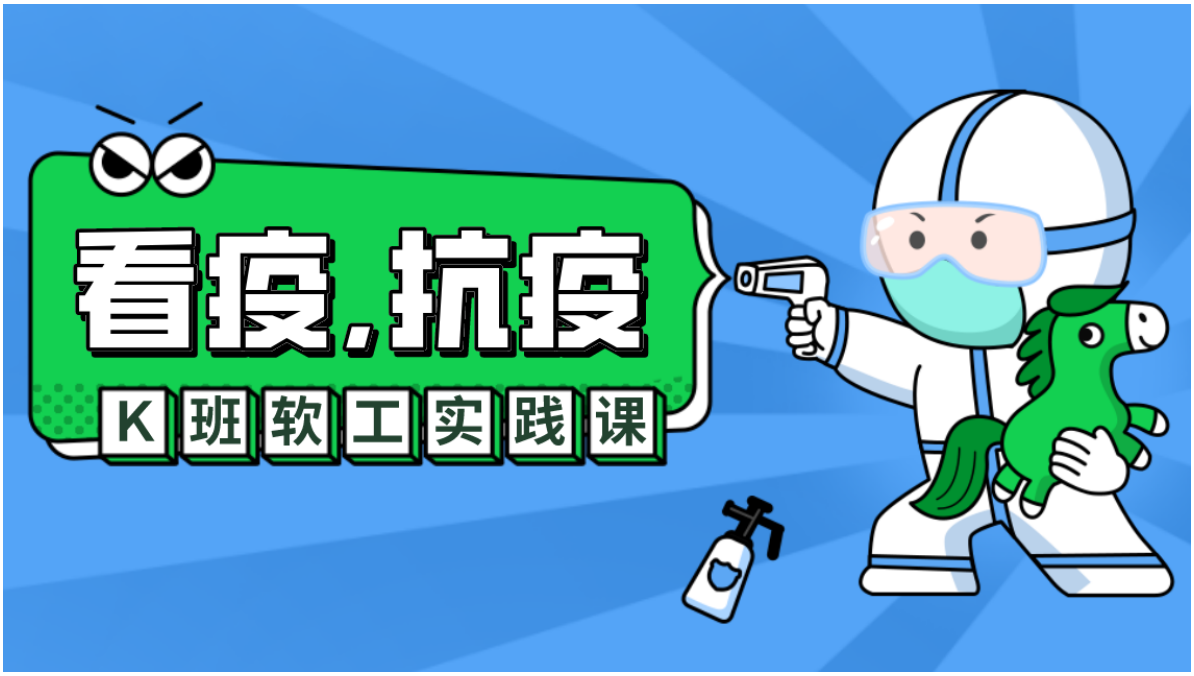

【看疫，抗疫】2022·K班个人编程任务

本次个人编程任务为：**新冠疫情本土病例分析挖掘**
作业提交截止时间为：**2022.09.18 23:59**

Part·1 任务背景

新冠肺炎疫情是百年来全球发生的最严重的传染病大流行，是新中国成立以来我国遭遇的传播速度最快、感染范围最广、防控难度最大的重大突发公共卫生事件。国家卫健委官网疫情通报板块记录着疫情发生以来所有省份的每日疫情情况，但通报里成篇的省份名和数据让人看得实在眼花缭乱，如何快速获取重点信息，并实现可视化，成为了热点问题。

本次作业要求实现中国本土新冠疫情统计，使用爬虫在卫健委官网爬取每日疫情数据，其中需要完成对中国大陆以及所有省份每日新增确诊、每日新增无症状感染病例数的统计，并生成Excel表格，同时实现可视化，实现每日热点功能。

Part·2 任务要求

2.1 数据获取

1. 利用爬虫在卫健委官网爬取每日所需疫情数据，数据范围从疫情开始至今为宜。

2.2 数据统计

1. 统计**中国大陆**每日本土**新增确诊**人数及**新增无症状感染**人数，境外输入类型和疑似病例等无需统计。
2. 统计**所有省份**包括港澳台每日本土**新增确诊**人数及**新增无症状感染**人数，境外输入类型和疑似病例等无需统计。
3. 将统计的数据利用编程工具或开发包自动写入Excel表中。

2.3 每日热点

1. 利用现有数据加上统计与机器学习算法等分析挖掘出任何一日值得关注的事件（如某地**突发**疫情、疫情的重要**转折与变化**等）。以能反映疫情每日态势为宜。

2.4 数据可视化

1. 对采集的数据集进行可视化表示，可使用曲线图或者柱状图等来实现，需要有**时间**、**省份**维度等，具体实现可自由发挥。
2. 附加题：使用**数据可视化大屏**来实现统计数据的可视化，动态数据大屏更佳。

