

文档名称 Document Name		密级 Confidentiality Level
景区信息管理系统		仅供收件方查阅
模板编号 Template Code	版本 Version	文档编号 Document Code

景区信息管理系统

——图的操作和应用

版权所有 不得复制

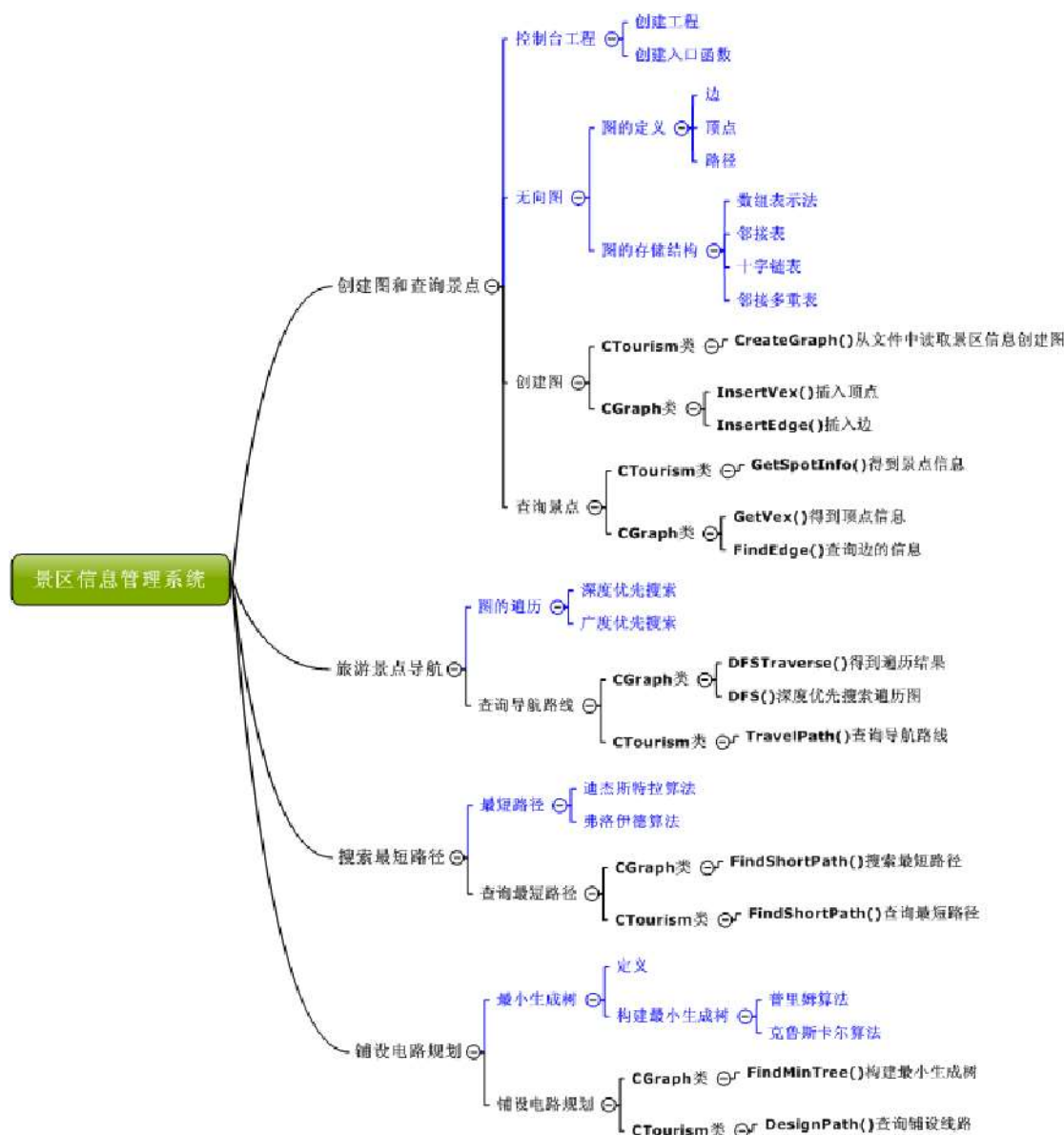
Copyright © Ruankosoft Technologies(Shenzhen), Co., Ltd.

