

DESIGN PATTERNS IN C# MADE SIMPLE

MODULE 9 Treating Collection as an Object with the Composite Pattern

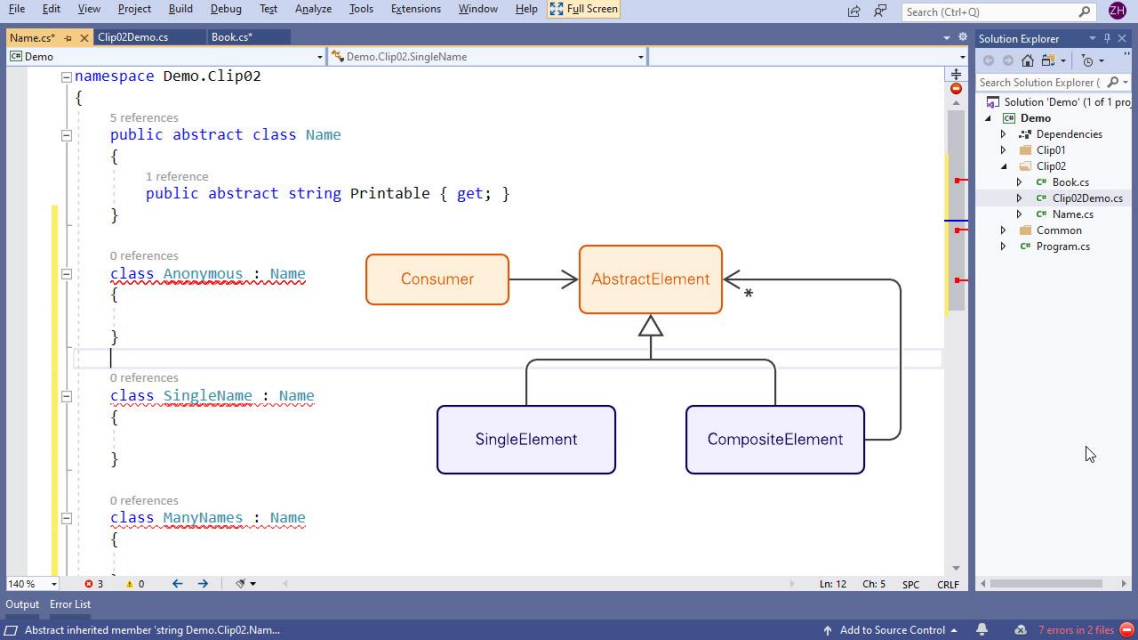


ZORAN HORVAT
CEO AT CODING HELMET

<http://codinghelmet.com>

zh@codinghelmet.com

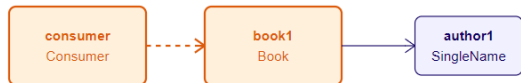
 zoranh75



Common Composite Implementation

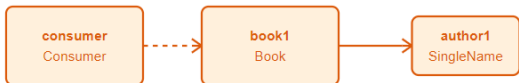


Common Composite Implementation



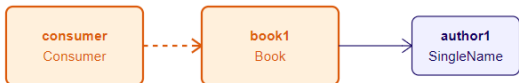
consumer calls `book1.ToString()`

Common Composite Implementation



`author1.Printable` returns "Bertrand Meyer"

Common Composite Implementation

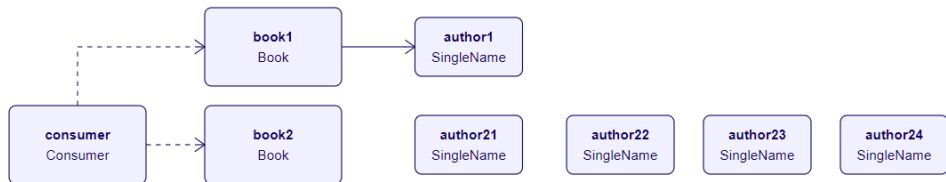


`book1.ToString()` returns "Bertrand Meyer, Object-Oriented Software Construction"

