**上市公司名录及信息收集v1.2**

**进度计划：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Index | Function’s name | coding | unittest | Due |
| system | | | |  |
| 1 | System flow |  |  | 2/10 |
| module | | | |  |
| 1 | Initialization |  |  | 2/8 |
| 2 | Collect\_info |  |  | 2/9 |
| 3 | output |  |  | 2/8 |
| detail | | | |  |
| 1 | Create\_class | Done | Done | 2/3 |
| 2 | Collect\_detail\_flow |  |  | 2/5 |
| 3 | Check\_stock | Done | Done | 2/3 |
| 4 | Collect\_short\_names | Done | Done | 2/4 |
| 5 | Collect\_brief |  |  | 2/4 |
| 6 | Collect\_finance |  |  | 2/4 |
| 7 | Build\_csv |  |  | 2/6 |
| 8 | Write\_to\_csv |  |  | 2/6 |
| 9 | Exception\_report |  |  | 2/7 |

|  |  |
| --- | --- |
| 整体设计 | 上市公司公开信息搜集 |
| 目的 | 1. 统计名录 2. 搜集有价值信息 3. 帮助联络 |
| 功能 | 1. 自动批量爬取上市公司公开信息 2. 根据需要做标签或分类，以便进一步处理 3. 输出格式便于处理 4. 输出信息便于处理 5. 提供可拓展接口 6. 正确处理异常 7. 可监控处理过程 |
| 输入 | 1. 供爬取网站URL 2. 预制标签分类 |
| 输出 | 1. 信息表格（.csv） 2. 错误报告 |
| 流程 | 1. Initialization(import, definition, etc..) 2. Function definition(self-defined modules, functions) 3. Main(flow control) |
| NOTE | ***监控信息*** |

|  |  |
| --- | --- |
| 系统设计 | System flow |
| 功能 | 1. handle主流程 2. 合理、正确调用功能模块 |
| 输入 | 1. 网站main URL 2. 标签分类子function |
| 输出 | 1. 信息表格（.csv） 2. 错误报告 |
| 流程 | 1. Initialization    * Import modules    * Define global variables    * create\_class 2. Collect\_info    * Check\_stock    * Collect\_detail\_flow      + Collect\_short\_name      + Collect\_brief      + Collect\_finance 3. output    * Build\_csv    * Write\_to\_csv    * Exception\_report |
| NOTE | Collect\_detail\_flow需要预留接口，可以补充其他需要的模块  Exception\_report作为单独模块还是分配在各个模块中实现？ |

|  |  |
| --- | --- |
| 模块1 | 初始化（initialization） |
| 功能 | 1. 导入模块 2. 初始化设定 3. 定义全局变量 4. 定义存放类 |
| 输入 | NA |
| 输出 | 全局变量 |
| 流程 | 1. Utf编码定义 2. Import module 3. Define global variable 4. Define class（create\_class） |
| NOTE |  |

|  |  |
| --- | --- |
| 模块2 | 爬取信息(collect\_info) |
| 功能 | 1. 爬取公司信息，包括“公司概况”和“财务分析” 2. 以类存放 |
| 输入 | url |
| 输出 | Class |
| 流程 | 1. Check\_stock 2. Collect\_detail\_flow    1. Collect\_short\_name    2. Collect\_brief    3. Collect\_finance |
| NOTE |  |

|  |  |
| --- | --- |
| 模块3 | 输出(output) |
| 功能 | 1. 将类信息放入.csv 2. 输出错误报告 |
| 输入 | 1. Class 2. Key\_url\_list |
| 输出 | 1. Company info(.csv) 2. Report(.txt) |
| 流程 | 1. Build\_csv 2. Write\_to\_csv 3. Exception\_report |
| NOTE |  |

|  |  |
| --- | --- |
| 细节1 | 创建类（create\_class） |
| 功能 | 创建类，存放每个公司信息 |
| 输入 | NA |
| 输出 | Class |
| 流程 | 1. 初始化 2. 定义变量 3. 定义函数（可选） |
| NOTE | 需要的信息如下：   1. 代码 2. 简称 3. 公司名称 4. 联系电话 5. 邮箱 6. 行业（东财分类） 7. 行业（证监会分类） 8. 区域 9. 雇员 10. 营业总收入 11. 归属净利润 |
|  |  |