. All Rights Reserved

目录

1 目标	3
2 功能需求	4
2.1 业务背景	4
2.2 景点数据	4
2.3 系统功能	6
3 开发环境	9
4 设计思路	9
4.1 程序设计	9
4.2 界面设计	10
4.3 算法设计	10
4.4 类设计	11
4.5 数据结构设计	12
5 技术分析	13
6 跟我做	14
6.1 迭代开发思路	14
6.2 编程实现	15
7 运行	16
7.1 程序导入	16
7.2 运行结果	16

1 目标



实践目标

- (1) 掌握图的定义和图的存储结构
- (2) 掌握图的创建方法
- (3) 掌握图的两种遍历方法
- (4) 理解迪杰斯特拉 (Dijkstra) 算法
- (5) 理解最小生成树的概念和普里姆 (Prim) 算法
- (6) 掌握文件操作
- (7) 使用 C++ 语言，利用图的数据结构，开发景区信息管理系统。

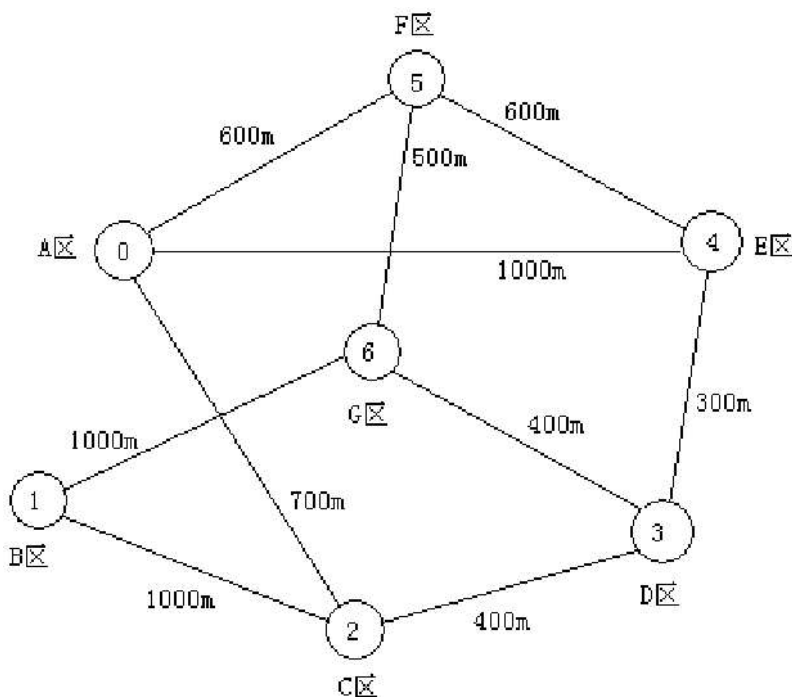
2 功能需求

2.1 业务背景

现有一个景区，景区里面有若干个景点，景点之间满足以下条件：

- (1) 某些景点之间铺设了道路（相邻）
- (2) 这些道路都是可以双向行驶的（无向图）
- (3) 从任意一个景点出发都可以游览整个景区（连通图）

开发景区信息管理系统，对景区的信息进行管理。使用图的数据结构来保存景区景点信息，为用户提供创建图、查询景点信息、旅游景点导航、搜索最短路径、铺设电路规划等功能。



2.2 景点数据

景区的数据包含景点信息和景点之间的道路信息。分别由两个文本文件存储。Vex.txt文件用来存储景点信息； Edge.txt 文件用来存储道路信息。

- (1) 景点信息：景点编号、名字和介绍

编号	名字	介绍
0	A区
1	B

	区	
2	C
	区	
3	D
	区	
4	E
	区	
5	F
	区	
6	G
	区	

景点信息

Vex.txt文件第 1 行，记录景区的景点个数。从第 2 行开始，每 3 行记录一个景点信息。每个景点信息的记录格式为：

- ① 景点编号 (编号从 0 开始，逐个加 1)
- ② 景点名字
- ③ 景点介绍

文件格式与示例如下：

景点个数
景点编号
景点名字
景点介绍
景点编号
景点名字
景点介绍
.....

