# 功能需求与总体方案设计

## 2**018211309 班 学号：2018211362**

本软件设计时完全参照评分标准与实际需求，并加入了我自创的一些特色功能。此文档我会结合评分标准分析所需的功能，并阐述我的设计思想与方法。而对于各种设计是如何实现的我在《数据结构说明与算法分析》文档中有详细说明。

## 目录

1. **总体设计方案说明**- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - P3

1.1软件开发环境 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -- P3

1.2总体结构和模块划分- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -- P3

1. **界面需求分析与设计思想**- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -- P5

2.1编程语言与编程环境的选取- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - P5

2.2图形界面的需求与其实现的设计方法- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -- P5

**三、旅游线路规划（核心）需求分析**- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - P8

3.1 内核的功能需求- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - P8

3.2 内核设计思想：- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - P9

3.2.1 对需求1的设计- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - P9

3.2.2 对需求2，3，5的设计- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - P11

3.2.3 对需求4的设计- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -- P12

3.2.4对需求6的设计- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - P12

3.2.5对需求7的设计- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - P12

**四、其他需求与对应设计出的特色功能**- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - P13

4.1 交互友好与保护程序安全- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - P13

4.1.1交互友好与程序安全的需求- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -- P13

4.1.2 针对交互友好与程序安全的设计- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - P13

4.2 时间与空间开销需求- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -- P17

4.2.1 程序对时间空间开销的需求- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -- P17

4.2.2 针对时空开销需求的设计- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -- P17

4.3 在评分标准与PPT中未提到的设计- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - P18

## 一、总体设计方案说明

##### 1.1软件开发环境

软件环境：Windows 10

硬件环境：Intel Core i7-8750H 8G内存

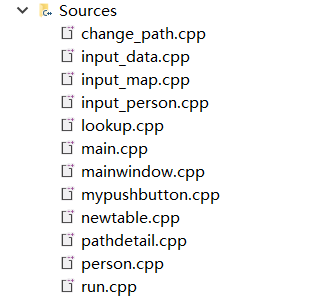
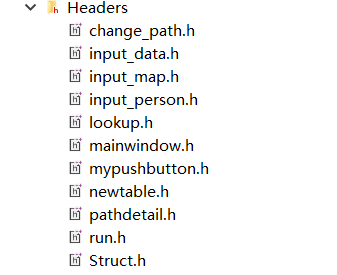
开发工具：Visual Studio 2019 + QT 5.12.8

##### 1.2总体结构和模块划分

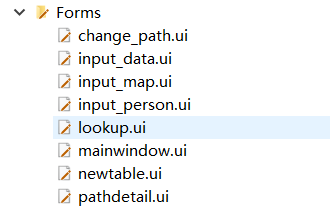
本软件实现由12个C++文件模块组成，其中主体逻辑实现由以下9个模块组成：main.cpp，mainwindow.cpp，input\_data.cpp，input\_map.cpp，input\_person.cpp，lookup.cpp，change\_path.cpp，newtable.cpp，run.cpp

在另外3个模块中，person.cpp实现了旅客的创建，规划路径。pathdetail.cpp为图形界面提供了详细的旅客旅行计划的输出，mypushbutton.cpp重写了QT的按钮类，使得本软件的主要功能按钮拥有点击特效。

本软件还有8个QT的ui文件，11个头文件，ui文件实现了大部分的窗口与对话框界面，头文件对上述C++文件中使用的类提供了定义，值得说明的是，地图信息与旅客类由Struct.h定义实现。



