**Includes.h:**

包含宏定义(#define),标准库头文件,自制库头文件.

注意宏定义应放于最上方,自制库头文件应放于最下方.

**Class.h**

定义了城市群体类,城市单体类,路径类.

**路径类(包含):**

公有变量:

行进方式 ; 起点 ; 终点 ; 路径长度 ; 路径花费金钱 ; 路径花费时间 ;

函数:

输入路径信息.

**城市单体类:**

公有变量:

城市名称 ; 不同方式对应的路径数组 ; 距离(用于计算最短路) ;

总花费金钱,总花费时间(单源最短路时,到达终点的花费) ;

最短路的来源城市名称(用于打印最短路) ;

Vis记录值(用于最短路计算时的标记) ;

函数:

寻找路径(给定路径信息,寻找是否存在该路径) ;

**城市群体类:**

私有变量:

城市数量 ;

Map映射(从城市名称映射到vector中该城市所在的下标) ;

公有变量:

城市vector(储存多个城市单体) ;

函数:

构造函数(放置哨兵置城市vector,以使其下标从1开始) ;

addCityByNumber(批量添加城市) ;

addCity(添加一个城市) ;

reviseCityName(修改城市名称) ;

eraseCity(擦除城市) ;

添加路径 ; 修改路径 ; 擦除路径 ;

寻找最便宜的道路 ;

寻找最快速的道路 ;

**Class.cpp**

**void Citys::addCity():**

输入城市名称,并将其加入vector末端,下标此时为vector的大小-1,因为下标从1开始,前置有1个哨兵元素.

**void Citys::addCityByNumber():**

输入城市数量,再调用相应次数的addCity()函数.

**int Citys::reviseCityName():**

输入需要修改的城市的名称和要修改为的名称. 将新名称的map映射设置为与原来的一样,再擦除原来的.

**int Citys::eraseCity():**

输入要擦除的城市名称,并直接删除map映射.

**int Citys::addPath():**

依次输入路径的信息,并在起点城市的path数组里从末端添加该路径.

**int Citys::revisePath():**

输入要修改的路径的信息,在对应起点的path数组查找是否有信息一致的一条路径,若存在,则修改该条(一条)路径的信息.

**int Citys::erasePath():**

输入并查找对应路径,并使用迭代器擦除该路径.

**int Citys::findCheapestPath():**

使用最短路算法(堆优化的Dijkstra或者Floyd),寻找花费金钱最少的路径.

每次分开寻找飞机或火车两种行进方式的最短路,并输出相应的路径,花费的金钱和时间.

**int Citys::findFastestPath():**

使用最短路算法(堆优化的Dijkstra或者Floyd),寻找花费时间最少的路径.

每次分开寻找飞机或火车两种行进方式的最短路,并输出相应的路径,花费的金钱和时间.

**int Citys::listCitys():**

列出所有城市.

**int Citys::listPathsOfOneCity():**

列出一个城市的所有能到的路径,即a->b,c->a,若列出a的路径,则只列出a->b.

**TAS.cpp**

**(TrafficAdvisorySystem)**

**(交通咨询系统)**

主程序

打印用户提示信息和执行相应函数.