```
 1  using System;
 2  using System.IO;
 3  using System.Diagnostics;
 4
 5  namespace BubbleSort
 6  {
 7      class Bubble_Sort
 8      {
 9          static void Main(string[] args)
10          {
11              int seed = (int)DateTime.Now.Ticks & 0x0000FFFF;
12
13              // Antras etapas
14              Test_File_Array_List(seed);
15
16          }
17          public static void BubbleSort(DataArray items)
18          {
19              double prevdata, currentdata;
20              for (int i = items.Length - 1; i >= 0; i--)
21              {
22                  currentdata = items[0];
23                  for (int j = 1; j <= i; j++)
24                  {
25                      prevdata = currentdata;
26                      currentdata = items[j];
27                      if (prevdata > currentdata)
28                      {
29                          items.Swap(j, currentdata, prevdata);
30                          currentdata = prevdata;
31                      }
32                  }
33              }
34          }
35
36          public static void BubbleSort(DataList items)
37          {
38              double prevdata, currentdata;
39              for (int i = items.Length - 1; i >= 0; i--)
40              {
41                  currentdata = items.Head();
42                  for (int j = 1; j <= i; j++)
43                  {
44                      prevdata = currentdata;
45                      currentdata = items.Next();
46                      if (prevdata > currentdata)
47                      {
48                          items.Swap(currentdata, prevdata);
49                          currentdata = prevdata;
50                      }
51                  }
52              }
```

```
53          }
54
55          public static void Test_File_Array_List(int seed)
56          {
57              int n = 12;
58              string filename;
59              filename = @"mydataarray.dat";
60              MyFileArray myfilearray = new MyFileArray(filename, n, seed);
61              using (myfilearray.fs = new FileStream(filename, FileMode.Open,
                  FileAccess.ReadWrite))
62              {
63                  Console.WriteLine("\n FILE ARRAY \n");
64                  myfilearray.Print(n);
65                  BubbleSort(myfilearray);
66                  myfilearray.Print(n);
67              }
68              filename = @"mydatalist.dat";
69              MyFileList myfilelist = new MyFileList(filename, n, seed);
70              using (myfilelist.fs = new FileStream(filename, FileMode.Open,
                  FileAccess.ReadWrite))
71              {
72                  Console.WriteLine("\n FILE LIST \n");
73                  myfilelist.Print(n);
74                  BubbleSort(myfilelist);
75                  myfilelist.Print(n);
76              }
77          }
78
79      }
80      abstract class DataArray
81      {
82          protected int length;
83          public int Length { get { return length; } }
84          public abstract double this[int index] { get; }
85          public abstract void Swap(int j, double a, double b);
86          public void Print(int n)
87          {
88              for (int i = 0; i < n; i++)
89                  Console.Write(" {0:F5} ", this[i]);
90              Console.WriteLine();
91          }
92      }
93
94      abstract class DataList
95      {
96          protected int length;
97          public int Length { get { return length; } }
98          public abstract double Head();
99          public abstract double Next();
100         public abstract void Swap(double a, double b);
101         public void Print(int n)
102         {
```

```
103             Console.Write(" {0:F5} ", Head());
104             for (int i = 1; i < n; i++)
105                 Console.Write(" {0:F5} ", Next());
106             Console.WriteLine();
107         }
108     }
109 }
110
```

```
1  using System;
2  using System.IO;
3
4  namespace BubbleSort
5  {
6       class MyFileArray : DataArray
7      {
8          public MyFileArray(string filename, int n, int seed)
9          {
10             double[] data = new double[n];
11             length = n;
12             Random rand = new Random(seed);
13             for (int i = 0; i < length; i++)
14             {
15                 data[i] = rand.NextDouble();
16             }
17             if (File.Exists(filename)) File.Delete(filename);
18             try
19             {
20                 using (BinaryWriter writer = new BinaryWriter(File.Open(filename,
                     FileMode.Create)))
21                 {
22                     for (int j = 0; j < length; j++)
23                         writer.Write(data[j]);
24                 }
25             }
26             catch (IOException ex)
27             {
28                 Console.WriteLine(ex.ToString());
29             }
30         }
31
32         public FileStream fs { get; set; }
33
34         public override double this[int index]
35         {
36             get
37             {
38                 Byte[] data = new Byte[8];
39                 fs.Seek(8 * index, SeekOrigin.Begin);
40                 fs.Read(data, 0, 8);
41                 double result = BitConverter.ToDouble(data, 0);
42                 return result;
43             }
44         }
45         public override void Swap(int j, double a, double b)
46         {
47             Byte[] data = new Byte[16];
48             BitConverter.GetBytes(a).CopyTo(data, 0);
49             BitConverter.GetBytes(b).CopyTo(data, 8);
50             fs.Seek(8 * (j - 1), SeekOrigin.Begin);
51             fs.Write(data, 0, 16);
```

```
52          }
53      }
54  }
55
```

```
1  using System;
2  using System.IO;
3
4  namespace BubbleSort
5  {
6       class MyFileList : DataList
7      {
8          int prevNode;
9          int currentNode;
10         int nextNode;
11
12         public MyFileList(string filename, int n, int seed)
13         {
14             length = n;
15             Random rand = new Random(seed);
16             if (File.Exists(filename)) File.Delete(filename);
17             try
18             {
19                 using (BinaryWriter writer = new BinaryWriter(File.Open(filename,
                     FileMode.Create)))
20                 {
21                     writer.Write(4);
22                     for (int j = 0; j < length; j++)
23                     {
24                         writer.Write(rand.NextDouble());
25                         writer.Write((j + 1) * 12 + 4);
26                     }
27                 }
28             }
29             catch (IOException ex)
30             {
31                 Console.WriteLine(ex.ToString());
32             }
33         }
34         public FileStream fs { get; set; }
35         public override double Head()
36         {
37             Byte[] data = new Byte[12];
38             fs.Seek(0, SeekOrigin.Begin);
39             fs.Read(data, 0, 4);
40             currentNode = BitConverter.ToInt32(data, 0);
41             prevNode = -1;
42             fs.Seek(currentNode, SeekOrigin.Begin);
43             fs.Read(data, 0, 12);
44             double result = BitConverter.ToDouble(data, 0);
45             nextNode = BitConverter.ToInt32(data, 8);
46             return result;
47         }
48         public override double Next()
49         {
50             Byte[] data = new Byte[12];
51             fs.Seek(nextNode, SeekOrigin.Begin);
```

```
52              fs.Read(data, 0, 12);
53              prevNode = currentNode;
54              currentNode = nextNode;
55              double result = BitConverter.ToDouble(data, 0);
56              nextNode = BitConverter.ToInt32(data, 8);
57              return result;
58          }
59
60          public override void Swap(double a, double b)
61          {
62              Byte[] data;
63              fs.Seek(prevNode, SeekOrigin.Begin);
64              data = BitConverter.GetBytes(a);
65              fs.Write(data, 0, 8);
66              fs.Seek(currentNode, SeekOrigin.Begin);
67              data = BitConverter.GetBytes(b);
68              fs.Write(data, 0, 8);
69          }
70
71      }
72  }
73
```