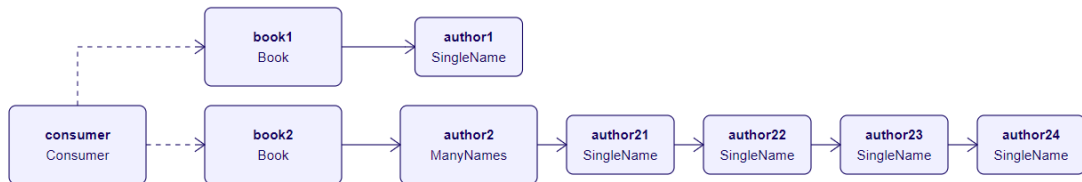
Common Composite Implementation



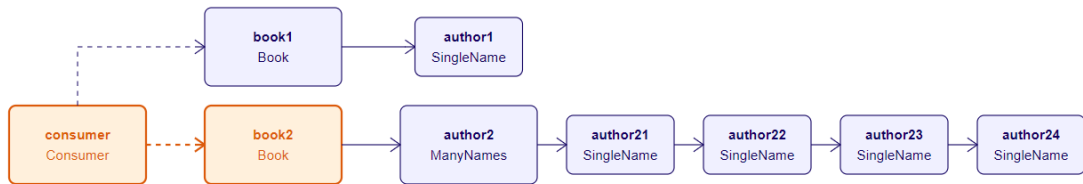
Common Composite Implementation



Common Composite Implementation

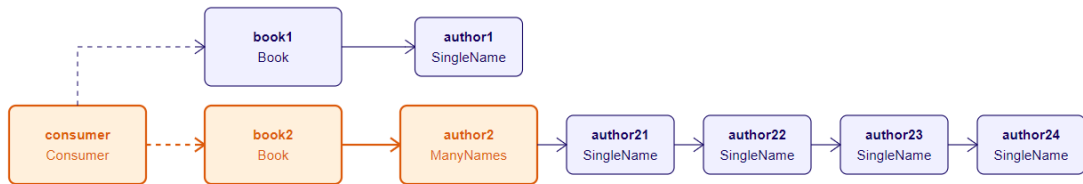


Common Composite Implementation



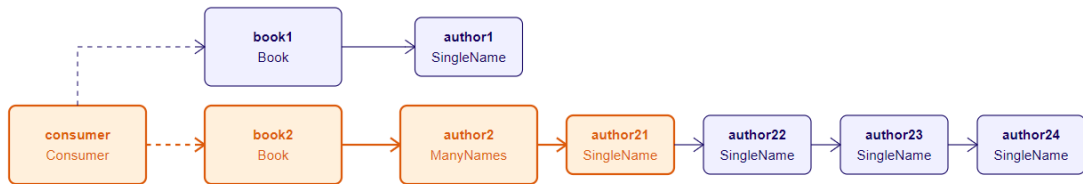
consumer calls `book2.ToString()`

Common Composite Implementation



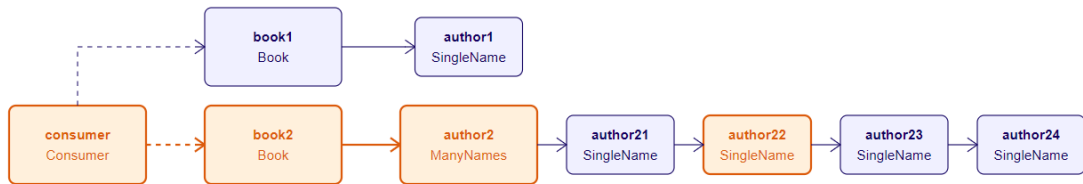
book2 calls author2.Printable

Common Composite Implementation



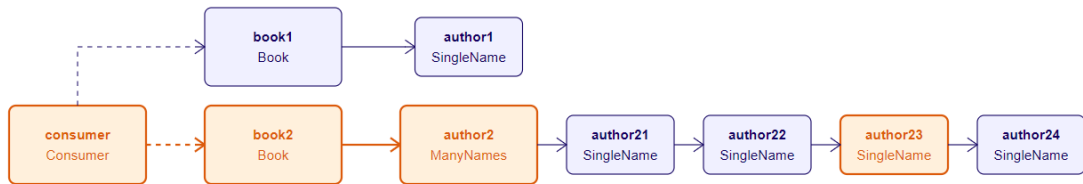
author21.Printable returns "Erich Gamma"