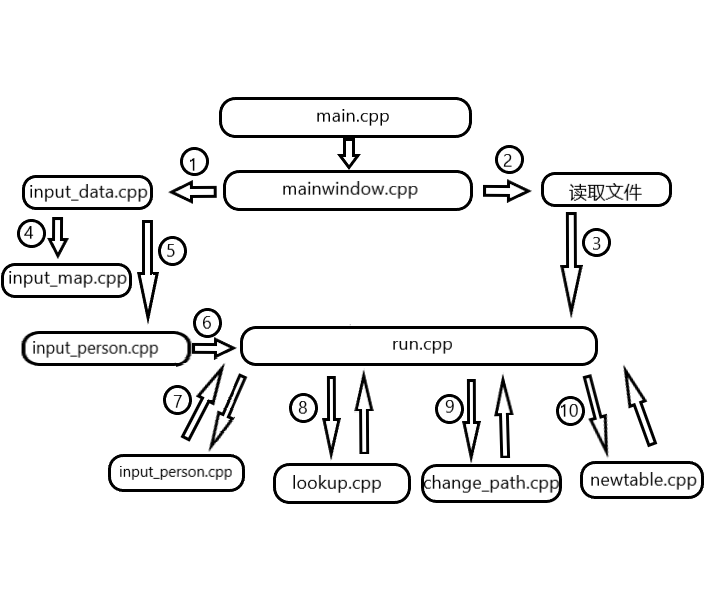
头文件 C++文件



UI文件

下图是本软件的主要逻辑流程图，我将对应着流程图简述软件的结构与各模块的功能。

1. 程序由main.cpp中的主函数进入，主函数调用mainwindow.cpp来显示主窗口。
2. 进入主窗口后，会有两种输入数据的方式，手动输入和直接读取，手动输入会调用input\_data.cpp的输入窗口进入输入模式（图中①），而读取数据文件的函数（图中②）在mainwindow.cpp内，不需要调用其他文件。
3. 我将地图信息的输入与旅客信息的输入划分为了两个模块input\_map.cpp，input\_person.cpp，由input\_data.cpp顺序调用，先输入地图信息（图中④），再输入旅客信息（图中⑤）。
4. 输入/读取完信息后会进入（图中③，⑥）到运行界面，由run.cpp实现。
5. 运行界面中，由主体界面与五个功能组成，界面显示地图信息与旅客行程状态，五个功能分别是：添加旅客（图中⑦），详细查询（图中⑧），更改旅客计划（图中⑨），添加时刻表（图中⑩），添加城市（在run.cpp中实现，所以图中未标出）。当这五个功能调用结束后，都会返回到run.cpp中，继续维持运行界面。



## 二、界面需求分析与设计思想

##### 2.1编程语言与编程环境的选取

一个软件与用户直接交互的地方就是界面了，所以界面对用户体验来说至关重要。在选做部分（b）条：“使用地图反映出旅行过程”提出了此需求，所以我在设计时，此软件需采用图形界面，最后我决定使用QT来完成UI界面的编写。因为要使用QT，所以决定了本软件使用的语言：C++。

##### 2.2图形界面的需求与其实现的设计方法

在图形界面上，用户的输入会更方便；为了使信息更直观，我还设计用不同的颜色来表示城市的风险等级：红色，0.9；黄色：0.5；绿色：0.2；在运行时，地图上的城市会显示出不同的颜色；此外，我还将两点之间的线路用虚线表示出来，蓝色代表航班，绿色代表火车路线，黄色代表汽车路线，两个城市之间有无通路一目了然；最后，得益于图形界面，我使用以下三种图片分别来代表汽车，火车和飞机，在线路上移动。

汽 车 火 车 飞 机

运行界面如图（1）所示，可以看到，图上有各色的线连接起了城市，当前时间为第一天的4点，图中有一列火车正在哈尔滨到长春之间，另一辆火车在重庆滞留，图上各色的点也反应出了各城市的风险值。即使不进行详细查询，我认为此界面也能最大限度地呈现出重要信息。在图上，利用图形界面我还实现了以下需求：（参加评分标准功能验收部分（2）中的四条，与加分项中的4）