→

Vex.txt - 记事本

文件(F) 编辑(E) 格式(O)

7

0

A区

风景优美，气候宜人。

1

B区

风景优美，气候宜人。

2

C区

风景优美，气候宜人。

Vex.txt

(2) 道路信息：景点 1，景点 2、两个景点之间的距离。

景点 1	景点 2	距离(m)
A	C	700
A	E	1000
A	F	600
B	C	1000
B	G	1000
C	D	400
D	E	300
D	G	400
E	F	600
F	G	500

道路信息

Edge.txt 文件中，每 1 行记录 1 条道路信息。格式为：“景点 1 的编号 景点 2 的编号 道路的长度”。（每个字段使用空格符分割）

若文件中没有某两个景点的信息，就表示这两个景点之间没有直接的路径。

文件格式及示例如下：

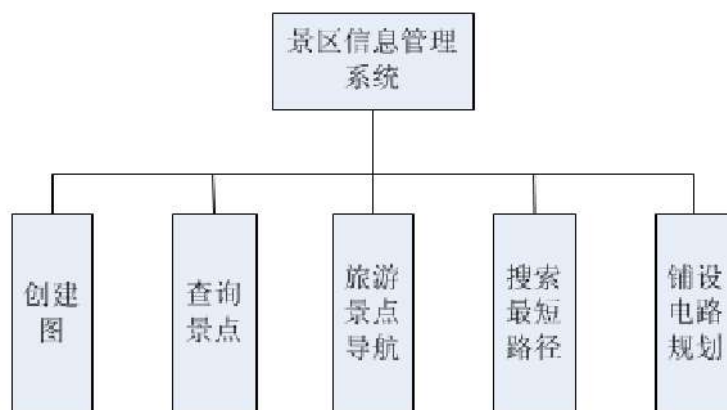
景点1的编号	景点2的编号	道路的长度
景点1的编号	景点2的编号	道路的长度
景点1的编号	景点2的编号	道路的长度
.....		



Edge.txt - 记事本		
文件(F)	编辑(E)	格式(O)
0	2	700
0	4	1000
0	5	600
1	2	1000

Edge.txt

2.3 系统功能



(1) 创建图

输入：从 Vex.txt 文件中读取景点信息，从 Edge.txt 文件中读取道路信息。

处理：根据读取的景区信息创建景区景点图。

输出：

创建成功，依次输出：

- ① 顶点数目
- ② 顶点编号
- ③ 顶点名字
- ④ 边两端的顶点编号
- ⑤ 边的权值

创建失败，输出失败的提示信息。

输出格式如下：

```
C:\WINDOWS\system32\cmd.exe

===== 创建景区景点图 =====
顶点数目: 7
----- 顶点 -----
0-A区
1-B区
2-C区
3-D区
4-E区
5-F区
6-G区
----- 边 -----
(0,0,2) 700
(0,0,4) 1000
(0,0,5) 600
(0,1,2) 1000
(0,1,6) 1000
(0,2,3) 400
(0,3,4) 300
(0,3,6) 400
(0,4,5) 600
(0,5,6) 500
```

(2) 查询景点

输入：想要查询的景点的编号。

处理：根据输入的景点编号，查询该景点及相邻景点的信息。

输出：

- ① 景点名字
- ② 景点介绍
- ③ 相邻景区的名字
- ④ 到达相邻景区的路径长度

输出格式如下：

```
C:\WINDOWS\system32\cmd.exe

===== 查询景点信息 =====
0-A区
1-B区
2-C区
3-D区
4-E区
5-F区
6-G区
请输入想要查询的景点编号: 2
C区
风景优美，气候宜人。门票20元。
----- 周边景区 -----
C区->A区 700m
C区->B区 1000m
C区->D区 400m
```

(3) 旅游景点导航

输入：起始景点的编号。

处理：使用深度优先搜索 (DFS) 算法，查询以该景点为起点，无回路游览整个景区的路线。

输出：所有符合要求的导航路线。

输出格式如下：

```
C:\WINDOWS\system32\cmd.exe
===== 旅游景点导航 =====
0-A区
1-B区
2-C区
3-D区
4-E区
5-F区
6-G区
请输入起始点编号: 2
导游路线为:
路线1: C区 -> A区 -> F区 -> E区 -> D区 -> G区 -> B区
路线2: C区 -> B区 -> G区 -> D区 -> E区 -> A区 -> F区
路线3: C区 -> B区 -> G区 -> D区 -> E区 -> F区 -> A区
路线4: C区 -> B区 -> G区 -> F区 -> A区 -> E区 -> D区
路线5: C区 -> D区 -> E区 -> A区 -> F区 -> G区 -> B区
```

