

## BaseCatcher 部分

作者——张鸣昊

设计说明：

该部分的框架较为清晰，对于每个功能有四个文件：.cpp,.h,.py,.txt，前两个是与整体框架连接的基类和派生类，py 文件是功能真正实现的文件，txt 文件用来实现 C++

与 python 的简易交互，属于缓存文件。BaseCatcher 有 12 个派生类，每个派生类有 4 个文件，大多数功能相似。当 Strategy 中调用成员函数 MakeCatcher 时，py 工作并将信息保存于 txt 中，当 Strategy 中调用 SaveinObject 时，C++ 读入文件内容并清空，以便下次使用。之后的整体信息交给 Strategy 处理，分割成个体信息，也就是数据基类，然后储存到 BaseObject 中的 vector 里。

文件解释：

功能派生类为网站名\_by\_目标名.格式（例如 IMDB\_by\_people.cpp, Douabn\_by\_movies.py），每个网站有 4 种目标，每个目标包括四个相关文件。结构大体一致，用户可以选择某一种查看（py 差异大，因为功能有别）。

注意事项：

由于网页格式多变，加上本程序与其他组电影信息略有不同——本程序类似搜索引擎，主要用于即时爬取网站信息，所以难免有难以处理的问题。

1. 经分析，豆瓣网站上关于演员信息的搜索，无法用 <https://www.douban.com/search?q=> 获取（这种方法只能获取用户），而真正用于搜索演员、导演等信息的结果界面中，相关信息全是 JavaScript，无法获取链接，故我们无法像另外两个网站一样，通过直接点击演员名字来即时爬取演员信息。作者将 Douban\_by\_people.py、Douban\_by\_people.cpp、Douban\_by\_people.h 代码一并上传，但这些代码并不能与整体协调搭配起作用，只能用于另外的使用。

2. 对于 IMDB 中的 getallphoto 功能，实际上是可以爬取几百张相关的照片，而且经分析，只有 IMDB 较为适配这个功能，而且经过多次测试，该功能会花大量的时间（真的大量，几分钟），并不适合在 qt 上使用。所以，我们把此功能单独拿出来，如果用户希望获取大量的 IMDB 电影照片，可以单独使用“获取某部影片大量照片”的功能，然后根据提示在指定文件夹中查找（qt 界面显示过多照片不够美观，且不利于下载）。

3. 对于排行榜功能，由于排行榜中电影数量巨大，信息多耗时长（IMDB 的最慢），且一些有意义的信息作者也非常希望能保存下来，所以，我们的排行榜功能也不在 qt 界面上展示，对于该信息我们也计划在 txt 中显示，其中，IMDB 榜单功能较慢，但能收集统计出在 100 年里，分布在各个时间区间的电影数量统计，以一张图标的形式展示，过程也较慢。

4. 由于 RottenTomatoes 网站的搜索功能不佳（搜索之后得到的链接有许多乱码，无法直接当做网址使用），作者的设计目前只能接受相当标准的查找（而 IMDB 的搜索可以支持关键字）我们会提示用户标准输入格式，如果格式不匹配，搜索将会失败。

5.由于作者能力有限，并没有实现多线程等高级功能，用户体验方面因情况而已。但是总体上可爬取的信息种类全，几乎涵盖了网页页面上能显示的大多数信息。且本程序定位为一款搜索引擎形式的软件，Catcher 基本完成了爬取的功能。