**信息与通信工程学院通信工程专业**

**《Python程序设计》课程大作业**

**示例题目：基于Python程序设计的学生课程成绩分析**

学 院（系）：

专 业：

学 生 姓 名：

学 号：

完 成 日 期：

大连民族大学

**1．背景及目的**

1.1 疫情爬取及数据可视化

2020年的疫情对于全世界来说都是一个痛苦的回忆，3月份开始首先在我国爆发，后期在国外更是呈现指数式增长，截至今日，全球的疫情还未根除！为了有效的观察世界疫情局势的变化，我们通过Python爬虫和简单的数据可视化将疫情情况通过图片的形式展示出来从而更方便的帮助大家了解局势，以及做好应对措施！

1.2数据资源的获取与分析

本次采用python的requests模块从腾讯疫情网站抓取世界每日疫情变化数据，抓取的数据包括新增死亡、新增治愈、新增确诊以及对应的日期，另外我们也关注世界累计人数，这对于宏观调控也有很大的帮助，抓取的数据有累计死亡、累计治愈、累计确诊以及对应的日期信息！

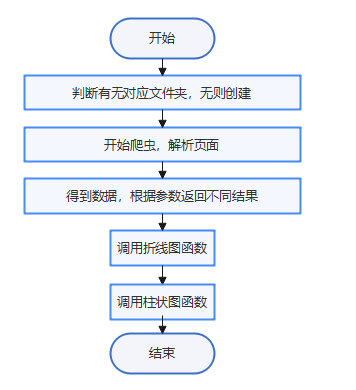
**2．Python程序方案设计**

包括以下内容：

* Python程序设计总体目标

首先会抓取页面的相关数据信息，将所有信息抓取到后，通过调用对应的函数画出对应的折线图和柱状图，然后存放到新建的文件夹（疫情图片）中！

* Python程序设计总体方案(附总体方案流程图)



* 总体方案中各组成部分的功能说明

**3．Python程序编程实现**

3.1 Python编程环境说明 (Python程序开发平台及开发工具)

3.2 Python程序设计功能实现

3.2.1 数据导入

通过requests爬取腾讯疫情页面以及通过response.json()按照字典规则进行提取对应属性的数据，通过append()方法把所有同属性的数据存放到一个相同的列表中，方面后面return给别画图函数进行画图！

3.2.2 数据预处理

因为疫情时间太长，如果把每天的数据都保存的话，画图的时候会导致画布整体密集，融合在一起，非常不雅观！于是使用if (I % 30 == 0)语句来每隔30个数据选取一个，这样做既能保证疫情发展总体趋势不变，同时方便画图清晰的展示坐标轴情况！

3.2.3 整体性能测试

因为爬取的数据量不大，加上腾讯疫情的反爬机制不强，所以没有采用多线程以及其他的反爬策略，整个程序运行时间也就20s左右！

**4. Python程序设计总结**

4.1 系统运行环境配置

计算机配置：Windows10/Pycharm

Pyhton3.6：Python 是用来编写应用程序的高级编程语言。目前，Python 有 两个版本，一个是 2.x 版本，一个是 3.x 版本，这两个版本是不兼容的。现在 3.x 版本越来越普及，所以我们使用的是 python3.6 版本。Python 为了提供非常完善 的基础代码库，覆盖了网络，文件，GUI，数据库等大量内容。许多功能不必从 0 开始写，直接使用现成的即可。例如，此次设计要用到的 matplotlib 模块，numpy 模块等都可以直接使用。且 Python 有相对较少的关键字，结构简单，和一个明 确定义的语法，代码定义的更清晰，学习起来更加简单。Python 的最大的优势之 一是丰富的库，跨平台的，在 UNIX，Windows 和 Macintosh 兼容很好。Python支持互动模式，可以从终端输入执行代码并获得结果的语言，互动的测试和调试代码片断。

4.2 系统实现方式和结果

4.2.1 爬取相应数据及返回处理后数据

（1）程序代码及详细注释

def data\_spyder(number):

# 定义新变量，接收传进的参数

num = number

# 经过分析网页获取到的数据接口

url = 'https://api.inews.qq.com/newsqa/v1/query/inner/publish/modules/list?modules=chinaDayList,chinaDayAddList,cityStatis,nowConfirmStatis,provinceCompare'

response = requests.get(url=url)

# 万能解码，使中文不会以乱码的形式展现

response.encoding = response.apparent\_encoding

# 接口的数据是json格式的

html = response.json()

"""下面的代码是获取每天的新增的数目"""

day\_add\_datas = html['data']['chinaDayAddList']

# 使用列表的append方法将对应种类的新增数目添加到各自的列表中，以便于后面的可视化操作

day\_deads = [] # 新增死亡

day\_heals = [] # 新增治愈

day\_dates = [] # 日期

day\_confirms = [] # 新增确诊

# 如果数据全部存在列表中，最后绘图时密密麻麻，于是隔30天写入一次

for i in range(len(day\_add\_datas)):

if (i % 30 == 0):

day\_deads.append(day\_add\_datas[i]['dead'])

day\_heals.append(day\_add\_datas[i]['heal'])

day\_confirms.append(day\_add\_datas[i]['confirm'])

day\_dates.append(day\_add\_datas[i]['date'])

"""下面的代码是获取当天的各种累计数目"""

day\_datas = html['data']['chinaDayList']

# 同样的操作，将各种栏目的数据添加到对应的列表中

deads = []

heals = []

dates = []

confirms = []

for i in range(len(day\_datas)):

if(i % 30 == 0):

deads.append(day\_datas[i]['dead'])

heals.append(day\_datas[i]['heal'])

dates.append(day\_datas[i]['date'])

confirms.append(day\_datas[i]['confirm'])

