Буниф Мажда

18-КБ-ПР2

**Отчет КПО Lab 7**

**3 Задание**

5) Задание выполняется совместно несколькими студентами (2-3 человека в группе).

6) Реализовать составную структуру для представления сущностей, указанных в варианте задания из пункта 4. При разработке использовать паттерн Composite для составных структур.

7) Составить тесты к методам класса, проверяющие корректность их работы.

8) Запустить тестирование и проверить результаты работы.

9) Написать программу, демонстрирующую работу класса, осуществляющее визуальное представление результатов.

10) Составить отчет.

**4 Варианты заданий**

1) Каталог (на диске). Может содержать два типа сущностей: другие каталоги и файлы. Необходимо определять суммарный объем любого каталога (размеры файлов задаются при создании).

**Листинг :**

**Component.cs**

public abstract class Component

{

protected string Name { get; set; }

public Component(string name)

{

Name = name;

}

public virtual int Size { get; }

public virtual void Add(Component component) { }

public virtual void Remove(Component component) { }

public virtual void Replace(Component component) { }

public virtual void Space() { }

public virtual void Print()

{

Console.WriteLine(Name + " (" + Size + "МБ)");

}

}

**Directory.cs**

public class Directory : Component

{

private List<Component> components;

private int size;

public Directory(string name) : base(name)

{

components = new List<Component>();

}

public override int Size

{

get => size;

}

public override void Add(Component component)

{

components.Add(component);

}

public override void Remove(Component component)

{

components.Remove(component);

}

public override void Space()

{

size = 0;

foreach (var i in components)

{

i.Space();

if (i is Directory || i is File)

size += i.Size;

}

Console.WriteLine(Name + " (" + size + "МБ)");

}

public override void Print()

{

Console.WriteLine(Name);

for (int i = 0; i < components.Count; i++)

{

components[i].Print();

}

}

}

}

**File.cs**

public class File : Component

{

public override int Size { get; }

public override void Space()

{

}

public override void Print()

{

Console.WriteLine("--" + Name + " (" + Size + "МБ)");

}

public File(string name, int size) : base(name)

{

Size = size;

}

}

**Program.cs**

class Program

{

static void Main(string[] args)

{

Component mylap = new Directory("Computer admin");

Component disc = new Directory("disc C");

Component imageFile = new File("img.jpeg", 12);

Component textFile = new File("word.txt", 10);

disc.Add(imageFile);

disc.Add(textFile);

mylap.Add(disc);

mylap.Print();

mylap.Space();

Console.WriteLine();

disc.Remove(imageFile);

disc.Print();

mylap.Space();

Console.ReadKey();

// here I could create another directory and another files but I prefer to use it in tests

}

}

**Tests.cs**

[TestFixture]

public class TestsClass

{

[Test]

public void LaptopSpace()

{

Component mylap = new Directory("computer admin");

Component disk = new Directory("Disk C");

Component fileTxt = new File("Text.txt", 10);

Component fileJpeg = new File("img.jpeg", 12);

disk.Add(fileTxt);

disk.Add(fileJpeg);

mylap.Add(disk);

mylap.Space();

var actual = mylap.Size;

var expected = 22;

Assert.AreEqual(actual, expected);

}

[Test]

public void DiskSpaceTest()

{

Component mylap = new Directory("computer admin");

Component disk1 = new Directory("Disk C");

Component disk2 = new Directory("Disk D");

Component fileTxt = new File("Text.txt", 10);

Component fileJpeg = new File("img.jpeg", 12);

disk2.Add(fileJpeg);

disk1.Add(fileTxt);

mylap.Add(disk2);

disk1.Space();

var actual = disk1.Size;

var expected = 10;

Assert.AreEqual(actual, expected);

}

[Test]

public void RemoveFileTest()

{

Component disk = new Directory("Disk E");

Component fileTxt = new File("Text.txt", 10);

Component filePng = new File("Png.png", 7);

Component fileDoc = new File("Doc.docx", 9);

disk.Add(fileTxt);

disk.Add(filePng);

disk.Add(fileDoc);

disk.Remove(filePng);

disk.Remove(fileDoc);

disk.Space();

var actual = disk.Size;

var expected = 10;

Assert.AreEqual(actual, expected);

}

[Test]

public void AddDisksInPCTest()

{

Component mylap = new Directory("computer admin");

Component disk1 = new Directory("Disk C");

Component disk2 = new Directory("Disk D");

Component fileTxt = new File("Text.txt", 10);

Component filePng = new File("Png.png", 15);

Component fileDoc = new File("Doc.docx", 9);

disk1.Add(fileTxt);

disk2.Add(filePng);

disk1.Add(fileDoc);

mylap.Add(disk1);

mylap.Add(disk2);

mylap.Space();

var actual = mylap.Size;

var expected = 34;

Assert.AreEqual(actual, expected);

}

}

**Результат Работы :**