（a）能够实现系统时钟的推进；

（b）系统时间至少精确到小时；

（c）日志文件记录旅客状态变化；

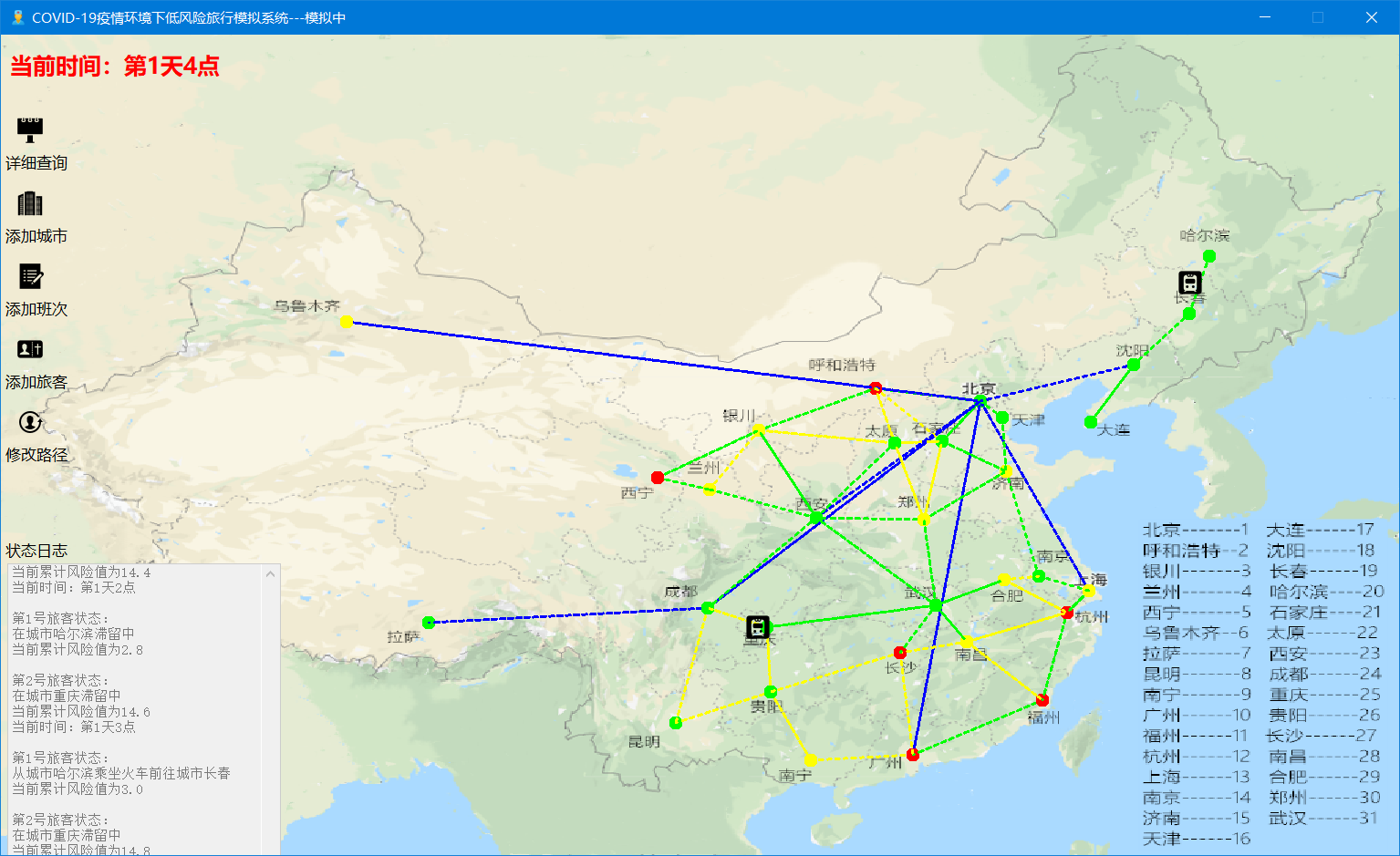
（d）能够查询不同旅客的当前位置信息；

（e）能够采用独立时钟，进行并发旅客推进和查询；

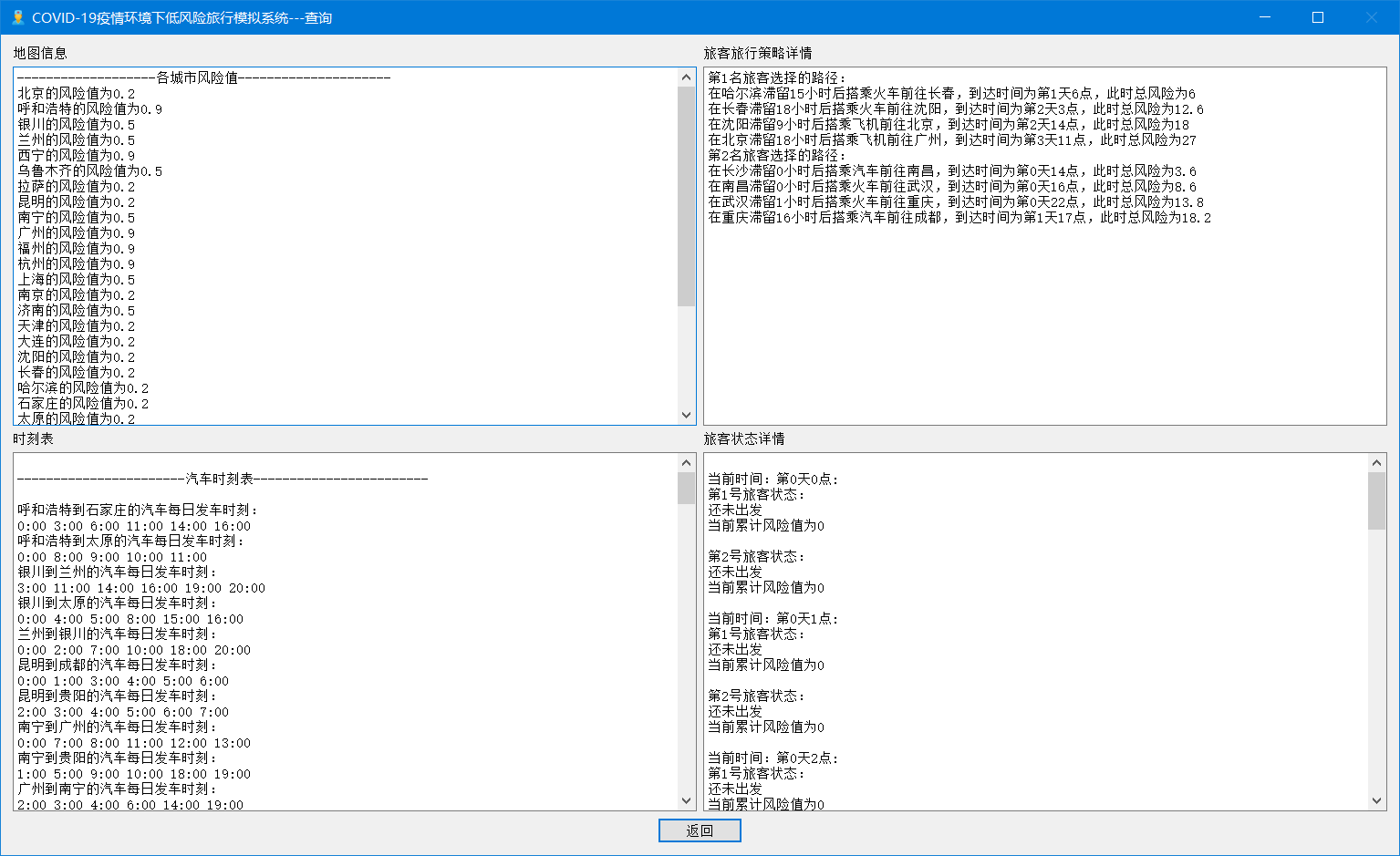
对于（a）需求和（b）需求，我在图片上左上角设置了红色粗体醒目的时钟，实时呈现当前模拟系统的时间，并以小时为单位。此设计也满足了加分项（e）的前半部分。

对于（c）需求，可以看到我的界面左下角有实时的日志显示，不仅能保存到文件，还能实时地把当前事件呈现给用户，用户可以通过此窗口看到：“添加了旅客”“查询了信息”“修改了路线”“添加了城市”“当前时间”“各旅客信息”等等事件摘要。

对于（d）需求，通过界面上的图标、状态日志窗口知晓不同旅客的位置信息，是远远不够的。为此，我添加了详细查询按键（左侧第一个按钮），查询界面如图（2）所示。此设计满足了加分项（e）的后半部分。