Common Composite Implementation



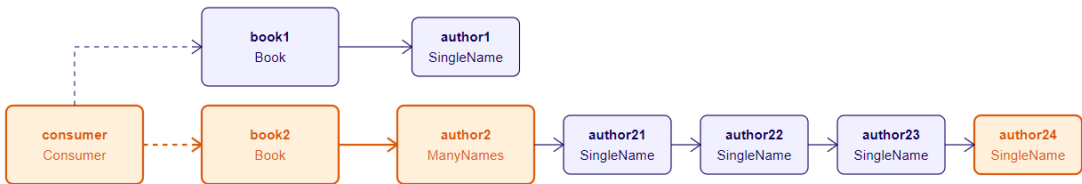
author22.Printable returns "Richard Helm"

Common Composite Implementation



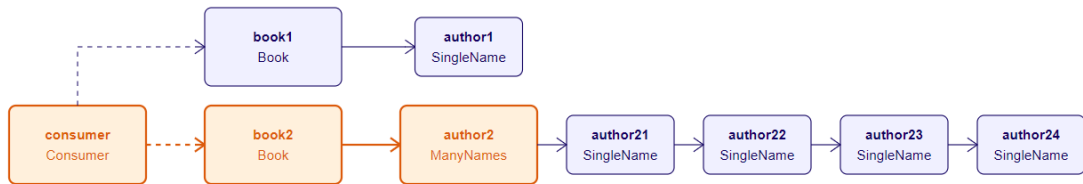
`author23.Printable` returns "Ralph Johnson"

Common Composite Implementation



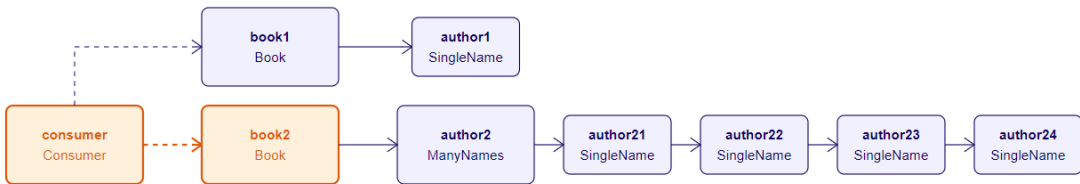
author24.Printable returns "John Vlissides"

Common Composite Implementation



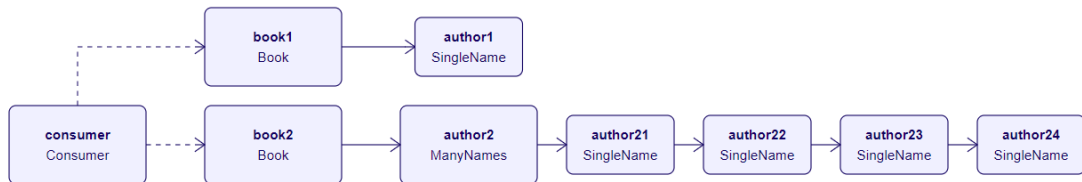
`author2.Printable` returns "Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides"

Common Composite Implementation

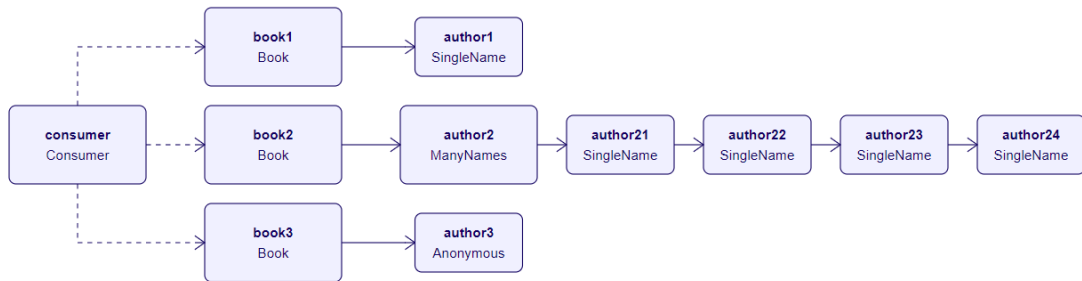


book2.ToString() returns "Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Design Patterns"

