程序开发说明

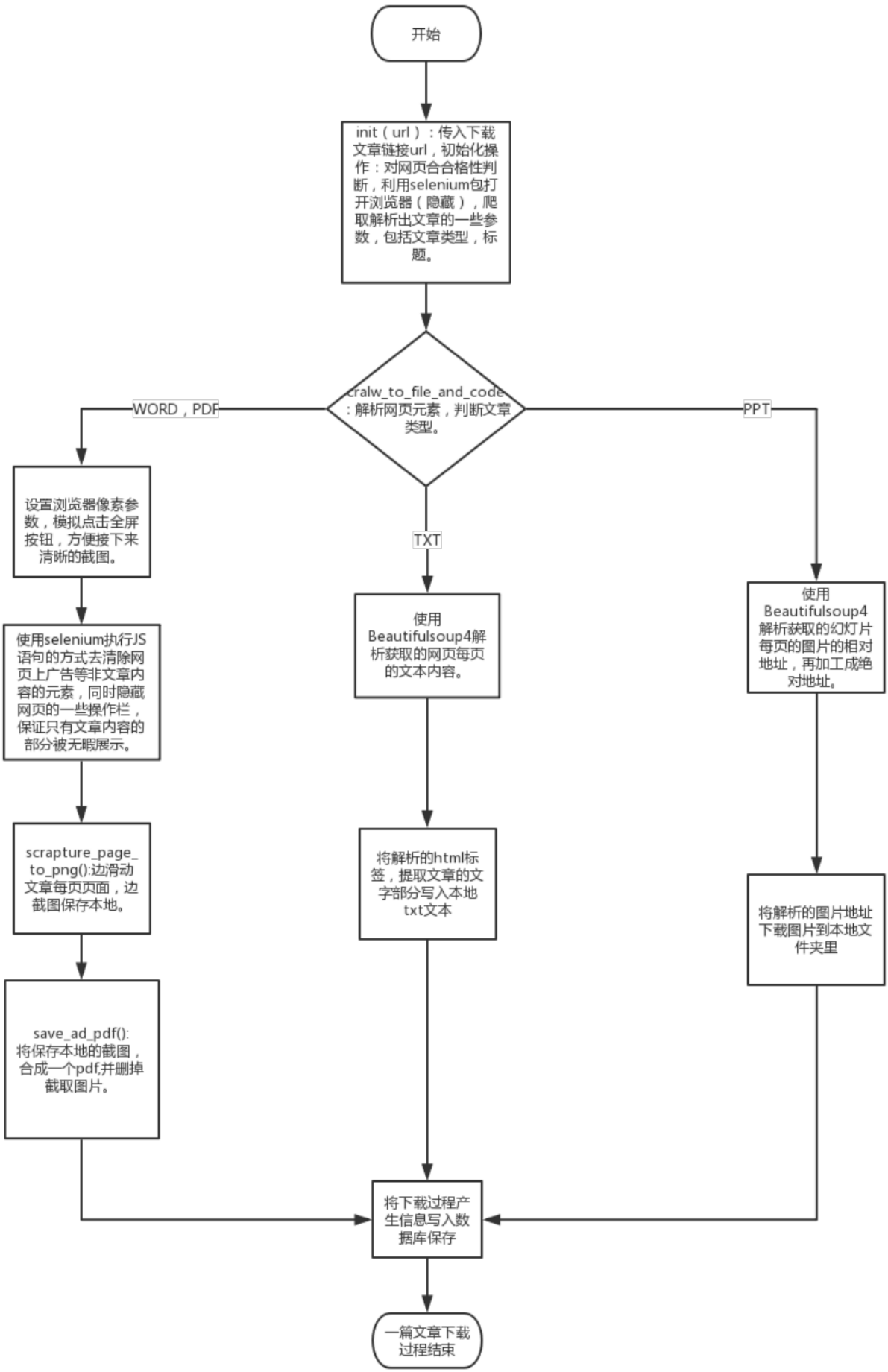
实现需要技术：

- 对计算机网络HTTP协议理解，使用python的requests库解析服务器返回response的各种资源（html,json等）。
- 使用测试工具selenium模拟对浏览器各种操作。
- 利用测试工具selenium与解析库Beautifulsoup的CSS选择器方式对网页元素的定位与解析。
- python多线程程序的编写 。
- 调用reportlab库将多张图片合成一个pdf文件。
- python与sqlites3数据库操作。

实现主要难点：

- 对百度文库不同类型文件的下载策略的分析（pdf，txt，ppt文件的解析方式都不相同）：
- 选择selenium + firefox截图方案，并配置环境
- selenium操作浏览器的一些细节（为了要截图时干净，要对网页除文章以外广告等元素隐藏，删除，更新等）
- 百度动态加载每一页内容，需要边滑动浏览器，边截图当前页面。
- 当前浏览器截图时浏览器大小等参数需要调试，使截出图片高清无瑕疵。
- python多线程程序的实现

下载程序流程图


开始
init（url）：传入下载文章链接url，初始化操作：对网页合合格性判断，利用selenium包打开浏览器（隐藏），爬取解析出文章的一些参数，包括文章类型，标题。
cralw_to_file_and_code：解析网页元素，判断文章类型。
WORD，PDF
TXT
PPT
设置浏览器像素参数，模拟点击全屏按钮，方便接下来清晰的截图。
使用selenium执行JS语句的方式去清除网页上广告等非文章内容的元素，同时隐藏网页的一些操作栏，保证只有文章内容的部分被无暇展示。
scrapture_page_to_png():边滑动文章每页页面，边截图保存本地。
save_ad_pdf():将保存本地的截图，合成一个pdf,并删掉截取图片。
使用Beautifulsoup4解析获取的网页每页的文本内容。
将解析的html标签，提取文章的文字部分写入本地txt文本
使用Beautifulsoup4解析获取的幻灯片每页的图片的相对地址，再加工成绝对地址。
将解析的图片地址下载图片到本地文件夹里
将下载过程产生信息写入数据库保存
一篇文章下载过程结束

每个函数的参数及作用详见后期开放的代码

由于此程序是某人毕业设计，毕业后再考虑开源部分代码

开发过程（待补充）

实现的过程参考资料工具：

- python3.6+selenium+firefox环境配置
- 利用 Python + Selenium 实现页面截图
- selenium 常见元素定位方法和操作
- Beautifulsoup4.04 文档
- pyhton正则表达式指南
- 在线测试正则表达式
- python3.6程序打包exe
- 使用python显示当前系统中的所有进程并关闭某一进程
- Mongo文档