(4) 搜索最短路径

输入：

① 起始景点的编号

② 终点的编号。

处理：使用迪杰斯特拉 (Dijkstra) 算法，求得从起始景点到终点之间的最短路径，计算路径总长度。

输出：

① 最短路线

② 路径总长度

输出格式如下：

```
C:\WINDOWS\system32\cmd.exe
===== 搜索最短路径 =====
0-A区
1-B区
2-C区
3-D区
4-E区
5-F区
6-G区
请输入起点的编号: 1
请输入终点的编号: 4
最短路线为: B区->C区->D区->E区
最短距离为: 1700
```


(5) 铺设电路规划

输入：景区的所有景点信息和道路信息

处理：根据景区景点图使用普里姆 (Prim) 算法构造最小生成树，设计出一套铺设线路最短，但能满足每个景点都能通电的方案。

输出：

- ① 需要铺设电路的道路
- ② 每条道路铺设电路的长度
- ③ 铺设电路的总长度

输出格式如下：

```

C:\WINDOWS\system32\cmd.exe
===== 铺设电路规划 =====
在以下两个景点之间铺设电路：
A区 - F区 600m
F区 - G区 500m
G区 - D区 400m
D区 - E区 300m
D区 - C区 400m
C区 - B区 1000m
铺设电路的总长度为：3200
  
```

3 开发环境

类型	工具与环境
开发工具	Microsoft Visual Studio 2010
技术运用	图的定义、图的存储、图的遍历、图的连通性、最短路径、最小生成树、类和对象、文件操作

4 设计思路

程序输入的数据，可从文件中直接读取，也可以从界面中输入。本程序将直接从 Vex.txt 和 Edge.txt 这两个文件中读取数据，采用图的存储结构，应用深度优先搜索算法 (DFS)、迪杰斯特拉 (Dijkstra) 算法、普里姆 (Prim) 算法等，开发景区信息管理系统。

4.1 程序设计

使用 Microsoft Visual Studio 2010 开发工具，创建一个空的控制台工程 (Win32 Console Application)。利用图的存储结构来保存景区景点图，使用 C++ 语言开发景区

信息管理系统，工程名为 **GraphCPro**。

- (1) 添加 **CGraph** 类，用来定义图的数据结构，实现图的相关操作。
- (2) 添加 **CTourism**类，用来实现景区信息管理系统的相关功能。
- (3) 添加 **Main.cpp** 文件，在文件中创建程序入口函数 **int main(void)**。