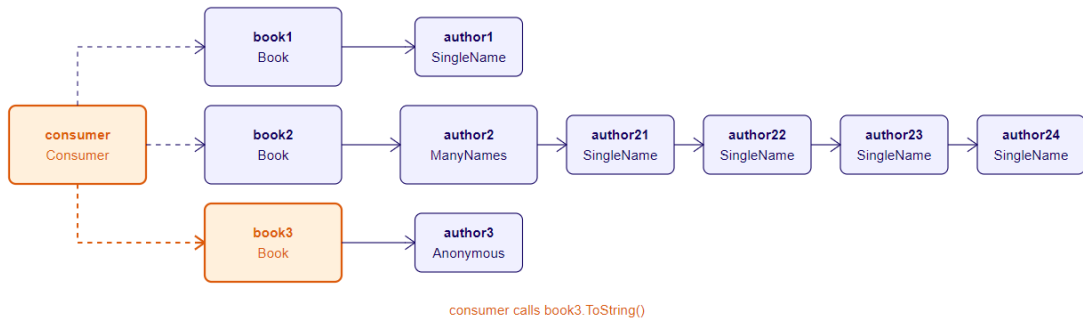
Common Composite Implementation



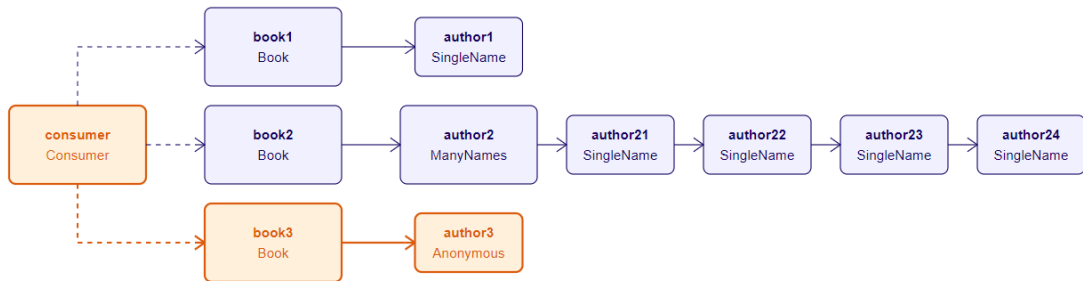
Common Composite Implementation



Common Composite Implementation

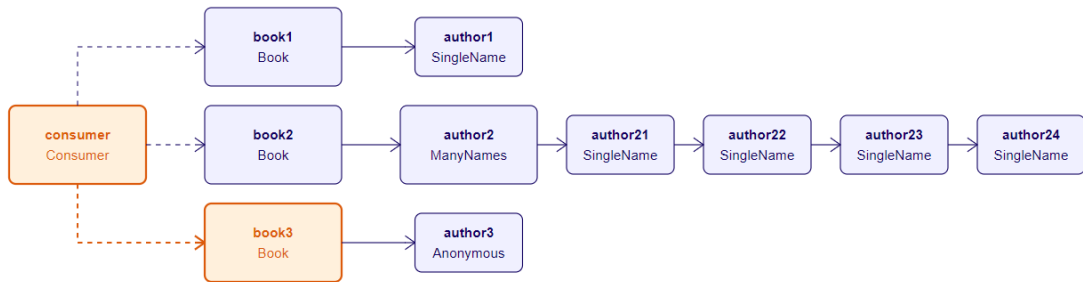


Common Composite Implementation



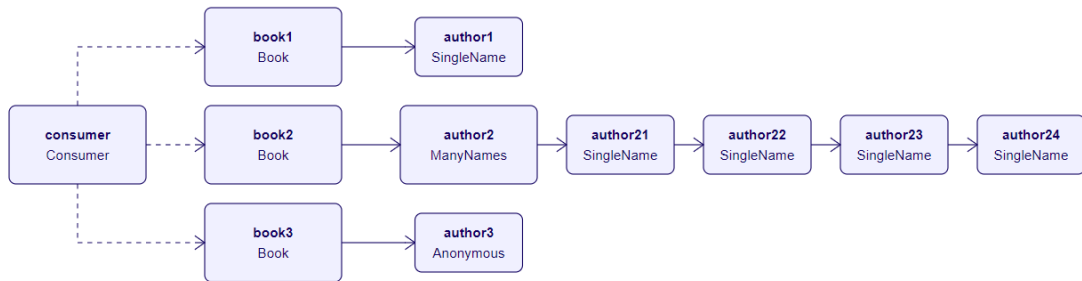
author3.Printable returns "Anonymous"

