数据结构说明和数据字典

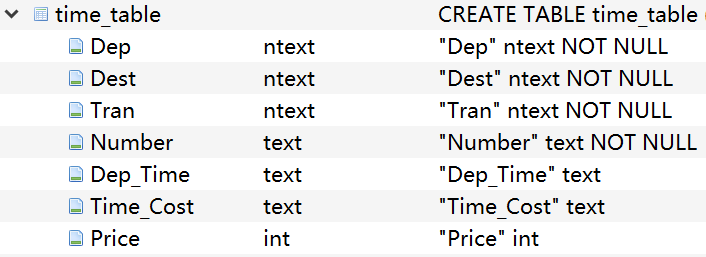
1. 数据
2. 城市、交通工具危险值

|  |  |
| --- | --- |
| 城市 | 风险值 |
| 交通工具 | 风险值 |

1. city\_dangers = {'南京': 0.5, '北京': 0.9, '成都': 0.5, '杭州': 0.2, '广州': 0.5, '武汉': 0.9, '上海': 0.9, '重庆': 0.2, '青岛': 0.9,'深圳': 0.2, '郑州': 0.5, '西安': 0.2}    # 在城市停留的风险值
2. trans\_dangers = {"汽车": 2, "火车": 5, "飞机": 9}     # 交通工具的风险值
3. 所有班次时间表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 起点 | 终点 | 交通工具 | 航班号 | 起始时间 | 路程时间 | 价格 |

数据库中time\_table的信息：

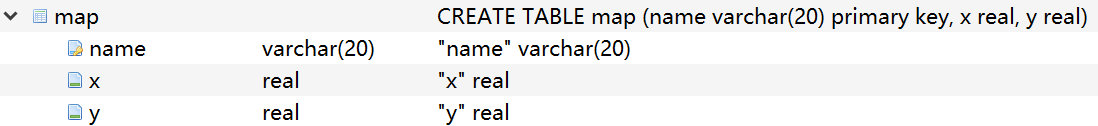


如：

1. time\_table\_values = [
2. ('成都', '北京', '飞机', '3U8883', '9:30', '2:45', 1710),
3. ]
4. 所有城市经纬度信息

|  |  |  |
| --- | --- | --- |
| 城市名 | 经度 | 维度 |

数据库中map的信息：



如：

map\_geo = {'成都': [104.06, 30.67],

'武汉': [114.31, 30.52],}

1. 类

1.界面类

1. **class** MainWindow(QMainWindow):      # 主图形界面
2. **def** \_\_init\_\_(self, qApp=None):  # 初始化
3. self.log\_file = open("log.txt", "w") # 日志文件
4. self.gui = uic.loadUi(ui\_file, self) # 界面设计
5. self.solution = Strategy.Solution() # 策略
6. self.map\_pic = QPixmap("./map.jpg") # 背景图
7. self.current\_x = 0 # 当前位置（经纬坐标）
8. self.current\_y = 0
9. self.start\_time = "" # 起始时间
10. self.path = [] # 计算出来的路径
11. self.set\_traveller = False # 是否模拟行程
12. self.waiting = False # 旅客是否在等待航班
13. self.current\_time = 0 # 当前时间
14. **def** set\_initial(self):  # 设置默认参数
15. **def** kind\_changed(self):     # 选择不同的策略
16. **def** compute\_clicked(self):  # 计算路径
17. **def** simulate\_clicked(self):     # 开始模拟
18. **def** simulation(self):       # 模拟旅客路径
19. **def** paintEvent(self, event):    # 刷新旅客位置、显示旅客状态
20. **def** drawText(self, qp):  # 画图，每次旅客位置更新，或初始化时重新画图
21. **def** quit(self):     # 关闭窗口并退出

2.策略类

1. **class** Solution:  # 根据不同的参数、策略返回最有路径
3. **def** \_\_init\_\_(self):
4. self.start\_place = "北京" # 默认参数
5. self.end\_place = "广州"
6. self.start\_time = "7:00"
7. self.limit\_time = -1 # 默认最小风险策略
8. **def** shortestPath(self, start\_place="北京", end\_place="广州",
9. start\_time="9:15", limit\_time=-1)  # 计算最优路径