FileSystem

- _contents: List<Thing>

+ FileSystem()
+ Add(Thing toAdd)
+ PrintContent()

Use

<> Thing

- _name: String

+ Thing(string name)
+ Size(); int <>
+ Print() <>

+ Name : String <<read-only property>>

Folder

- _contents : List<Thing>

+ Folder(string name)
+ Add(Thing toAdd)
+ Size(); int <<override>>
+ Print() <<override>>

File

- _extension : String
- _size : Int

+ File(string name, string extension, int size)
+ Size(); int <<override>>
+ Print() <<override>>

*

```csharp
 1  using System;
 2  using System.Collections.Generic;
 3  using System.Linq;
 4  using System.Text;
 5  using System.Threading.Tasks;
 6
 7  namespace SemTest
 8  {
 9      public abstract class Thing
10      {
11          private string _name;
12
13          public Thing(string name)
14          {
15              _name = name;
16          }
17
18          public abstract int Size();
19
20          public abstract void Print();
21
22          public string Name
23          {
24              get { return _name; }
25              set { _name = value; }
26          }
27      }
28  }
29
```

```csharp
 1 using System;
 2 using System.Collections.Generic;
 3 using System.Linq;
 4 using System.Text;
 5 using System.Threading.Tasks;
 6
 7 namespace SemTest
 8 {
 9     public class Folder : Thing
10     {
11         private List<Thing> _contents;
12
13         public Folder(string name) : base(name)
14         {
15             _contents = new List<Thing>();
16         }
17
18         public void Add(Thing toAdd)
19         {
20             _contents.Add(toAdd);
21         }
22
23         public override int Size()
24         {
25             int size = 0;
26             foreach (Thing thing in _contents)
27             {
28                 size += thing.Size();
29             }
30             return size;
31         }
32
33         public override void Print()
34         {
35             Console.WriteLine("Folder: {0} ({1} bytes)", Name, Size());
36             foreach (Thing item in _contents)
37             {
38                 item.Print();
39             }
40         }
41     }
42 }
43
```

```csharp
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace SemTest
8  {
9      public class File : Thing
10     {
11         private string _extension;
12         private int _size;
13
14         public File(string name, string extension, int size) : base(name)
15         {
16             _extension = extension;
17             _size = size;
18         }
19
20         public override int Size()
21         {
22             return _size;
23         }
24
25         public override void Print()
26         {
27             Console.WriteLine("File: {0}.{1} ({2} bytes)", Name,
                    _extension, _size);
28         }
29     }
30 }
31
```

```csharp
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace SemTest
8  {
9      public class FileSystem
10     {
11         private List<Thing> _contents;
12
13         public FileSystem()
14         {
15             _contents = new List<Thing>();
16         }
17
18         public void Add(Thing toAdd)
19         {
20             _contents.Add(toAdd);
21         }
22
23         public void PrintContents()
24         {
25             foreach (var item in _contents)
26             {
27                 item.Print();
28             }
29         }
30     }
31 }
32
```

```csharp
1  namespace SemTest
2  {
3      public class Program
4      {
5          static void Main(string[] args)
6          {
7              FileSystem fileSystem = new FileSystem();
8
9              File file1 = new File("Document1", "txt", 100);
10             File file2 = new File("Image1", "jpg", 2000);
11
12             fileSystem.Add(file1);
13             fileSystem.Add(file2);
14
15             Folder folder1 = new Folder("Folder1");
16             folder1.Add(new File("FileInFolder1", "doc", 300));
17
18             fileSystem.Add(folder1);
19
20             Folder folder2 = new Folder("Folder2");
21             Folder subFolder = new Folder("SubFolder");
22             subFolder.Add(new File("FileInSubFolder", "pdf", 400));
23             folder2.Add(subFolder);
24
25             fileSystem.Add(folder2);
26
27             Folder emptyFolder = new Folder("EmptyFolder");
28
29             fileSystem.Add(emptyFolder);
30
31             fileSystem.PrintContents();
32         }
33     }
34 }
35
```

```
File: Document1.txt (100 bytes)
File: Image1.jpg (2000 bytes)
Folder: Folder1 (300 bytes)
File: FileInFolder1.doc (300 bytes)
Folder: Folder2 (400 bytes)
Folder: SubFolder (400 bytes)
File: FileInSubFolder.pdf (400 bytes)
Folder: EmptyFolder (0 bytes)


D:\workspace\COS20007\COS20007-working\SemTest\SemTest\bin\Debug\net8.0\SemTest.exe (process 11936) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console w
hen debugging stops.
Press any key to close this window . . .
```