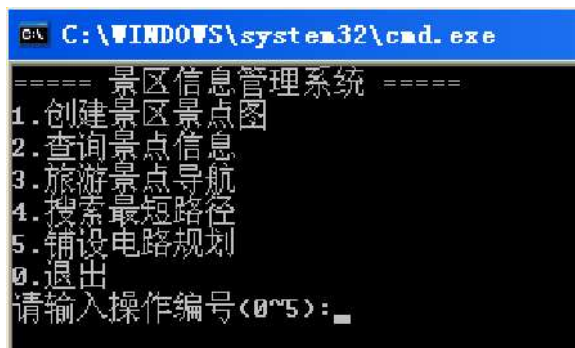


4.2 界面设计

在 **int main(void)**函数中输出菜单，将系统功能列出来，供用户选择。

输入数字	功能
1	创建景区景点图
2	查询景点信息
3	旅游景点导航
4	搜索最短路径
5	铺设电路规划
0	退出

使用 **while** 循环输出主菜单。使用 **cin** 获得用户的输入，使用 **switch-case** 语句判断具体是哪个功能。界面效果图如下：



4.3 算法设计

(1) 旅游景点导航

- ① 使用**深度优先搜索算法 (DFS)** 遍历景区景点图，得到一条导航路线；
- ② 改进深度优先搜索算法，得到多条导航路线。

(2) 搜索最短路径

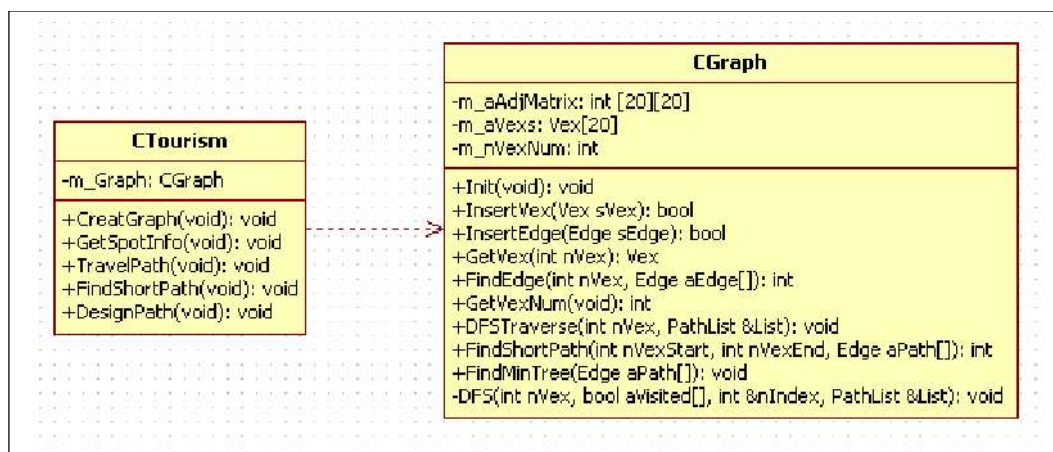
使用**迪杰斯特拉 (Dijkstra) 算法**求得两个顶点间的最短路径。

(3) 铺设电路规划

使用**普里姆 (Prim) 算法**构造最小生成树，得到铺设电路规划图。

4.4 类设计

本程序将使用两个类： CGraph 类与 CTourism类。 CGraph 类用于实现图的数据结构和相关操作。 CTourism类用于实现系统的主要功能。



(1) CGraph类

CGraph 类为程序的数据结构类，用于存储景区景点图，实现图的相关操作。

① 数据成员

private:

`int m_aAdjMatrix[20][20];` —— 邻接矩阵

`Vex m_aVexs[20];` —— 顶点数组

`int m_nVexNum;` —— 顶点个数

② 成员函数

public:

`void Init(void)`，初始化图结构。

`int InsertVex(Vex sVex)`，将顶点添加到顶点数组中。

`int InsertEdge(Edge sEdge)`，将边保存到邻接矩阵中。

`Vex GetVex(int nVex)`，查询指定顶点信息。

`int FindEdge(int nVex, Edge aEdge[])` , 查询与指定顶点相连的边。
`int GetVexnum(void)` , 获取当前顶点数。
`void DFSTraverse(int nVex, PathList &List)` , 获得遍历的结果。
`int FindShortPath(int nVexStart, int nVexEnd, Edge aPath[])` , 搜索两点间的最短路径。

`int FindMinTree(Edge aPath[])` , 构造最小生成树。

private:

`void DFS(int nVex, bool aVisited[], int &nIndex, PathList &List)` , 深度优先搜索遍历图

(2) CTourism 类

CTourism类为景点旅游信息管理功能类,用于实现系统的主要功能。

① 数据成员

private:

`CGraph m_Graph;` —— CGraph 对象,用于存储景区景点图。

② 成员函数

public:

`void CreateGraph(void)` , 读取文件,创建景区景点图。
`void GetSpotInfo(void)` , 查询指定景点信息,显示到相邻景点的距离。
`void TravelPath(void)` , 查询旅游景点导航路线。
`void FindShortPath(void)` , 搜索两个景点间的最短路径。
`void DesignPath(void)` , 查询铺设电路规划图。

4.5 数据结构设计

(1) 图的存储

当保存图结构时,即要保存**顶点信息**,也要保存**边**。图可用数组或链表来存储。

- ① **数组表示**, 常用一维数组来保存顶点的集合,使用二维数组来保存边的集合;
- ② **链表表示**, 常用**邻接表**、**十字链表**等方式存储图的顶点和边的信息。

本程序中使用数组表示法存储图:

定义**一维数组** `Vex m_aVexs[20]` 保存顶点信息,最多允许有 20个顶点。

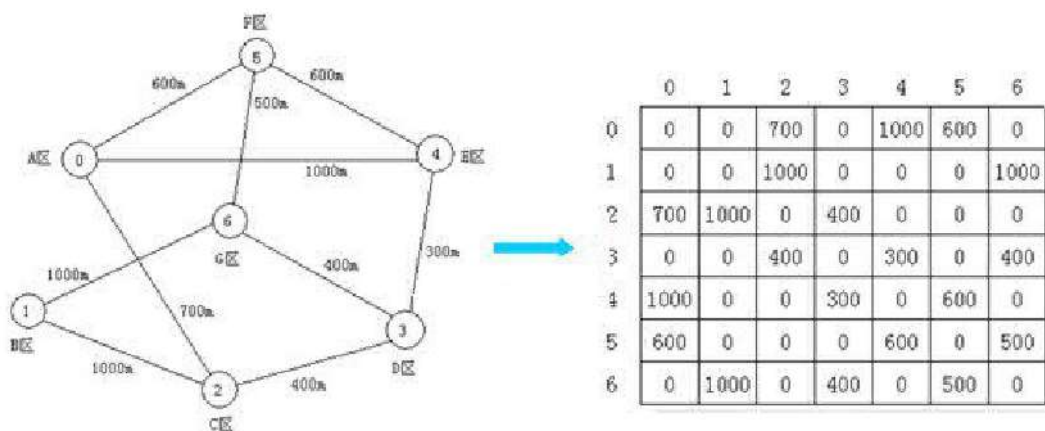
定义**二维数组** (邻接矩阵) `int m_AdjMatrix[20][20]` 保存边的集合,数组中**每个元素的值即为边的权值**。

(2) 景区景点图

景区的地图可以看做是一个**带权无向图**,使用**邻接矩阵**来保存。

- (1) 景区中的**所有景点即为图的顶点**。

- (2) 当两个景点之间铺设的有道路时，表示两个顶点相连，为一条边。
- (3) 两个景点之间的距离，即为边的权值。权值为 0 表示两个顶点不相连。



(3) 顶点和边的信息

定义 Vex 结构体，存储图的顶点信息。

```
struct Vex
{
    int num;           // 景点编号
    char name[20];     // 景点名字
    char desc[1024];   // 景点介绍
};
```

定义 Edge 结构体，存储图的边的信息。

```
struct Edge
{
    int vex1;          // 边的第一个顶点
    int vex2;          // 边的第二个顶点
    int weight;        // 权值
};
```

5 技术分析

1、图

(1) 图的定义

无向图：顶点、边、邻接点、连通性

(2) 图的存储结构

顶点的存储：一维数组

边的存储：关系矩阵

(3) 图的创建 (创建图和景点查询)

(4) 图的遍历 (旅游景点导航)

深度优先搜索遍历

(5) 最短路径 (搜索最短路径)

迪杰斯特拉算法

(6) 最小生成树 (铺设电路规划)

普里姆算法

2、类和对象

(1) 类的定义

(2) 成员变量和成员函数

(3) 访问权限： public 、 private 、 protected

3、文件操作 (创建图)

(1) 打开文件： fopen()

(2) 格式化读文件： fscanf()

(3) 关闭文件： fclose()