# 现在是将上面的代码整体封装成一个函数，通过调用不同的参数获取不同的return返回对象

if num == 1:

# 如果num是1的话返回的是累计的数目

return(dates,deads,heals,confirms)

elif(num == 2):

# 如果num是2的话返回的是每天的新增数目

return(day\_dates,day\_deads,day\_heals,day\_confirms)

else:

print('传进的参数有问题，调用失败！')

（2）运行结果及分析

成功爬取相应数据，通过给定number不同的值以获取不同的返回数据进行可视化展示

4.2.2 画折线图

（1）流程及操作步骤说明：

程序运行后可画出对应的新增/累计三项数据折线图

（2）程序代码及详细注释

day\_dates, day\_deads, day\_heals, day\_confirms = data\_spyder(2)

titles = ['日期','新增死亡数','新增治愈数','新增确诊数']

datas = [day\_dates,day\_deads,day\_heals,day\_confirms]

for i in range(1,len(titles)):

plt.plot(day\_dates,datas[i],label="每日{}".format(titles[i]),linewidth=3,color='r',marker='o',markerfacecolor='blue',markersize=12)

plt.xlabel('日期')

plt.ylabel(titles[i])

plt.title('每日{}'.format(titles[i]))

# 使中文可以正常显示出来

plt.rcParams['font.sans-serif'] = ['SimHei']

plt.rcParams['axes.unicode\_minus'] = False

plt.legend()

plt.savefig(path + '/' + titles[i] + '\_折线图.jpg')

（3）运行结果及分析

运行该程序后会在【疫情图集】文件夹下面生成折线图

4.2.3 画柱状图

（1）流程及操作步骤说明

程序运行后可画出对应的新增/累计三项数据柱状图

（2）程序代码及详细注释

day\_dates, day\_deads, day\_heals, day\_confirms = data\_spyder(2)

titles = ['日期','新增死亡数','新增治愈数','新增确诊数']

datas = [day\_dates,day\_deads,day\_heals,day\_confirms]

for i in range(1,len(titles)):

# 下面时绘制柱状图

plt.bar(day\_dates,datas[i])

plt.xlabel('日期')

plt.ylabel(titles[i])

plt.title('每日{}'.format(titles[i]))

# 使中文可以正常显示出来

plt.rcParams['font.sans-serif'] = ['SimHei']

plt.rcParams['axes.unicode\_minus'] = False

plt.savefig(path + '/' + titles[i] + '\_柱状图.jpg')

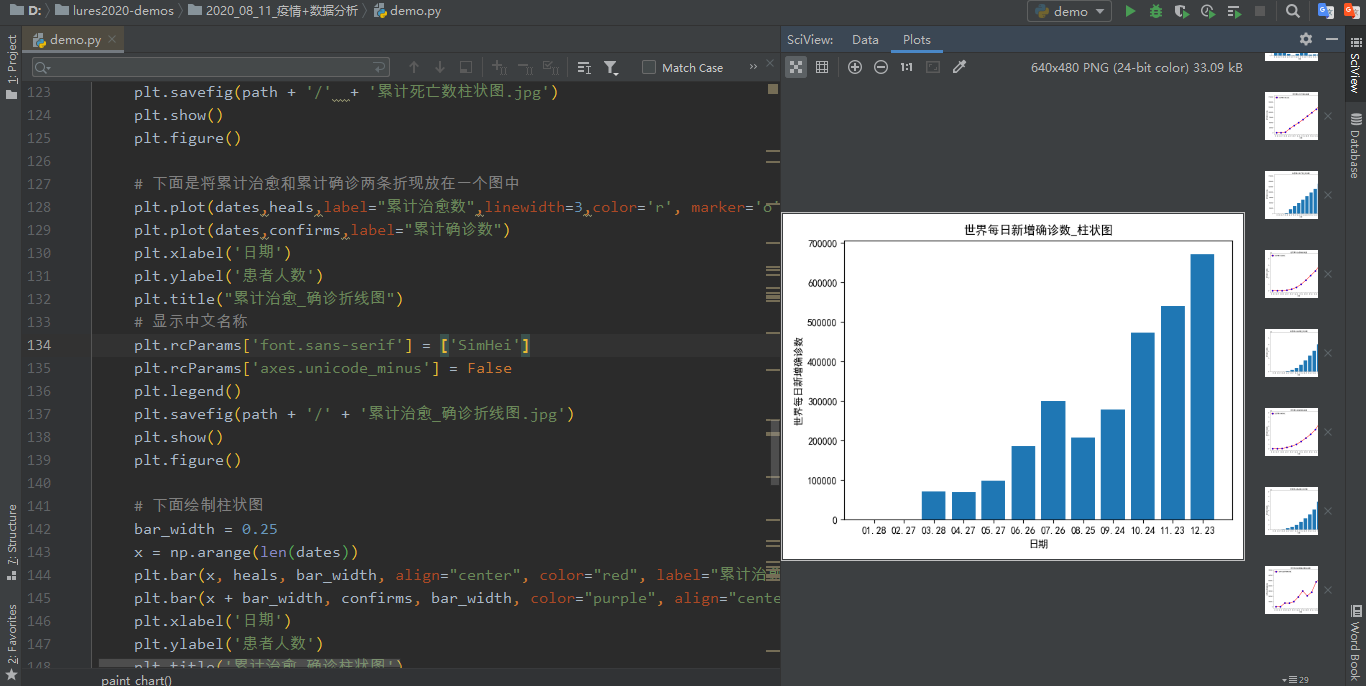
plt.show()

（3）运行结果及分析

运行该程序后会在【疫情图集】文件夹下面生成柱状图

**5. 完整结果展现**

**（1）Pycharm终端截图：**

****

（2）效果截图：