Part·3 编码要求

1. 在Github仓库中新建一个**学号为名**的文件夹。
2. 在开始实现程序之前，在PSP表格[附录1]记录下你**估计**在程序开发各个步骤上耗费的时间，在你实现程序之后，在PSP表格记录下你在程序的各个模块上**实际**花费的时间。
3. 使用C++ 、Java或者python3实现，提交python代码时请附带上requirements.txt。C++请使用Visual Studio Community 2019进行开发，运行环境为64-bit Windows 10。对于C++/Java，还需将编译好的程序发布到Github仓库中的releases中。
4. 提交的代码尽可能经过Code Quality Analysis工具的分析并消除所有的警告。
5. 完成项目的首个版本之后，请使用性能分析工具（例如Studio Profiling Tools）来找出代码中的性能瓶颈并进行改进。
6. 使用Github[附录2]来管理源代码和测试用例，**代码有进展即签入Github**。签入记录不合理的项目可能会被询问项目细节。
7. 使用单元测试[附录3]对项目进行测试，并使用插件查看测试分支覆盖率等指标。

Part·4 评分细则

4.1 得分表

评分点	描述	得分
1.1	在Github仓库中新建一个学号为名的文件夹，同时在**博客正文首行**给出作业Github链接，**并登录软工在线平台完善信息。**	2
2.1	在开始实现程序之前，用附录提供的**PSP表格**记录下在各个模块上开发的**预估耗时**	3
2.2	在完成程序之后，用附录提供的**PSP表格**记录下在各个模块上开发的**实际耗时**	3
3.1	**项目设计与技术栈**	5
3.2	**爬虫与数据处理**	20
3.3	**数据统计接口部分的性能改进**	6
3.4	**每日热点的实现思路**	6
3.5	**数据可视化界面的展示**	15
4.1	完成作业过程中的**心得体会。**	10
5.1	项目结构的完整性（单元测试、异常处理、模块设计是否满足低耦合的要求）	10
5.2	代码的**可读性**（注释等）	10
5.3	变量、函数、类命名的**规范化**	10
6.1	附加题：使用**数据可视化大屏**来实现统计数据的可视化，动态数据大屏更佳。	+10
6.2	附加题：除了任务要求四点外的自行实现其他与题目相关的创新功能。在博客中给出：功能的实现思路（简单描述）、功能的创新点以及对于该功能能过解决的问题	+10

4.1.1 博客评分规则（70%）

1.(1.1)在Github仓库中新建一个学号为名的文件夹，同时在**博客正文首行**给出作业Github链接，**并登录软工在线平台完善信息。（2'）**

请用以下列一级标题分割你的博客（冒号后的文字设置为一级标题）

——博客评分为半自动，如果没有按要求分割博客，造成评分出现问题，将不予处理

2.标题一：**一、PSP表格**

> (2.1)在开始实现程序之前，在附录提供的**PSP表格**记录下你估计将在程序的各个模块的开发上耗费的时间。 **（3'）**
>
> (2.2)在你实现完程序之后，在附录提供的**PSP表格**记录下你在程序的各个模块上实际花费的时间。 **（3'）**

3.标题二：**二、任务要求的实现**

> (3.1)**项目设计与技术栈。**从阅读完题目到完成作业，这一次的任务被你拆分成了几个环节？你分别通过什么渠道、使用什么方式方法完成了各个环节？列出你完成本次任务所使用的技术栈。 **（5'）**
>
> (3.2)**爬虫与数据处理。**说明业务逻辑，简述代码的设计过程（例如可介绍有几个类，几个函数，他们之间的关系），并对关键的函数或算法进行说明。 **（20'）**

> (3.3)**数据统计接口部分的性能改进。**记录在数据统计接口的性能上所花费的时间，描述你改进的思路，并展示一张性能分析图（例如可通过VS 2019/JProfiler的性能分析工具自动生成），并展示你程序中消耗最大的函数。 **(6')**
>
> (3.4)**每日热点的实现思路。**简要介绍实现该功能的算法原理，可给出必要的步骤流程图、数学公式推导和核心代码实现，并简要谈谈所采用算法的优缺点与可能的改进方案。 **(6')**
>
> (3.5)**数据可视化界面的展示。**在博客中介绍数据可视化界面的组件和设计的思路。 **(15')**

4.标题三：**三、心得体会**

> (4.1)在这儿写下你完成本次作业的心得体会，当然，如果你还有想表达的东西但在上面两个板块没有体现，也可以写在这儿~**(10')**

4.1.2 代码评分规则（30%）

总分**30**分，程序评分是根据代码质量综合考量给出的评分，主要考察如下方面：

1. 项目结构的完整性（单元测试、异常处理、模块设计是否满足低耦合的要求） **(10')**
2. 代码的可读性（注释等） **(10')**
3. 变量、函数、类命名的规范化**(10')**