Common Composite Implementation



`book3.ToString()` returns "Anonymous, One Thousand and One Nights"

Common Composite Implementation



FileEditViewProjectBuildDebugTestAnalyzeToolsExtensionsWindowHelpFull Screen

ManyNames.csClip03Demo.cs*Book.cs

DemoClip03.Clip03DemoImplementation()

```
Book oosc = new Book(  
    "Object-Oriented Software Construction",  
    new SingleName("Bertrand Meyer"));  
  
Book patterns = new Book(  
    "Design Patterns",  
    new ManyNames(new Name[] {  
        new ManyNames(new [] {  
            new SingleName("Erich Gamma"), new SingleName("Richard Helm") }},  
        new ManyNames(new [] {  
            new SingleName("Ralph Johnson"), new SingleName("John Vlissides") })  
        }  
    ));
```

ManyNames

Printable: , , ,

140 %No issues foundLn: 26Ch: 97SPCCRLF

OutputError List

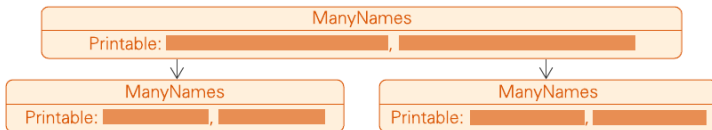
ReadyAdd to Source Control

Solution Explorer

Search Solution Explorer (Ctrl+Q)

Solution 'Demo' (1 of 1 projects)
Demo
Dependencies
Clip01
Clip02
Clip03
Names
Anonymous.cs
ManyNames.cs
SingleName.cs
Book.cs
Clip03Demo.cs
Name.cs
Common
Program.cs


```
Book oosc = new Book(  
    "Object-Oriented Software Construction",  
    new SingleName("Bertrand Meyer"));  
  
Book patterns = new Book(  
    "Design Patterns",  
    new ManyNames(new Name[] {  
        new ManyNames(new [] {  
            new SingleName("Erich Gamma"), new SingleName("Richard Helm") }},  
        new ManyNames(new [] {  
            new SingleName("Ralph Johnson"), new SingleName("John Vlissides") })  
        }));
```



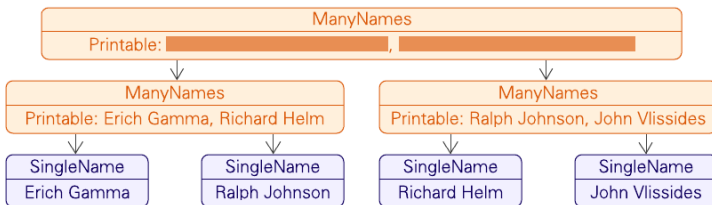
- Demo
 - Dependencies
 - Clip01
 - Clip02
 - Clip03
 - Names
 - Anonymous.cs
 - ManyNames.cs
 - SingleName.cs
 - Book.cs
 - Clip03Demo.cs
 - Name.cs
 - Common
 - Program.cs

ManyNames.cs Clip03Demo.cs* Book.cs

Demo Demo.Clip03.Clip03Demo Implementation()

```
Book oosc = new Book(
    "Object-Oriented Software Construction",
    new SingleName("Bertrand Meyer"));

Book patterns = new Book(
    "Design Patterns",
    new ManyNames(new Name[] {
        new ManyNames(new [] {
            new SingleName("Erich Gamma"), new SingleName("Richard Helm") }),
        new ManyNames(new [] {
            new SingleName("Ralph Johnson"), new SingleName("John Vlissides") })
    }));
```



