**数据结构说明和数据字典**

**1 数据结构说明**

本部分将列举低风险旅行模拟系统系统中使用到的所有基本数据结构与示例。

**1.1 用户旅行需求**

·数据结构：String数组；

·存储内容：出发城市、目的城市。

【示例】

String[] listsource = **new** String[]{"北京","上海","广州","哈尔滨","长春","天津","西宁","成都","贵阳","武汉","福州","乌鲁木齐"};

**1.2 班次信息**

·数据结构：Travel\_ticket；

·存储内容：所存储的为某一班次的出发城市、目的城市、交通工具、出发时间、抵达时间，风险城市类型、班次号。

时刻表列表由上述类构成，其长度取决于服务器发送的班次数量。

【示例】

**public** **class** Travel\_ticket {

String[] listsource = **new** String[]{"北京","上海","广州","哈尔滨","长春","天津","西宁","成都","贵阳","武汉","福州","乌鲁木齐"};

**int** source;

**int** dest;

**int** start\_time;

**int** end\_time;

**double** media\_type;

**int** city\_type;

String num;

}

**1.3 计算结果**

·数据结构：path类；

·存储内容：本结构中存放两种策略在计算完成后的返回结果。分别用于存放有序的旅行安排路径、抵达时间以及总开销，按序存放旅行安排的班次，每个班次均有所存储的为某一班次的出发城市、目的城市、交通工具、出发时间、抵达时间，风险城市类型、班次号。

【示例】

**public** **class** path {

**int** total=0;

**int** timetotal=0;

Travel\_ticket[] paths=**new** Travel\_ticket[30];

}

**1.3 时刻表**

·数据结构：Scedule类；

·存储内容：本结构存储所有城市的的班次，每个城市的班次均有完整的所存储的某班次的出发城市、目的城市、交通工具、出发时间、抵达时间，风险城市类型、班次号。

【示例】

**public** **class** Scedule {

path[] citys=**new** path[12];

**public** Scedule()

{

**int** totalcity=0;

**int** totalNum=0;

}

**2 数据字典**

此部分主要对本系统中所使用到的关键变量、函数的名称、作用以及数据结构等进行介绍。

**2.1 关键变量**

|  |  |  |
| --- | --- | --- |
| 变量名 | 类型 | 作用 |
| timer | Timer | 定时器，用于推进系统时间 |
| Timetabletest | Schedule类 | 存储所有的班次表 |
| pathtest | path类 | 格式化存储路径信息 |
| date | LocalDateTime | 用于实时获取当前时间 |
| pane | JLayeredPane | 分层窗口网络，用于将窗口分层 |
| listsource | 字符串数组 | 存储出发城市 |
| Listdest | 字符串数组 | 存储目的城市 |
| bestpathtest | best\_path类 | 类中有获取最短路径方法 |
| Bestpathtest.START | 整型 | 存储出发时间 |
| Bestpathtest.timetable | Schedule类 | 存储系统发送的时刻表 |
| Bestpathtest.existNum | 整型数组 | 记录已经经过的城市 |
| Bestpathtest.path\_num | 整型数组 | 记录路径 |
| Bestpathtest.path\_weight | 整型数组 | 记录源点到各个城市之间的权值 |
| Path.total | 整型 | 路径中班次个数 |
| Travel\_ticket.num | 字符串 | 班次序列号 |

**2.2 关键函数**

|  |  |  |
| --- | --- | --- |
| 函数名 | 返回值类型 | 作用 |
| Travel\_ticket.toString() | 字符串 | 将班次信息格式化为字符串并返回 |
| path.gethours | int | 获取从出发到到达目的城市所需要的小时数 |
| path.getindexs | int | 在查询状态下，返回当前所处的状态 |
| Schedule类构造函数 | 无 | 从Time.txt文件中读取所有的班次信息 |
| Best\_path.getweight | double | 获取两个城市之间班次的最小风险值 |
| Best\_path.getTimeweight | double | 获取两个城市之间班次的限时最小风险值 |
| user\_socket\_threading | 无返回值 | 与服务端进行信息交互 |
| best\_path. find\_least\_weight | Int类型 | 最小风险策略算法 |
| Time\_limited\_minimum\_risk\_strategy\_Path | Int 类型 | 限时最小风险策略 |
| Userlog.appendMethod | 无返回值 | 用户日志的记录与保存 |