**作业1:爬虫**

姓名：白植权 学号：221050020

1.作业简介

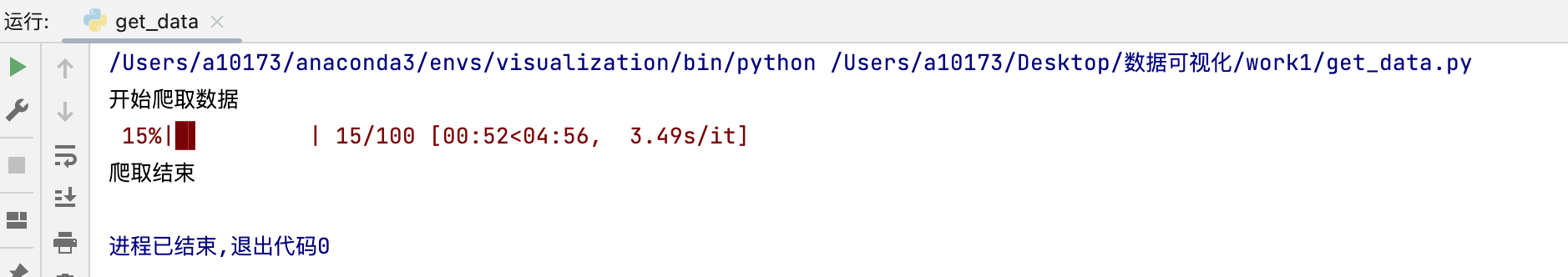
本作业完成的是一个微博数据爬虫，它的主要功能是爬取微博上关于特定话题（如“研究生” “杭州” “周杰伦”）的微博内容。代码会创建一个CSV文件，用于存储爬取的数据，包括用户名和对应的微博内容。代码在预设的最大迭代次数（这里设为100次）内，构建并访问微博的搜索页URL，这个URL根据特定话题进行构建。每次访问页面时，爬虫会获取该页的微博数据，将其中的用户名和评论内容提取出来，并保存到CSV文件中。

2.爬虫Python部分代码：

|  |
| --- |
| *# encoding=gbk* import requests import pandas as pd import time import re from tqdm import trange topic = "大学生" epochs = 100 def get\_data(): data2 = pd.DataFrame([("id", "comment")])  data2.to\_csv('%s.csv' % topic, header=False, index=False) *# 导出数据到{topic}.csv  #. . . 获取网页header和cookie   .  .  .*  print("开始爬取数据")  *# 爬取数据* for ii in trange(epochs):  url\_base = . . . *# url冗长,在此省略*  url = url\_base + str(ii + 1)   json = requests.get(url, headers=header, cookies=Cookie).json()  try:  if len(json['data']['cards']) == 0:  print("爬取结束")  return  for jj in range(len(json['data']['cards'])):text = re.sub(r'<.\*?>', '', json['data']['cards'][jj]['mblog']['text'])  *# 去英文数字和部分符号* data1 = [(json['data']['cards'][jj]['mblog']['user']['screen\_name'], text)]  data2 = pd.DataFrame(data1)  data2.to\_csv('%s.csv' % topic, header=False, index=False, mode='a+')  *# 导出数据到{topic}.csv* except:  print("第%d页抓取失败" % ii)  time.sleep(3)  *# 防止高频连续爬取被禁止* if \_\_name\_\_ == "\_\_main\_\_":  get\_data() |

