**class Tourist**

{

String Name, Country; int Lions, Rhinos;

public Tourist(String N, String C)

{

Name = N; Country = C; Lions = 0; Rhinos = 0;

}

public Tourist(String N, String C, int R, int L)

{

Name = N; Country = C;

Lions = L;

Rhinos = R;

}

public void setLions(int c)

{

Lions += c;

}

public void setRhinos(int c)

{

Rhinos += c;

}

public String getName()

{

return Name;

}

public String getCountry()

{

return Country;

}

public int getNrLions()

{

return Lions;

}

public int getNrRhinos()

{

return Rhinos;

}

public void addLions(int L)

{

Lions += L;

}

public void addRhinos(int R)

{

Rhinos += R;

}

public void Display()

// displays the Registration Number and Price of a car

{

Console.WriteLine("Name : {0} Country {1} Lions {2} Rhino {3}", Name, Country, Lions, Rhinos);

}

}

**class TouristList**

{

ArrayList List;

String FileName;

**int SortedState; //0 Not Sorted 1 Lions 2 Rhinos 3 Name**

public TouristList(String File)

{

FileName = File;

List = new ArrayList();

ReadData();

**SortedState = 0;**

}

public void Close()

{

WriteData();

}

**public void Add(Tourist NewOne)**

**// Adds a new Tourist to List**

**{**

**List.Add(NewOne);**

**SortedState = 0;**

**}**

public int Count()

{

return List.Count;

}

public void Display()

{

foreach (Tourist cur in List)

{

cur.Display();

}

}

public void Delete(int Pos)

{

List.RemoveAt(Pos - 1);

}

**public int FindName(String Wanted)**

**{**

**if (SortedState == 3)**

**return BinarySearchName(Wanted) + 1;**

**else**

**return LinearSearch(Wanted) + 1;**

**}**

**public void SortLionsDown()**

**// Sorts the list on number of lions seen in descending order, using the   
 // BubbleSort**

**{**

**BubbleSort();**

**SortedState = 1;**

**}**

**public void SortRhinosUp()**

**// Sorts the list on number of rhionos seen in ascending order, using the**

**// SelectionSort**

**{**

**SelectionSort();**

**SortedState = 2;**

**}**

**public void SortNameUp()**

**// Sorts the list on name in ascending order**

**{**

**InsertionSort();**

**SortedState = 3;**

**}**

private void SelectionSort()

{

for (int pass = 1; pass <= List.Count - 1; pass++)

{

int MinPos = 0;

for (int x = 1; x <= List.Count - pass; x++)

if (((Tourist)List[x]).getNrRhinos() > ((Tourist)List[MinPos]).getNrRhinos())

MinPos = x;

swop(Count() - pass, MinPos);

}

}

private void BubbleSort()

{

for (int pass = 1; pass <= List.Count - 1; pass++)

for (int compare = 1; compare <= List.Count - pass; compare++)

{

Tourist first = (Tourist)List[compare - 1];

Tourist second = (Tourist)List[compare];

if (first.getNrLions() < second.getNrLions())

swop(compare - 1, compare);

}

}

private void InsertionSort()

{

for (int x = 1; x <= List.Count - 1; x++)

{

Tourist newOne = (Tourist)List[x];

List.RemoveAt(x);

Insert(newOne, x - 1);

}

}

private void Insert(Tourist Newone, int Last)

// Inserts Newone keeping the list ending at Last, sorted on Price, in ascending order

{

int curPos = Last;

Tourist cur = (Tourist)List[curPos];

while ((curPos != -1) && (Newone.getName().CompareTo(cur.getName()) < 0))

{

curPos--;

if (curPos != -1) cur = (Tourist)List[curPos];

}

List.Insert(curPos + 1, Newone);

}

private int LinearSearch(String Wanted)

// Searches for name in a list not sorted on name

{

for (int x = 0; x <= List.Count; x++)

{

Tourist cur = (Tourist)List[x];

if (Wanted.Equals(cur.getName()))

return x;

}

return -1;

}

**private int BinarySearchName(String wanted)**

**// Searches for name in a list sorted on name in ascending order**

**{**

**int first = 0; int last = List.Count - 1;**

**while (first <= last)**

**{**

**int mid = (first + last) / 2;**

**Tourist cur = (Tourist)List[mid];**

**if (wanted.Equals(cur.getName())) return mid;**

**else if (wanted.CompareTo(cur.getName()) < 0) last = mid - 1;**

**else first = mid + 1;**

**}**

**return -1;**

**}**

public Tourist Get(int Pos)

{

Pos--;

if ((Pos >= 0) && (Pos <= List.Count - 1))

return (Tourist)List[Pos];

else return null;

}

private void ReadData()

{

StreamReader SR;

SR = new StreamReader(FileName);

String Name, Country;

int Lions, Rhinos;

while (!SR.EndOfStream)

{

Name = SR.ReadLine();

Country = SR.ReadLine();

Lions = int.Parse(SR.ReadLine());

Rhinos = int.Parse(SR.ReadLine());

Tourist NewOne = new Tourist(Name, Country, Rhinos, Lions);

Add(NewOne);

}

SR.Close();

}

private void WriteData()

{

StreamWriter SW;

SW = new StreamWriter(FileName);

for (int x = 0; x <= List.Count - 1; x++)

{

Tourist cur = (Tourist)List[x];

SW.WriteLine(cur.getName());

SW.WriteLine(cur.getCountry());

SW.WriteLine(cur.getNrLions());

SW.WriteLine(cur.getNrRhinos());

}

SW.Close();

}

private void swop(int x, int y)

{

object temp = List[x];

List[x] = List[y];

List[y] = temp;

}

}

**class Program**

{

static void Main(string[] args)

{

TouristList List = new TouristList("Data.Dat");

List.SortNameUp();

List.Display();

int Pos = List.FindName("Jean");

Console.WriteLine(Pos); Console.ReadLine();

//MostAnimals(List);

//TotalLions(List);

//DisplaySorted(List);

}

static void MostAnimals(TouristList List)

{

int Max = 0; String MaxName = "";

for (int x = 1; x <= List.Count(); x++)

{

Tourist cur = List.Get(x);

int nr = cur.getNrLions() + cur.getNrRhinos();

if (nr > Max)

{

Max = nr; MaxName = cur.getName();

}

}

Console.WriteLine("Best tourist {0}", MaxName);

}

static void TotalLions(TouristList List)

{

int count = 0;

for (int x = 1; x <= List.Count(); x++)

{

Tourist cur = List.Get(x);

count += cur.getNrLions();

}

Console.WriteLine("Total lions {0}", count);

}

static void DisplaySorted(TouristList List)

// Display tourists, sorted on Rhinos spotted, in ascending order

{

List.SortRhinosUp();

List.Display();

}

static void increaseLionsView(TouristList List)

// Request the name of the tourist and how many new rhinos he/she observed

// Update the list

{

Console.WriteLine("Name");

String Name = Console.ReadLine();

int pos = List.FindName(Name);

if (pos != 0)

{

Console.WriteLine("How many saw");

int count = int.Parse(Console.ReadLine());

Tourist cur = List.Get(pos);

int curcount = cur.getNrLions();

cur.setLions(count + curcount);

}

}

}