华北电力大学

**实 验 报 告**

|

|

**实验名称** 线性表结构 (2)

**课程名称**  数据结构综合实践

|

|

专业班级：信息2202 学生姓名：徐梓乔

学 号：220221100327 成 绩：

|  |
| --- |
| 一、实验内容  利用单链表解决城市管理的编程  将若干城市的信息存入一个单链表，结点中的城市信息包括：城市  名称、城市的位置坐标。要求能够利用城市名称和位置坐标进行有  关查找、插入、删除、更新等操作。  基本要求如下：  能在单链表中插入、删除、更新城市信息。  给定一个城市名称，能返回其位置坐标。  给定一个位置坐标P 和一个距离D，返回所有距P 的距离小于D的城市。  二、实验方法与步骤  实验流程：  建立一个单链表，在里面存储城市的信息  具体代码：  单链表插入:  private void cInsert\_Click(object sender, EventArgs e)  {  City city = GetCity();  if (city != null)  {  int num = (int)numInsert.Value;  if (num < 0 || num > \_lst.Length)  {  MessageBox.Show("插入索引越界.", "系统显示", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);  return;  }  \_lst.Insert(num, city);  ShowCityInfo();  }  }  单链表删除:  private void cDelete\_Click(object sender, EventArgs e)  {  int num = (int)numDelete.Value;  if (num < 0 || num > \_lst.Length - 1)  {  MessageBox.Show("删除索引越界.", "系统显示", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);  return;  }  \_lst.Remove(num);  ShowCityInfo();  }  更新城市信息:  private void cUpdate\_Click(object sender, EventArgs e)  {  City city = GetCity();  if (city != null)  {  int num = (int)numUpdate.Value;  if (num < 0 || num > \_lst.Length - 1)  {  MessageBox.Show("插入索引越界.", "系统显示", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);  return;  }  \_lst[num] = city;  ShowCityInfo();  }  }  搜索城市坐标:  private City GetCity()  {  string text = txtCity.Text.Trim();  if (string.IsNullOrEmpty(text))  {  MessageBox.Show("需要填写城市名称!", "提示信息", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);  return null;  }  double num = 0.0;  double num2 = 0.0;  try  {  num = double.Parse(txtX.Text);  num2 = double.Parse(txtY.Text);  }  catch (Exception ex)  {  MessageBox.Show(ex.Message, "提示信息", MessageBoxButtons.OK, MessageBoxIcon.Hand);  return null;  }  return new City(text, new Coordinate(num, num2));  }  给定坐标P和一个距离D,返回所有离该坐标P距离在D以内的城市:  private void btnSearchCity\_Click(object sender, EventArgs e)  {  double num = 0.0;  double num2 = 0.0;  double num3 = 0.0;  try  {  num = double.Parse(txtSearchX.Text);  num2 = double.Parse(txtSearchY.Text);  num3 = double.Parse(txtSearchD.Text);  }  catch (Exception ex)  {  MessageBox.Show(ex.Message, "提示信息", MessageBoxButtons.OK, MessageBoxIcon.Hand);  return;  }  string text = string.Empty;  rtxtCityResult.Clear();  for (int i = 0; i < \_lst.Length; i++)  {  double distance = \_lst[i].Coordinate.GetDistance(new Coordinate(num, num2));  if (distance <= num3)  {  text = text + "D=" + Math.Round(distance, 2) + " " + \_lst[i];  }  }  if (string.IsNullOrEmpty(text))  {  rtxtCityResult.Text = "未找到满足条件的城市。";  }  else  {  rtxtCityResult.Text = text;  }  }  三、实验结果    四、结论与总结  结论:通过单链表能够将城市和坐标储存并且能够管理城市和坐标。  总结:通过单链表可以不要求逻辑上相邻的元素在物理位置上也相邻,并且在插入或者删除元素后不必移动元素,只需要更新结点的下一个指针中存在的地址。 |
|  |