3.运行结果：

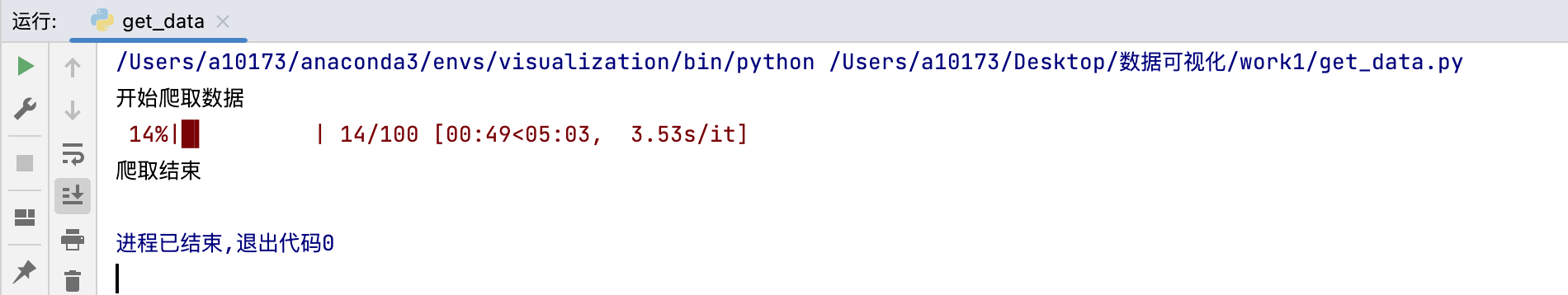
**（1）主题为“大学生”时：**

****

爬取资料部分截图如下：（此为2023年5月25日爬取数据）：

****

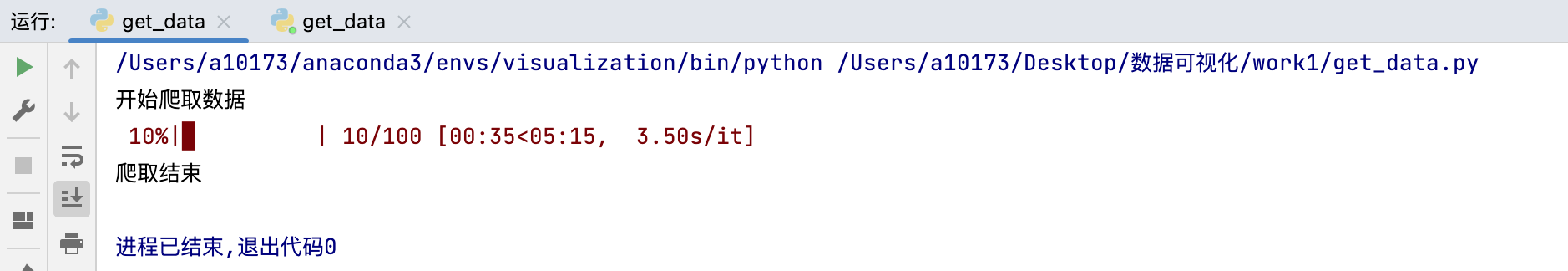
**（2）主题为“研究生”时：**

****

爬取资料部分截图如下：（此为2023年5月25日爬取数据）：



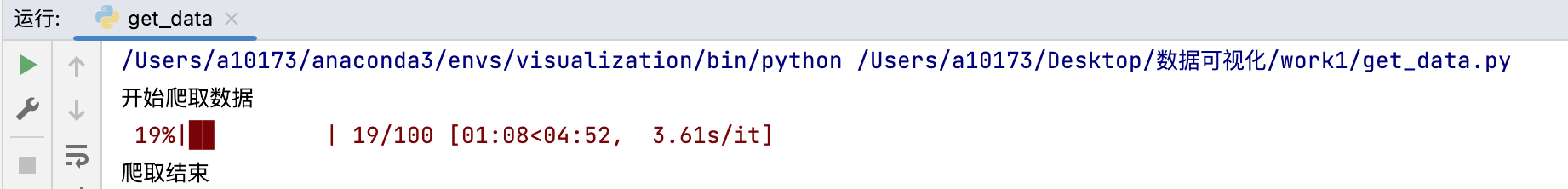
**（3）主题为“杭州”时：**

****

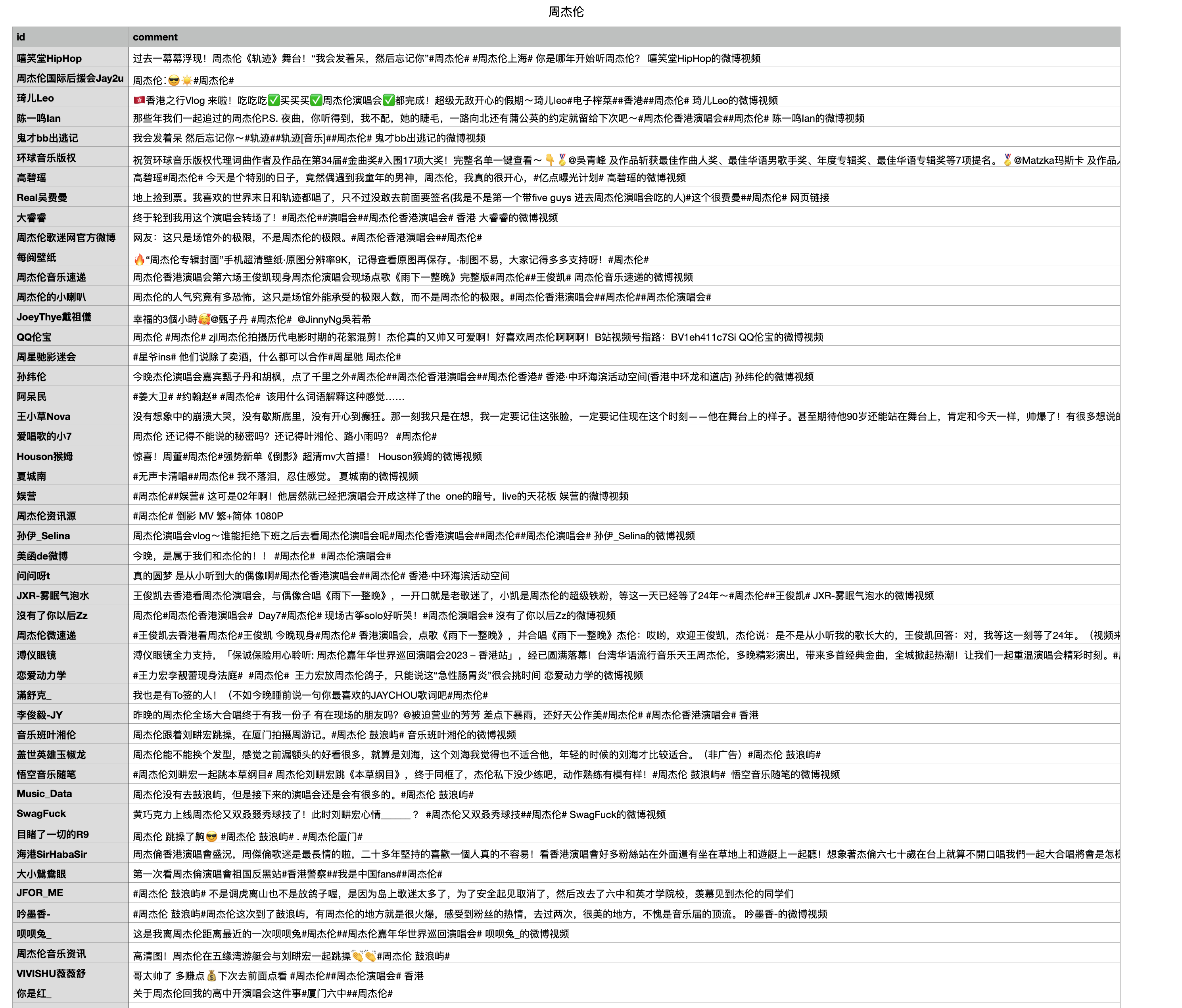
爬取资料部分截图如下：（此为2023年5月25日爬取数据）：



**（4）主题为“周杰伦”时：**

****

爬取资料部分截图如下：（此为2023年5月25日爬取数据）：



4.代码简介

代码目标是爬取微博上的帖子并将结果存储在CSV文件中。主要用到了requests、pandas、re和tqdm库。

首先，定义了一个要搜索的话题和爬取的轮次数。

然后，创建一个CSV文件用来存储结果，文件名根据指定的话题决定。CSV文件的标题行为"id"和"comment"。设置HTTP请求的header和cookies，这些信息将在后面的requests请求中使用。使用trange函数（tqdm库的一个函数，用于在Python长循环中添加一个进度提示器）来遍历指定的轮次数。对于每一轮，构造一个URL，该URL将话题和当前的页码插入到预定义的URL模板中。使用requests库的get函数向构造的URL发送HTTP请求，并将返回的JSON响应存储在json变量中。再解析JSON响应，对于响应中的每一条卡片（每条微博帖子），提取用户名和帖子文本，然后将这些数据存储在DataFrame中，并将其追加到之前创建的CSV文件中。这里用到了正则表达式来清除文本中的HTML标签。

最后，在每一轮结束时，休眠3秒，防止由于高频连续爬取而被微博服务器封锁。