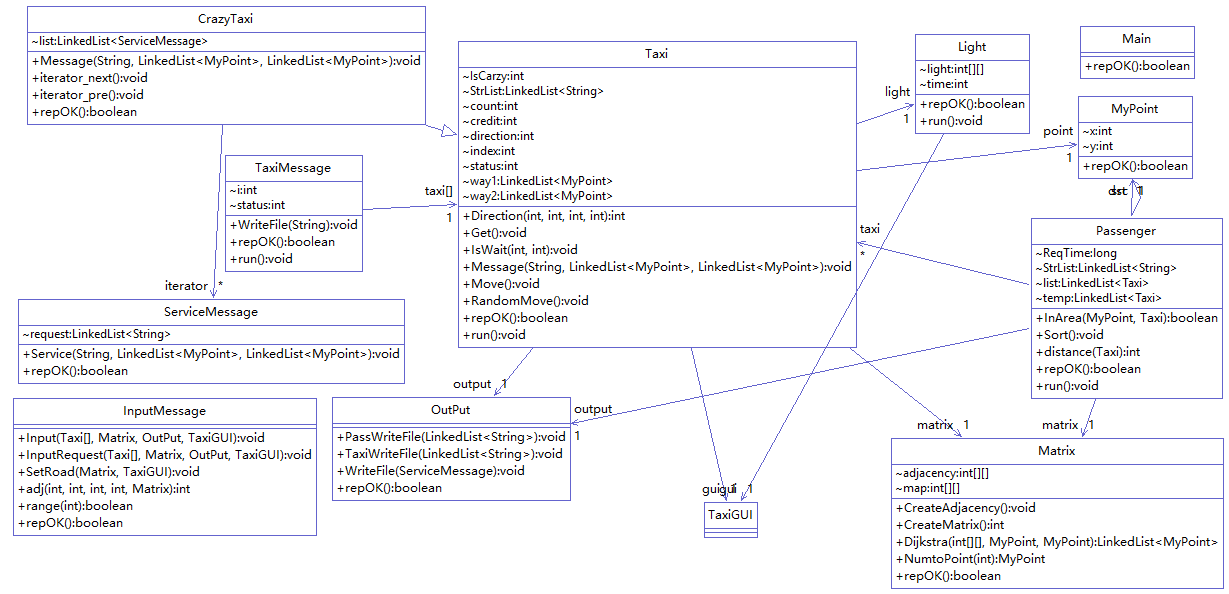
UML类图：



UML功能顺序图

：Main

mian()

:Taxi/CarzyTaxi

:light

:gui

:InputMessage

:Matrix

CreateMatrix()

CreateAdjacency()

LoadMap(matrix1.map, 80)

start()

start()

d

I input(taxi,matrix1,output,gui)

SetRoadStatus()

：Passenger

start()

State=2

线程协同顺序图：

:Main

:Taxi/CrazyTaxi

:Passenger

Pasenger(int src\_x,int src\_y,int dst\_x,int dst\_y,long ReqTime,Taxi [] taxi,Matrix matrix,OutPut output)

Start()

Taxi(Matrix matrix,int index,OutPut output,TaxiGUI gui,Light light)

Start()

State=2

way1=temp.get(i).matrix.Dijkstra(matrix.adjacency,temp.get(i).point,src); way2=temp.get(i).matrix.Dijkstra(matrix.adjacency,src,dst);

行为逻辑状态图：

出租车：

送乘客至目的地并等待1s/State=2

服务状态

等待服务

随机游走200ms[state=3]/ 到达乘客所在地并等待1s/State=1

接单状态

随机游走200ms[state=2]/

随机游走200ms[已经连续保持随机游走20s]/state=0

停止状态

停止1s/

乘客：

向4\*4范围出租车发出请求

进入三秒窗口期等待出租车响应

结束线程

结束线程

经过3s且无车可响应/结束线程

经过3s且有车可响应/选取距离最近、流量最小的车辆服务