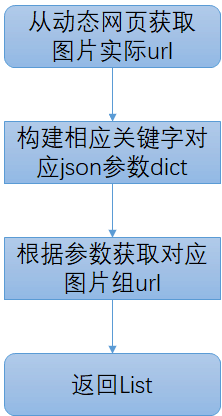
# 2、总体结构设计

**2.1、软件结构**

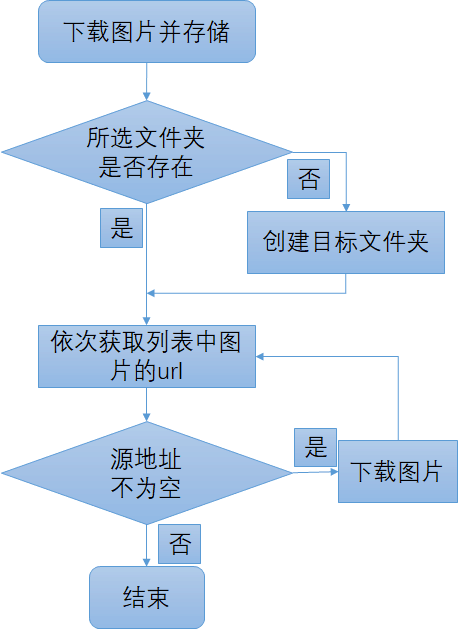
1. 爬虫主模块：



1. 输入解析模块：
2. 网页源码解析模块：



1. 图片下载解析模块：



各模块之间的调用关系

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A  A对B的调用接口  B | 爬虫主模块 | 输入解析模块 | 网页源码解析模块 | 图片下载模块 |
| 爬虫主模块 |  |  |  |  |
| 输入解析模块 | name.get()  name1.get() |  | keyword.get() | name.get()  name1.get() |
| 网页源码解析模块 | browser |  |  | browser |
| 图片下载模块 | download |  |  |  |

各模块之间的松散耦合关系

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 爬虫主模块 |  |  |  |  |
| 输入解析模块 | 内容耦合 |  |  |  |
| 网页源码解析模块 | 内容耦合 | 数据耦合 |  |  |
| 图片下载模块 | 内容耦合 | 数据耦合 | 非直接耦合 |  |
|  | 爬虫主模块 | 输入解析模块 | 网页源码解析模块 | 图片下载模块 |

**2.2、设计思想**

本网络爬虫是一个自动提取网页的程序，根据输入关键词配置代码中的网页链接并继续访问其它的页面，并将其下载直至满足用户的需求数量。

1. 确定爬取对象、下载图片数量、下载文件地址文件名；
2. 从动态网页获取图片URL，通过递归算法实现下一URL的访问，重复以上步骤；
3. 依次获取列表中图片的URL下载图片，并存储在目标文件夹中。

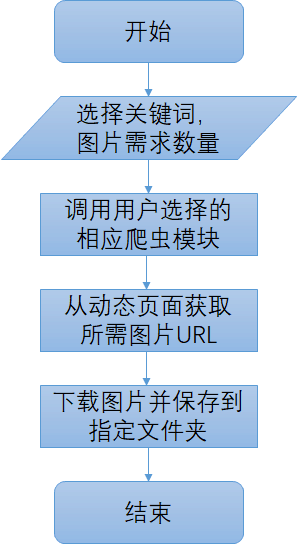
总体来说，程序通过输入配置从动态网页获取初始URL，不断构造对下一URL的访问并同时下载目标图片，直到达到下载数量。

# 3、模块设计

|  |  |  |  |
| --- | --- | --- | --- |
| 模块名称 | 功能 | 输入 | 输出 |
| 爬虫主模块 | 爬取关键词图片并下载目标文件夹中 | 所需图片的关键词和数目 | 爬去并命名完成的图片 |
| 输入解析模块 | 解析并连接到所需的页面 | 所需图片的关键词 | 定位到指定页面 |
| 网页源码解析模块 | 依次爬去每一张图片url | 所需图片数目 | 图片的url |
| 图片下载模块 | 下载图片 | 图片url，存储图片的路径 | 输出命名完成的图片到指定目录下 |

**3.1、爬虫主模块**

3.1.1、功能描述



3.1.2、接口描述

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 接口名称 | 输入信息 | 输出信息 | 异常处理 |
| 输入解析模块 | name.get()  name1.get()  keyword.get() | 下载图片数量  下载文件名 | 赋值到指定变量 | 不符合输入规则报错 |
| 网页源码解析模块 | browser | 起始url | 递归下一url | 无 |
| 图片下载模块 | download | 第一张图片下载链接 | 下一图片下载链接 | 无 |