6 跟我做

本程序将以迭代的思路进行开发，并分若干专题来讲解每个迭代的内容。迭代如下：

(1) 创建图和查询景点

(2) 旅游景点导航

(3) 搜索最短路径

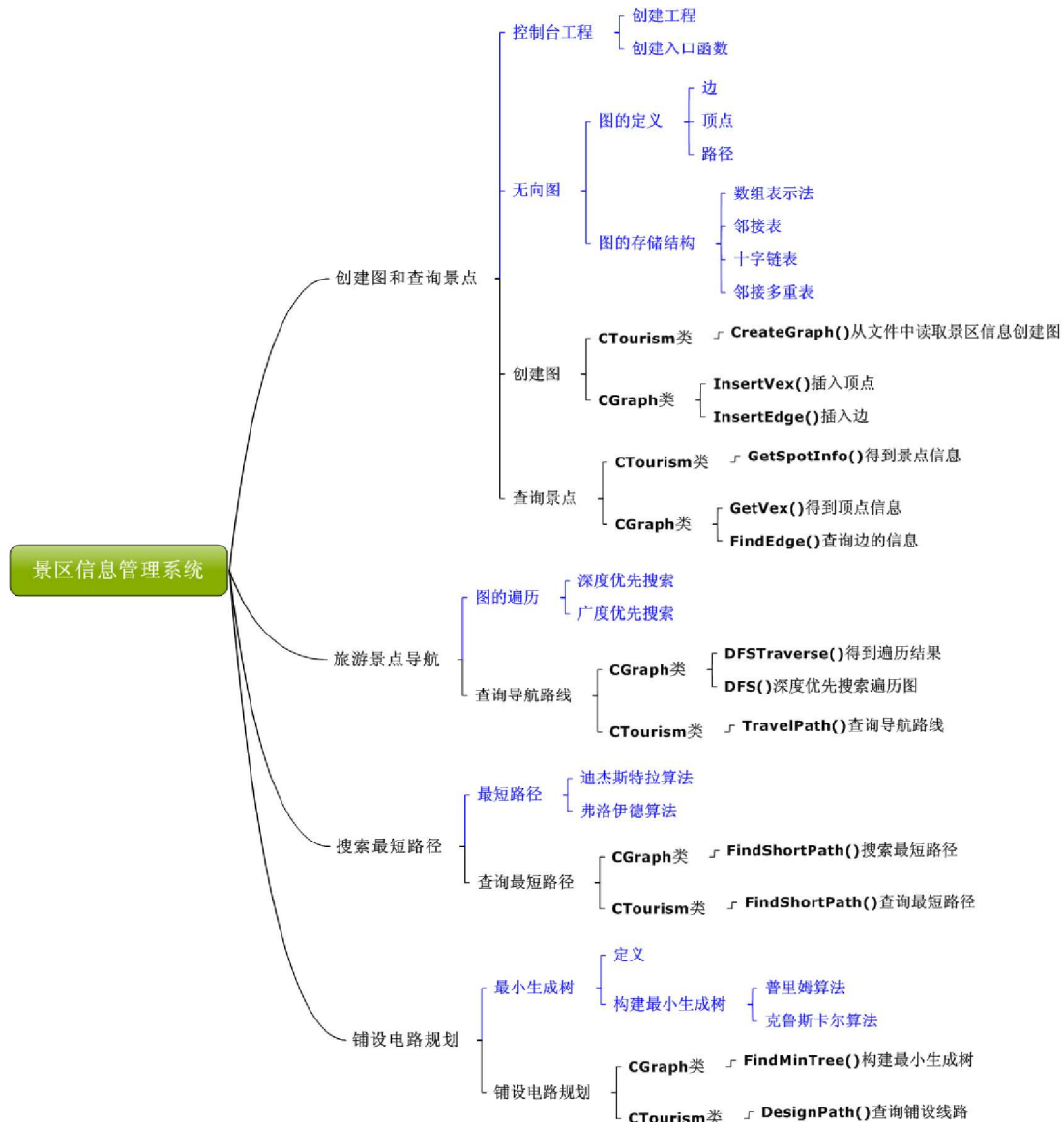
(4) 铺设电路规划

6.1 迭代开发思路

每次只设计和实现系统的一部分，逐步完成的方法叫迭代开发，本程序按功能模块划分为 4 个迭代。

“景区信息管理系统”项目以知识地图为主线，以软件工程的知识作为指导思想，按照项目的功能划分为 4 次迭代，以迭代的方式进行开发。采用迭代开发思想进行开发景区信息管理系统。其中的每个迭代就当作一个完整的项目开发过程，来进行开发。