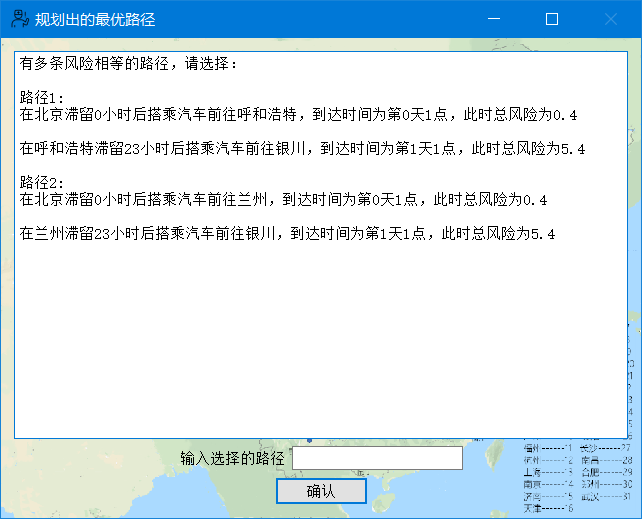
图（1）



图（2）

在图二中，时刻表，当前时刻之前各时刻旅客的状态，各城市风险值，旅客的选路策略都可以供用户查看。

此外，与界面相关的需求还有：“规划好的路线需要输出给用户，包括起点出发的时间，中转城市到达时间，再次出发时间，和终点城市的到达时间等信息；如果有多条线路符合要求，需要都输出给用户。”（功能验收（1）（d））。在设计时，我采用了对话框提示的方式，示意图如下：（因为用真实时刻表的话几乎不会有两点之间多条线路风险值相等的情况，所以此处演示我没有读取真实时刻表，而是手动输入了一些简单的数据便于说明功能。）



可见，北京到兰州有两条风险值完全相同的路径，对于每一条路径，旅客在哪一个城市停留多久，何时出发，何时到达，各时刻风险值是多少…都被输出给了用户，并将由用户输入进行选择。

此外，由于我的软件设计时为了保证灵活性，设计了两种数据输入方式，除了上面介绍的文件读入方式，还有手动输入数据方式，为了输入方便，我将城市用数字编号，以数字代替城市，使输入简便。我将城市与序号的对应关系贴在了界面上，见图（1）右下角。

## 三、旅游线路规划（核心）需求分析

##### 3.1 内核的功能需求

这一部分是本软件的核心。根据评分标准中的需求，此软件需要支持以下功能：

1. 可以支撑运行很多城市和列车时刻表（功能验收1a，加分项1，2）
2. 能够按照起点、终点、出发时间范围规划路线；（功能验收1b）
3. 能够按照限时和最小风险策略规划旅行路线（功能验收1c）
4. 为不同交通工具设置不同单位时间风险值，交通工具单位时间风险值分别为：汽车=2；火车=5；飞机=9。（PPT选做2）
5. 能够按照起点、终点、出发时间范围、途经某些城市和任一旅行策略规划旅行路线。（选做（d））
6. 2种要求的策略规划旅行路线，能够提出性能更好地策略，或者和常规不一样的策略（加分项3）
7. 在旅客进入旅行路线后，可以更改路线；（选做（a））

##### 3.2 内核设计思想：

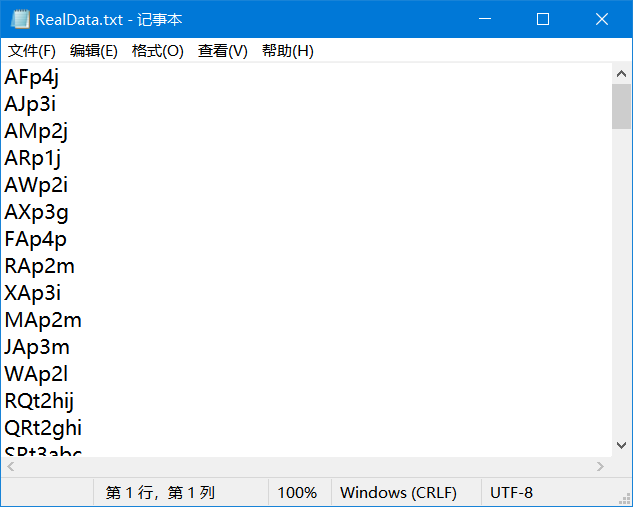
3.2.1 对需求1的设计

针对需求1，我使用python爬取了我国各省会城市+北京的所有航班，但是考虑到PPT中的一条要求：“整个系统中航班数不得超过10个，火车不得超过30列次；汽车班次无限制；”与“系统以小时为单位推进”，我做了如下设计：