3.1.3、数据结构描述

无

3.1.4、实现思路

实现用户输入关键词，进而进行相关图片爬取并下载所需数量的图片。在设计时将主模块划分成三个子模块:输入解析模块、网页源码解析模块、图片下载模块。子模块功能在下面具体介绍。

3.1.5、其他

爬虫主模块以应用程序的形式表现，人机交互界面设计友好简洁。

**3.2、输入解析模块**

3.2.1、功能描述

3.2.2、接口描述

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 接口名称 | 输入信息 | 输出信息 | 异常处理 |
| 图片下载模块 | name.get()  name1.get() | 下载图片数量  下载文件名 | 赋值到指定变量 | 不符合输入规则报错 |
| 网页源码解析模块 | keyword.get() | 起始url | 递归下一url | 无 |

3.2.3、数据结构描述

无

3.2.4、实现思路

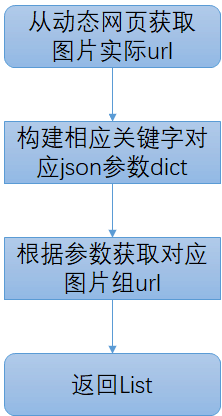
用Tkinter来进行网格布局设计用户界面。

3.2.5、其他

输入模块设计以服务程序为表现形式，人机交互友好简洁。

**3.3、网页源码解析模块**

3.3.1、功能描述



3.3.2、接口描述

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 接口名称 | 输入信息 | 输出信息 | 异常处理 |
| 输入解析模块 | keyword.get() | 爬取对象 | 赋值到指定变量 | 不符合输入规则报错 |

3.3.3、数据结构描述

无

3.3.4、实现思路

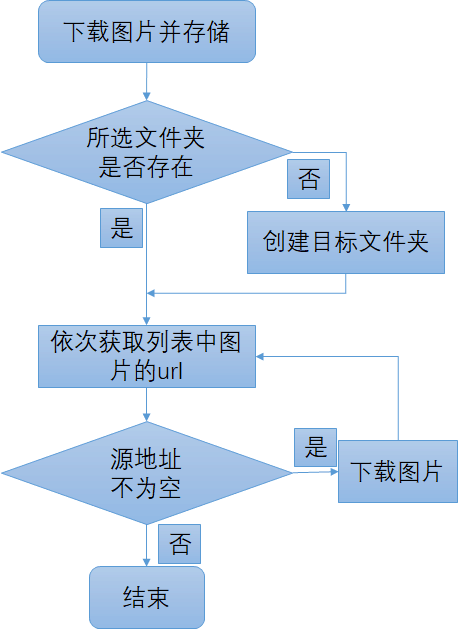
使用自动化测试工具获取中央大图，class为currentImage，的图片原url，变量名称为src，并且传值给下载模块进行下载，然后使用自动化测试工具找到翻到下一张图片的按钮，name为"img-next"，并且进行点击。

3.3.5、其他

网页源码解析模块以服务程序的形式运行。

**3.4、图片下载解析模块**

3.4.1、功能描述



3.4.2、接口描述

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 接口名称 | 输入信息 | 输出信息 | 异常处理 |
| 输入解析模块 | name.get()  name1.get() | 下载图片数量  下载文件名 | 赋值到指定变量 | 不符合输入规则报错 |
| 网页源码解析模块 | browser | 起始url | 递归下一url | 无 |

3.4.3、数据结构描述

无

3.4.4、实现思路

获取下载路径之后使用os.file选择下载路径，获取图片url并下载到这个目录中。同时按照一定规律命名进行存储。

3.4.5、其他

图片下载解析模块以服务程序的形式运行。

# 4、数据库与数据结构设计

**4.1、 数据库及数据表**

不以数据库方式存储数据。

**4.2、 数据结构设计**

无

**4.3、 数据存储设计**

1.爬取了所需要进行爬取图片操作的页面，存到了变量url中，然后再打开一个页面，直接访问url；

2.使用tkinter.filediolog.askdirectory来选取下来路径；

3.编写函数download()使用os.path来管理下载的路径。

# 5、接口设计

**5.1、外部接口**

无

**5.2、内部接口**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A  A对B的调用接口  B | 爬虫主模块 | 输入解析模块 | 网页源码解析模块 | 图片下载模块 |
| 爬虫主模块 |  |  |  |  |
| 输入解析模块 | name.get()  name1.get() |  | keyword.get() | name.get()  name1.get() |
| 网页源码解析模块 | browser |  |  | browser |
| 图片下载模块 | download |  |  |  |

# 6、其他设计

1、输入的图片数量异常的情况下，例如输入的数字为小数，进行警告和正确操作提示；

2、选择的文件目录下载在被更改成无效的文件目录并进行爬取时，进行警告和正确操作提示。