下图为“景区信息管理系统”项目知识地图。



6.2 编程实现

每次迭代的内容，详见各个专题的参考资料。

1、创建图和查询景点信息

- (1) 创建工程 GraphCPro
- (2) 定义图 CGraph 类
- (3) 创建景区景点图
- (4) 查询景点信息
- (5) 编译和运行

2、旅游景点导航

- (1) 遍历景区景点图 （一条路线）

- (2) 优化遍历算法 (多条路线)
- (3) 编译和运行

3、搜索最短路径

- (1) 搜索最短路径
- (2) 查询最短路径
- (3) 编译和运行

4、铺设电路规划

- (1) 构建最小生成树
- (2) 查询铺设电路规划图
- (3) 编译和运行

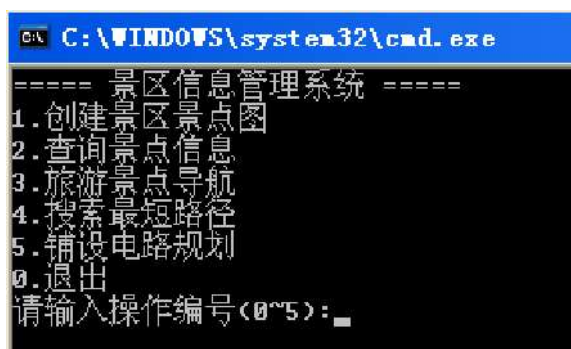
7 运行

7.1 程序导入

- 1、启动 Microsoft Visual Studio 2010。
- 2、选择菜单 File -> Open -> Project/Solution，打开文件选择对话框。
- 3、打开工程目录，选择“ [GraphCPro.sln](#)”文件，点击“打开”，即可打开工程。

7.2 运行结果

按 Ctrl+F5 运行程序，界面如下。



```
C:\WINDOWS\system32\cmd.exe
===== 景区信息管理系统 =====
1. 创建景区景点图
2. 查询景点信息
3. 旅游景点导航
4. 搜索最短路径
5. 铺设电路规划
0. 退出
请输入操作编号(0~5):
```

1、创建图

输入 1，按回车结束。读取文件，创建景区景点图。


```
C:\WINDOWS\system32\cmd.exe

===== 创建景区景点图 =====
顶点数目: 7
----- 顶点 -----
0-A区
1-B区
2-C区
3-D区
4-E区
5-F区
6-G区
----- 边 -----
(v0,v2) 700
(v0,v4) 1000
(v0,v5) 600
(v1,v2) 1000
(v1,v6) 1000
(v2,v3) 400
(v3,v4) 300
(v3,v6) 400
(v4,v5) 600
(v5,v6) 500
```

2、查询景点信息

输入 2，按回车结束，进入“查询景点信息”功能。

输入景点编号，例如 2。程序显示编号为 2 的景点信息，及其周边景区。

```
C:\WINDOWS\system32\cmd.exe

===== 查询景点信息 =====
0-A区
1-B区
2-C区
3-D区
4-E区
5-F区
6-G区
请输入想要查询的景点编号: 2
C区
风景优美，气候宜人。门票20元。
----- 周边景区 -----
C区->A区 700m
C区->B区 1000m
C区->D区 400m
```

3、旅游景点导航

输入 3，按回车结束，进入“旅游景点导航”功能。

输入景点编号，例如 2。程序显示从 2 号景点开始，无回路走遍所有景点的所有路线。

```
C:\WINDOWS\system32\cmd.exe

===== 旅游景点导航 =====
0-A区
1-B区
2-C区
3-D区
4-E区
5-F区
6-G区
请输入起始点编号: 2
导游路线为:
路线1: C区 -> A区 -> F区 -> E区 -> D区 -> G区 -> B区
路线2: C区 -> B区 -> G区 -> D区 -> E区 -> A区 -> F区
路线3: C区 -> B区 -> G区 -> D区 -> E区 -> F区 -> A区
路线4: C区 -> B区 -> G区 -> F区 -> A区 -> E区 -> D区
路线5: C区 -> D区 -> E区 -> A区 -> F区 -> G区 -> B区
```

4、搜索最短路径

输入 4，按回车结束，进入“搜索最短路径”功能。

```
C:\WINDOWS\system32\cmd.exe

===== 搜索最短路径 =====
0-A区
1-B区
2-C区
3-D区
4-E区
5-F区
6-G区
请输入起点的编号: 1
请输入终点的编号: 4
最短路线为: B区->C区->D区->E区
最短距离为: 1700
```

5、铺设电路规划

输入 5，按回车结束，进入“铺设电路规划”功能。

```
C:\WINDOWS\system32\cmd.exe

===== 铺设电路规划 =====
在以下两个景点之间铺设电路:
A区 - F区 600m
F区 - G区 500m
G区 - D区 400m
D区 - E区 300m
D区 - C区 400m
C区 - B区 1000m
铺设电路的总长度为: 3200
```

6、退出程序

输入 0，按回车结束，退出景区信息管理系统。