（1）只选取直达车次，航班。这么选取数据的设计原因如下：因为所有需要换乘的航班或者汽车，火车，一定可以被拆分成许多段的直达线路，那么只要有了各直达线路的数据就足够表示所有可能的车次了。

（2）经过1之后，光是航班就留下来了六百条，远远超过十条的标准。所以，与实际结合，我选出了五个交通枢纽城市：沈阳，西安，成都，广州，上海。我只保留了这五个城市与北京之间的往返航班。此外，由于乌鲁木齐，拉萨过于偏远，我将这两个城市与最近的枢纽城市西安，成都的往返航班保留。

（3）本软件的输入我设计了两种方式：手动输入和数据读入。在这里我主要结合时刻表阐述数据读入方式的文本格式设计：在文件目录下有一个txt文本文件叫做“RealData.txt”，其中数据如下：



为了方便读取，我没有使用文字，因为读取文件的过程是用户看不到的，此文件不需要对用户阅读友好，使用了ASCII码表示数据，这样还可以使读入数据的代码简洁，提高效率。在此软件中，31个我国主要城市在此文件中用ASCII码编码从“A”（北京）到“\_”（武汉）。交通工具用“b”（bus）“t”（train）“p”（plane）表示。路程所需时间用阿拉伯数字表示，而每班车（飞机）每日的发车时刻用a ~ x（0 ~ 23）表示。数据格式为：

“起始城市ASCII码 终点城市ASCII码 交通工具 所需时间 首发时间”。

城市ASCII编码对照表如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 北京 | A | 杭州 | L | 西安 | W |
| 呼和浩特 | B | 上海 | M | 成都 | X |
| 银川 | C | 南京 | N | 重庆 | Y |
| 兰州 | D | 济南 | O | 贵阳 | Z |
| 西宁 | E | 天津 | P | 长沙 | [ |
| 乌鲁木齐 | F | 大连 | Q | 南昌 | \ |
| 拉萨 | G | 沈阳 | R | 合肥 | ] |
| 昆明 | H | 长春 | S | 郑州 | ^ |
| 南宁 | I | 哈尔滨 | T | 武汉 | \_ |
| 广州 | J | 石家庄 | U |  |  |
| 福州 | K | 太原 | V |  |  |

现在让我以第一条数据“AFp4j”为例解读一下：从北京到乌鲁木齐的飞机，所需四小时，起飞时间为9点。对应的真实航班如下图（为了以小时为单位，对时刻进行了舍入）：



3.2.2 对需求2，3，5的设计

针对功能二，分析可知，旅客乘坐的交通工具与城市风险值不是旅客本身的属性，应该划分给地图模块去实现，所以在读取了数据构造了地图之后，表示一个旅客的行程需要一个五元组：出发城市，终点城市，出发时间，限时，必经城市。在此分析的基础上，我有如下设计：

（1）旅客数据与读入的文件分离，旅客的数据一定要灵活多变，所以无论是手动输入地图数据还是直接读取数据文件，我都要让用户手动输入旅客的数据。用户只需进入运行界面后，点击图（1）中的“添加旅客”案件就可进入输入界面。输入界面如下：

针对需求3，经过分析，我选择将限制时间作为可选项，做了下面的设计：

（1）以上图界面为例，当不勾选限制时间选项时，默认进行最小风险策略.

（2）当勾选上之后执行限时最小风险策略。

针对需求5，在以上两个需求的基础上，只需进行如下设计：

（1）将必经城市也作为可选项，当不勾选时，

（2）必经城市被勾选时，路径必须经过所规定的必经城市，若无这样的路径，会弹出消息提示“不存在这样的路径，添加失败”。

3.2.3 对需求4的设计

由于需求4与算法连系很紧密，所以这一需求的具体实现设计我选择在《数据结构说明与算法分析》文件中进行详细说明，在此不再赘述。

3.2.4对需求6的设计

通过以上的若干设计，此软件水到渠成地拥有了四种选路策略：

（1）风险最小

（2）限时风险最小

（3）风险最小且有必经城市

（4）限时风险最小且有必经城市

3.2.5对需求7的设计

针对需求7进行分析，我发现修改旅客的计划不再需要一个五元组，只需一个四元组就足矣。这是因为五元组中的起始地点已经默认了，就是发出修改计划命令时刻旅客所处的位置。实现此需求的问题是，发出修改命令可以在任何时刻发生，所以我根据旅客不同的状态进行了如下的设计：