Solution Explorer

Search Solution Explorer (Ctrl+Q)

- Solution 'Demo' (1 of 1 projects)
- Dependencies
 - Clip01
 - Clip02
 - Clip03
 - Names
 - Anonymous...
 - ManyNames...
 - SingleName...
 - Book.cs
 - Clip03Demo.cs
 - Name.cs
 - Common
 - Program.cs

140 % No issues found

Ln: 26 Ch: 97 SPC CRLF

Output Error List

Ready

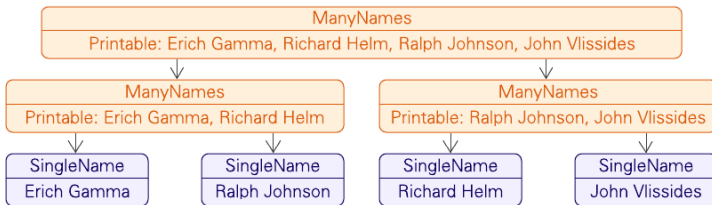
Add to Source Control

```

Demo
    Demo.Clip03.Clip03Demo
        Implementation()

Book oosc = new Book(
    "Object-Oriented Software Construction",
    new SingleName("Bertrand Meyer"));

Book patterns = new Book(
    "Design Patterns",
    new ManyNames(new Name[] {
        new ManyNames(new [] {
            new SingleName("Erich Gamma"), new SingleName("Richard Helm") }),
        new ManyNames(new [] {
            new SingleName("Ralph Johnson"), new SingleName("John Vlissides") })
    }));
    
```



Search Solution Explorer (Ctrl+Q)

- Solution 'Demo' (1 of 1 projects)
 - Dependencies
 - Clip01
 - Clip02
 - Clip03
 - Names
 - Anonymous.cs
 - ManyNames.cs
 - SingleName.cs
 - Book.cs
 - Clip03Demo.cs
 - Name.cs
 - Common
 - Program.cs

FileEditViewProjectBuildDebugTestAnalyzeToolsExtensionsWindowHelpFull Screen

ManyNames.csClip03Demo.cs*Book.cs

DemoClip03.Clip03DemoImplementation()

```
Book oosc = new Book(  
    "Object-Oriented Software Construction",  
    new SingleName("Bertrand Meyer"));  
  
Book patterns = new Book(  
    "Design Patterns",  
    new ManyNames(new Name[] {  
        new ManyNames(new [] {  
            new SingleName("Erich Gamma"), new SingleName("Richard Helm") }},  
        new ManyNames(new [] {  
            new SingleName("Ralph Johnson"), new SingleName("John Vlissides") })  
        }));
```

ManyNames

Printable: Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides

ManyNames

Printable: Erich Gamma, Richard Helm

SingleName

Erich Gamma

SingleName

Ralph Johnson

ManyNames

Printable: Ralph Johnson, John Vlissides

SingleName

Richard Helm

SingleName

John Vlissides

Solution Explorer

Search Solution Explorer ()

Solution 'Demo' (1 of 1 projects)
Demo
Dependencies
Clip01
Clip02
Clip03
Names
Anonymous.
ManyNames.
SingleName.c
Book.cs
Clip03Demo.cs
Name.cs
Common
Program.cs

140 %No issues found

Ln: 26Ch: 97SPC

OutputError List

Ready

Add to Source Control

Summary

The Composite design pattern

- Simplifies code which consumes multitudes of objects
- Pull out complex code which deals with many objects
- Move that code into a composite element

Summary

Composite implementation

- Expose an abstract base (abstract class or interface)
- One concrete implementation models a single object
- Another implementation models many objects
- Client only observes the abstract base type

Summary

Common issues regarding Composite pattern

- Composite supports recursion
- Problem domain might not be recursive
- Composite implementation can forbid recursive elements
- Constructor can request a concrete type

Summary

Combining Composite with other patterns

- Fits nicely with Factory Method and Abstract Factory
- They detach client from Composite implementation
- Client only depends on factory and composite base

Summary

Advanced Composite implementations

- It can expose a single object as many objects
- Client may require uniform access to many objects
- Turn a single object into a sequence with one element
- Composite will always simplify the client code