|  |  |
| --- | --- |
| 细节2 | 爬取企业详细信息流程控制（collect\_detail\_flow） |
| 功能 | 控制爬取详细企业信息的流程  检测股票的正确性  当前的信息包括“公司概况”和“财务分析”  预留接口提供给其他详细信息 |
| 输入 | url |
| 输出 | Name\_list |
| 流程 | 1. Build list 2. Stock list [url] 3. Stock list [html] 4. Single stock [url] 5. Single stock [html] 6. Check\_stock 7. Collect\_short\_names 8. Read key\_url\_list 9. url to html 10. collect\_brief 11. collect\_finance 12. define API 13. write name to name\_list |
| NOTE | ***\*\*\*企业简称创建一个list，帮助在写入CSV的时候控制写入*** |

|  |  |
| --- | --- |
| 细节3 | 确认股票正确性（check\_stock） |
| 功能 | 判断该页面信息是否正确 |
| 输入 | html |
| 输出 | True or False |
| 流程 | 1. Check validity 2. True or False |
| NOTE | *Check validity* |

|  |  |
| --- | --- |
| 细节4 | 爬取企业简称名录（collect\_short\_names） |
| 功能 | 爬取企业简称及股票代码  爬取“公司概况”和“财务分析”的 URL |
| 输入 | html（single stock） |
| 输出 | Class variable  Key\_url\_list（存“公司概况”及“财务分析”URL） |
| 流程 | 1. Collect brief name 2. Collect stock code 3. Write to class 4. 爬取“公司概况”URL 5. 爬取“财务分析”URL 6. URL存为列表（key\_url\_list） |
| NOTE | 信息如下：   1. 代码 2. 简称   *不同的股票展现出来的“概况”和“财务”结构不一样，需要统一爬取的方法，或使用不同方法爬取。*  *Collect\_detail\_flow中预留的其他信息接口需要collect\_short\_name模块中的支持，预留爬取其他信息URL的接口* |

|  |  |
| --- | --- |
| 细节5 | 爬取企业概况（collect\_brief） |
| 功能 | 爬取企业概况信息 |
| 输入 | url |
| 输出 | Class variable |
| 流程 | 1. Turn to html 2. read html 3. find company name 4. find telephone 5. find email 6. find industries class TYPE 1 7. find industries class TYPE2 8. find location 9. find # of employee 10. write to class |
| NOTE | 信息如下：   1. 公司名称 2. 联系电话 3. 邮箱 4. 行业（东财分类） 5. 行业（证监会分类） 6. 区域 7. 雇员   无法获取限定信息，直接获取所有显示信息  Tips：find(id = ‘Table0’) |

|  |  |
| --- | --- |
| 细节6 | 爬取企业财务（collect\_finance） |
| 功能 | 1. 爬取企业财务状况 2. 按年度统计 3. 爬取营业总收入 4. 爬取归属净利润 |
| 输入 | url |
| 输出 | Class variable |
| 流程 | 1. Turn to html 2. Read html 3. Adapt to annual data 4. Collect total income 5. Collect revenue 6. Write to class |
| NOTE | 信息如下：   1. 年度营业收入 2. 年度利润   *\*初始统计数据按报告期统计，需要调整为年度统计后再爬取记录*  *\*\*\*需要模拟点击操作*  *Tips：import selenium* |

|  |  |
| --- | --- |
| 细节7 | 创建表格文档（build\_csv） |
| 功能 | 生成CSV文档 |
| 输入 | NA |
| 输出 | CSV（empty） |
| 流程 | NA |
| NOTE | 定义 utf\_8\_sig，避免乱码 |

|  |  |
| --- | --- |
| 细节8 | 写入文档（write\_to\_csv） |
| 功能 | 把搜集的class写入csv |
| 输入 | * Name\_list * Class info |
| 输出 | Csv |
| 流程 | 1. Count to 100 (write every 100 data was collected) 2. Write to csv |
| NOTE | *Counter计数器需要提前定义并且计数* |

|  |  |
| --- | --- |
| 细节9 | 打印错误（exception\_report） |
| 功能 | 打印错误信息 |
| 输入 | NA |
| 输出 | 错误报告（.txt） |
| 流程 |  |
| NOTE | 可能出现错误的种类：   1. 访问错误（网络超时） 2. 访问错误（网页刷新） |

|  |  |
| --- | --- |
| 单元测试 |  |
| 用例 |  |
| 问题 |  |
| 修正 |  |

|  |  |
| --- | --- |
| 系统测试 |  |
| 用例 |  |
| 问题 |  |
| 修正 |  |