1. 旅客已经结束了旅行，不可以修改。
2. 旅客在交通工具上，不可以修改。
3. 旅客刚到达非终点城市，或者在非终点城市滞留，可以修改。

在可以修改旅客的计划时，会出现如下界面，提示出当前所在城市，并接受一个新的四元组。



## 四、其他需求与对应设计出的特色功能

##### 4.1 交互友好与保护程序安全

4.1.1交互友好与程序安全的需求

正如我在说明图形界面的需求的部分所提到的，我希望我的设计能够达到交互友好，我给予了使用者尽可能的方便。我对各种界面的设计，就是遵循着对使用者友好这一原则，我会直观地呈现尽可能多的信息给用户。此外，我不希望用户被过多的操作搞得手忙脚乱，我希望用户第一次使用这个软件就可以灵活自如地操作。

上面的原因催生了我的软件对用户交互友好的需求，然而，如果给予了用户太多的权力，不按照规范操作，会带来一系列不可预测的问题，所以在防止用户搞破坏上也需要进行设计，使程序有一定的鲁棒性。这些原因使得我对程序的操作安全提出了需求。

在下文中，我叙述了实现两个需求所进行的设计。

4.1.2 针对交互友好与程序安全的设计

（1）按钮

下图为主界面按钮图，各按钮功能清晰直观。按钮对称布局，采用了简洁风的图标，力图简洁明了美观。

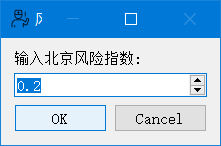


下面五个图标是运行界面的按钮图，同样地，每个按钮的图片与功能对应，下方用文字介绍了功能。

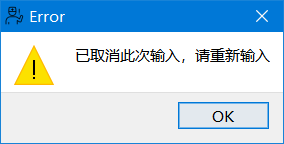
    

（2）操作规范

为防止用户不安规范操作，我将可能会造成数据错乱的操作界面都设置了保护，以下举例说明：

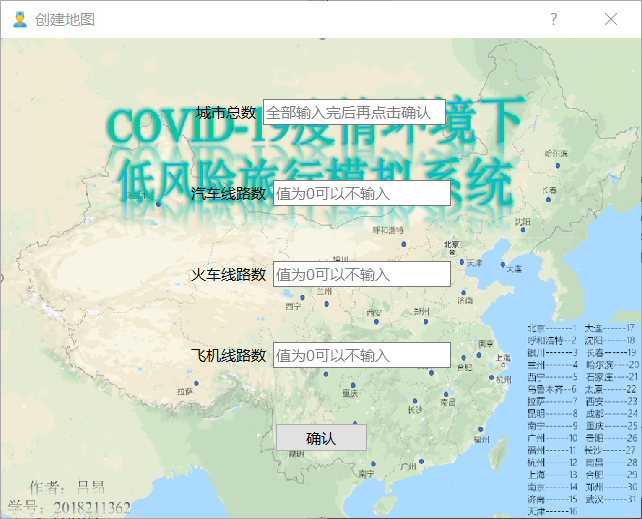


此图为输入城市风险的界面，为了防止用户调出此节目由放弃输入，直接进入运行导致数据不全而程序不能正常运行，我将右上角关闭窗口按钮禁用，如果用户输入错误的话可以点击cancel来重新输入，点击cancel后，遵循交互友好原则，出现下面的提示：



（3）操作提示

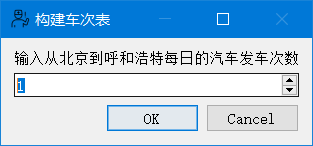
如下图所示，在手动输入模式下，我在输入框内设置了提示语句，为用户提供了指示。并且请注意窗口名字：“创建地图”。在此软件中，每一个界面都有自己功能的窗口名，以提示用户该进行的操作。

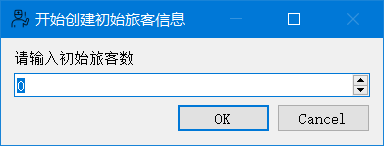
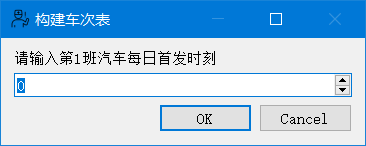


