Consider the problem below:

Say we have a FileReader class where the file can be read based on the combination of applying any of the formulas like it could be

* Zipped.
* Encrypted.
* Zipped and encrypted.

encrypted then zipped and ecrypted again We can apply any combination as and when needed and also new methods of encryption is very simple. Just add the new class as sibling of EncryptedReader.

* 

|  |
| --- |
| **DecoratorEx** |
| **Line** | **Code** |
| **1****2****3****4****5****6****7****8****9****10****11****12****13****14****15****16****17****18****19****20****21****22****23****24****25****26****27****28****29****30****31****32****33****34****35****36****37****38****39****40****41****42****43****44****45****46****47****48****49****50****51****52****53****54****55****56****57****58****59** | **using** System;**namespace** DecoratorEx { **interface** IReader { **string** Read(); } **class** FileReader : IReader { **public string** Read() { //Console.Write("Data>>"); //return Console.ReadLine().Trim(); // or //Read from File..... **return** "Data"; } } **abstract class** ReaderDecorator : IReader { **protected** IReader reader; **abstract public string** Read(); **public** ReaderDecorator(IReader reader) { **this**.reader = reader; } } **class** ZipReader : ReaderDecorator { **public** ZipReader(IReader r) : **base**(r) { } **override public string** Read() { **return** "Zip("+ reader.Read() + ")"; } } **class** EncryptedReader : ReaderDecorator { **public** EncryptedReader(IReader r) : **base**(r) { } **override public string** Read() { **return** "Enc(" + reader.Read() + ")"; } } **class** Program { **static void** Main(**string**[] args) { IReader r = **new** FileReader(); IReader zip = **new** ZipReader(r); IReader enc = **new** EncryptedReader(r); IReader zipzip = **new** ZipReader(zip); IReader zipenc = **new** ZipReader(enc); IReader enczipzip = **new** EncryptedReader(zipzip); Console.WriteLine("r:{0}", r.Read()); Console.WriteLine("zip:{0}", zip.Read()); Console.WriteLine("enc:{0}", enc.Read()); Console.WriteLine("zipzip:{0}", zipzip.Read()); Console.WriteLine("zipenc:{0}", zipenc.Read()); Console.WriteLine("enczipzip:{0}",enczipzip.Read()); Console.ReadKey(); } }} |