4.2 扣分规则

1. **作业截止后发布博文**：作业截止24小时内补交在原分数上扣20分，24-48小时内扣40分，48-72小时内扣60分，以此类推扣到0分为止；
2. **发布博文但未提交任务**：24小时内补提交扣10分，24-48小时内扣20分，48-72小时内扣30分，以此类推扣到0分为止（请注意：本次作业开始，不会再在QQ课程群中提醒需要提交任务）；
3. **缺交**：以上两项扣分规则仅适用于在**作业截止之后、评测开始之前**补提交的情况！作业截止后任意时间开始进行评测，评测开始的具体时间不会提前告知，评测开始时仍未发布博客或未提交任务均视为缺交，评为0分；
4. **抄袭**：后台检测到的不诚信行为评为0分。

Part·5 注意事项

1. 撰写博文时，请在编辑区右上角选择**【切换为MarkDown】**，使用MarkDown语言撰写今后的博客文档；
2. 将作业标题**严格命名**为**2022软工K班个人编程任务**，不要修改。
3. 不同日期的卫健委疫情日报数据格式可能不同，需要程序自动鉴别与自适应。
4. 从**卫健委官网**爬取数据为**爬虫与数据处理**模块的分数上限，若从其他网站爬取文本信息进行数据分析，满分为该模块分数上限的70%，若直接爬取已经处理好的统计数据，满分为该模块分数上限的40%。
5. 台湾是中国神圣的领土。

Part·6 附录

6.1 PSP表格

PSP是卡耐基梅隆大学（CMU）的专家们针对软件工程师所提出的一套模型：Personal Software Process (PSP， 个人开发流程，或称个体软件过程)。

PSP2.1	Personal Software Process Stages	预估耗时（分钟）	实际耗时（分钟）
Planning	计划		
· Estimate	· 估计这个任务需要多少时间		
Development	开发		
· Analysis	· 需求分析 (包括学习新技术)		
· Design Spec	· 生成设计文档		
· Design Review	· 设计复审		
· Coding Standard	· 代码规范 (为目前的开发制定合适的规范)		
· Design	· 具体设计		
· Coding	· 具体编码		
· Code Review	· 代码复审		
· Test	· 测试（自我测试，修改代码，提交修改）		
Reporting	报告		
· Test Repor	· 测试报告		
· Size Measurement	· 计算工作量		
· Postmortem & Process Improvement Plan	· 事后总结, 并提出过程改进计划		
	· 合计		

一个功能完备的程序不是一蹴而就的。可将一个大任务划分为可操作的小任务，同时最好按照任务难度或紧急程度指定各个任务的完成次序。因此，在动手开发之前，要先估计将在程序各模块开发所需耗费的时间，以及完成整个项目所需的时间，将这个[估计值]记录下来，写成PSP 的形式。
PSP的目的是：记录工程师如何实现需求的效率，和我们使用项目管理工具（例如微软的Project Professional，或者禅道等）进行项目进度规划类似。
有关PSP的更多内容，请自行阅读邹欣老师的博客：现代软件工程讲义 2 工程师的能力评估和发展

6.2 Github

请阅读邹欣老师的博客：源代码管理，了解源代码管理的10个实践问题。
本次作业要求使用Github进行源代码管理，**代码有进展即签入Github**。签入记录不合理的项目会被助教抽查询问项目细节。
对代码签入的具体要求如下：根据需求划分功能后，每做完一个功能，编译成功后，应至少commit一次。本例中，至少应区分基本功能和扩展功能，即分别针对基本功能、扩展功能，编译成功后，总共至少应commit两次。具体的功能划分，请自行定义，并在撰写博客时体现出来，遵循自己对需求的功能划分来提交代码即可。
对Commit不是很熟悉的话，请阅读阮一峰的博客：Commit message 和 Change log 编写指南，了解更多细节。

6.3 单元测试

请根据自己以往积累的测试经验，在编码完成之后，提交产品之前，设计测试用例，并编写单元测试，对自己的项目进行测试。
首先，至少应采用白盒测试用例设计方法来设计测试用例，其他测试方法不限。其次，要设计至少10个测试用例，确保你的程序能够正确处理各种情况。最后，结合测试评估的要求，对自己的测试设计进行评价，这些测试用例能满足该程序测试的要求吗？
另一个重要的措施是要把单元测试自动化，这样每个人都能很容易地运行它，并且可以使单元测试每天都运行。每个人都可以随时在自己的机器上运行。团队一般是在每日构建中运行单元测试的，这样每个单元测试的错误就能及时被发现并得到修改。
推荐阅读邹欣老师的博客：现代软件工程讲义 2 开发技术 - 单元测试 & 回归测试