我再就此举两个例子，下面两幅图是运行界面的窗口名。

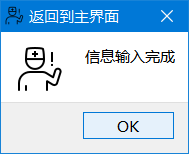
 

在下面三幅图中，与（2）一样，我都禁用了关闭按钮，提供了取消按钮，以限制用户操作规范以保护程序。与（3）一样，这三个窗口都有详细的指示说明，力求交互友好。

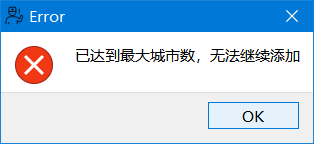




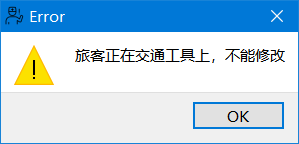
此外，在一些操作完成时，或者操作错误时，会有确认窗口弹出，如下图是数据输入完毕后弹出的提示框：



下图是当城市达到我预设的阈值时，再试图添加城市所弹出的警告窗口。



下图是当试图修改正在交通工具上的旅客的计划时弹出的警告。



类似的设计还有很多，我在此不再赘述。我会在《用户使用说明》文档中详细介绍。

##### 4.2 时间与空间开销需求

4.2.1 程序对时间空间开销的需求

如果降低软件的时间开销，那么会提高用户与程序的交互速度，在软件处理大量数据时，用户等待的时间会短，但是势必要牺牲空间作为代价。如果软件的空间开销很低，那么软件有更强大的能力处理更多的数据，我在设计的时候就可以增加更多的时刻表，但是势必要牺牲时间效率。程序的时间开销与空间开销不可兼得，我在平衡二者的思考中选择了偏向优化时间开销。理由如下：

4.2.2 针对时空开销需求的设计

首先对于存储地图数据用邻接矩阵还是邻接表的问题，我认为，如果想支持真实时刻标，最终我们会得到一个有项的稠密图，对于稠密的图，使用邻接表是不划算的。但是我的软件又支持手动输入，也就是说，在我的这个程序中，城市数不是固定的，可以是从2~31的任意数量，那么城市数很少的时候，使用邻接矩阵就会有很大的浪费。那么，是不是可以在城市数量少的时候使用邻接表，多的时候采用邻接矩阵呢？我认为没有必要，原因是我给了用户增加城市、增加路径、增加时刻表的权力，即使当前城市数很少，我不能保证用户此后不添加数据。如果用户把一个稀疏的图硬生生添加成一个非常稠密的图，并且时刻表非常非常多的时候，如果采用这种方法，在转换的时候就会有时间，空间上的开销。此外，我认为数据被我使用一些特殊设计的数据结构被压缩的很好，无论城市数目多少，直接使用邻接矩阵在空间上的开销对于程序，对于计算机都是微不足道的。

基于以上思想，我将地图与时刻表都存在了邻接矩阵中。（至于我上述提到的使用的数据结构我在《数据结构说明与算法分析》会详细说明）

虽然选择了牺牲空间换时间的这一总体策略，但是在能节约空间的地方还是要节约，不能太草率。比如一个城市到另一个城市的时刻表这种不确定长度的数据，还是做成链表存储为好，一方面节约空间，一方面方便扩展与新增的数据插入。其他的涉及到算法的设计我在此也不叙述了，还是把它们放到了《数据结构说明与算法分析》中。

##### 4.3 在评分标准与PPT中未提到的设计

评分标准中的选做部分（f）项提出了“额外的功能”的需求，我认为我的软件比评分标准与ppt等文档中多了以下功能：

（1）日志实时显示。

（2）增加城市功能。

（3）增加旅客功能。

（4）增加/修改城市之间的通路功能。

（5）增加/修改时刻表功能。

（6）旅客随时可出发，可以在运行前输入旅客信息，也可以在运行中输入。

（7）手动输入和文件读入两种数据输入方式。数据文本文件可扩展。

（8）上述所有针对交互友好和程序安全的设计。

这些功能与前文提到的所有特色设计功能，我都会在《用户使用说明中》，按照操作顺序进行完整细致的讲解。