
File I/O, Arrays

9/20/05

CS360 Windows Programming

1

Last Time

- We began looking at GUI Programming
- Completed talking about exception handling

9/20/05

CS360 Windows Programming

2

File Input/Output

- .NET Framework supports file I/O in the System.IO namespace
- What is the difference between files and streams
 - File: collection of data stored on a disk with a name and directory path
 - Stream: once a file is opened for reading or writing it becomes a stream. A stream is something on which you can perform read and write operations

9/20/05

CS360 Windows Programming

3

Text File Example

```
using System;
using System.IO;
public class TextToFile
{
    private const string FILE_NAME = "MyFile.txt";
    public static void Main(String[] args)
    {
        if (File.Exists(FILE_NAME))
        {
            Console.WriteLine("{0} already exists.", FILE_NAME);
            return;
        }
        StreamWriter sr = File.CreateText(FILE_NAME);
        sr.WriteLine ("This is my file.");
        sr.WriteLine ("I can write ints {0} or floats {1}, and
so
on.", 1, 4.2);
        sr.Close();
    }
}
}9/20/05
```

CS360 Windows Programming

4

Another Example

```
using System;
using System.IO;
public class StreamsIOApp
{
    public static void Main(string[] args)
    {
        StreamWriter s = new StreamWriter("Foo.txt");
        s.Write("some text or other");
        s.Close();

        StreamReader r = new StreamReader("Foo.txt");
        for( string line = r.ReadLine(); line != null;
            line = r.ReadLine() )
            Console.WriteLine( line );
        r.Close();
    }
}
}9/20/05
```

CS360 Windows Programming

5

File I/O

- There are many ways to open and access files
- There is no best way
- The .NET Framework class library contains a complete list of file and stream classes
- See page 62

9/20/05

CS360 Windows Programming

6

ASCII vs. Binary Files

- What is the difference between ASCII and binary files?
- Why do we use binary files?

9/20/05

CS360 Windows Programming

7

Number Systems

- Hexadecimal, decimal, and binary
- How do we convert from binary to decimal?
- How do we convert from decimal to binary?
- How do we convert from binary to hex?
- Convert the decimal 512 to Hex
 - Convert 512 first to binary
 - Convert the binary to hex

9/20/05

CS360 Windows Programming

8

Text Files

- What are characters in a text file stored as?
- What is 62 stored as?
- <http://www.lookuptables.com/>

9/20/05

CS360 Windows Programming

9

Example

- 32767 represented as a text file:

00110011	00110010	00110111	00110110	00110111
----------	----------	----------	----------	----------

- Why?

00110011	00110010	00110111	00110110	00110111
51	50	55	54	55
3	2	7	6	7

Binary

Decimal

Character

9/20/05

CS360 Windows Programming

10

Example

- Decimal 32767 represented as a binary file:

01111111	11111111
----------	----------

- What is that in Hex?

9/20/05

CS360 Windows Programming

11

Binary and Text Files

- The difference between binary and text files is the way they write data to the files
- In binary files, the binary representation of values is written to the file
 - The integer '4929067' which takes 4 bytes in memory will also take 4 bytes in the file
- In case of text file, each value is written as a series of characters (ASCII or Unicode)
 - The integer '4929067' will be written as text and will take 7 bytes in ASCII encoding and 14 (7 x 2) bytes in Unicode encoding
- Binary files are more efficient for reading and writing of data for machines while text files are more human readable.

9/20/05

CS360 Windows Programming

12

Example to Write a Binary File

```
Stream s = new FileStream("Foo.txt", FileMode.Create);
StreamWriter w = new StreamWriter(s);
w.Write("Hello World ");
w.Write(32767);
w.Write(' ');
w.Write(45.67);
w.Close();
s.Close();

Stream t = new FileStream("Bar.dat", FileMode.Create);
BinaryWriter b = new BinaryWriter(t);
b.Write("Hello World ");
b.Write(512);
b.Write(' ');
b.Write(45.67);
b.Close();
t.Close();
```

9/20/05

CS360 Windows Programming

13

Hex Dump

- Write a program that will read a file and output the hex values of that file

9/20/05

CS360 Windows Programming

14

Arrays

- Single dimension

```
int[] myInts = new int[20];
...
Console.WriteLine(myInts[i]);
```

- Multidimension

```
string[,] myStrings = new string[5,6];
double[, ,] myDoubles = new double[3,8,5];
...
Console.WriteLine(myDoubles[i,j,k]);
```

- Jagged

```
Point[][] myPolygons = new Point[3][];
myPolygons[0] = new Point[10];
myPolygons[1] = new Point[20];
myPolygons[2] = new Point[30];
...
for (int x = 0; x < myPolygons[1].Length; x++)
    Console.WriteLine(myPolygons[1][x]);
```

9/20/05

CS360 Windows Programming

15

```
Point[][] myPolygons = new Point[3][];
myPolygons[0] = new Point[10];
myPolygons[1] = new Point[20];
myPolygons[2] = new Point[30];
```

```
for (int x = 0;
     x < myPolygons[1].Length; x++)
    Console.WriteLine(myPolygons[1][x]);
foreach (Point p in myPolygons[1])
    Console.WriteLine(p);
```

Note: inside a `foreach` loop we have read only access to the array elements.

9/20/05

CS360 Windows Programming

16

- `int[] myInts;`
 - Creates a variable that can point to an array
- `myInts = new int[100];`
 - Creates an array of 100 ints.
 - These ints are initialized to 0, and stored, unboxed, in a memory block on the managed heap.
- `Control myControls;`
 - Creates a variable that can point to an array
- `myControls = new Control[100];`
 - Creates an array of Control references, initialized to null. Since Control is a reference type, creating the array creates references—the actual objects are not created.

9/20/05

CS360 Windows Programming

17

Array Initialization

- Single dimension
 - `int[] myInts = new int[3] {1, 2, 3};`
 - `int[] myInts = new int[] {1, 2, 3};`
 - `int[] myInts = {1, 2, 3};`

9/20/05

CS360 Windows Programming

18

Arrays are implicitly derived from System.Array

- `int[] myInts = new int[3] {1, 2, 3};`
- `double[, ,] myDoubles = new double[3,8,5];`
- Properties
 - `myInts.Length` is 3
 - `myInts.Rank` is 1
 - `myDoubles.Length` is 120
 - `myDoubles.Rank` is 3
 - `myDoubles.GetLength(1)` is 8
- Static Methods
 - `Array.Sort(myInts);`
 - `Array.Sort(keys, items);`
 - `Array.Reverse(myInts);`
 - `Array.Clear(myInts);`
 - `int i = Array.IndexOf(myInts, 17);`

9/20/05

CS360 Windows Programming

19

Summary

- Completed File I/O
- Arrays
- Covered p. 59-66

9/20/05

CS360 Windows Programming

20