

DESIGN PATTERNS IN C# MADE SIMPLE

MODULE 1 Understanding the Role of Design Patterns



ZORAN HORVAT
CEO AT CODING HELMET

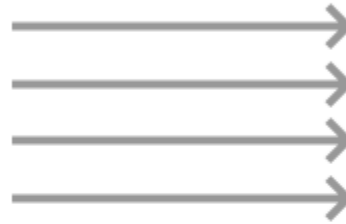
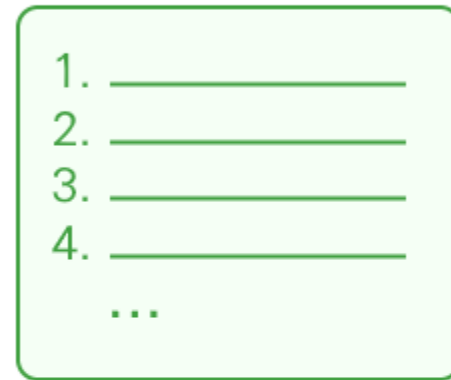
<http://codinghelmet.com>

zh@codinghelmet.com

 zoranh75

Programmer at Work

Requirements

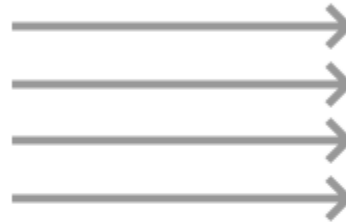
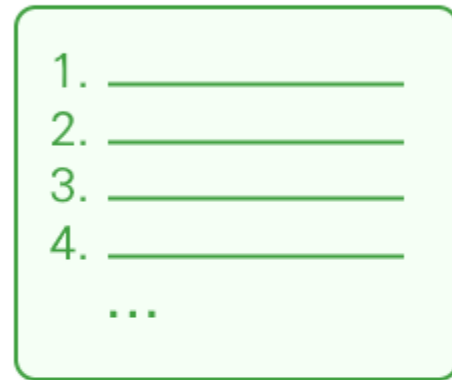


Implementation



Programmer at Work

Requirements

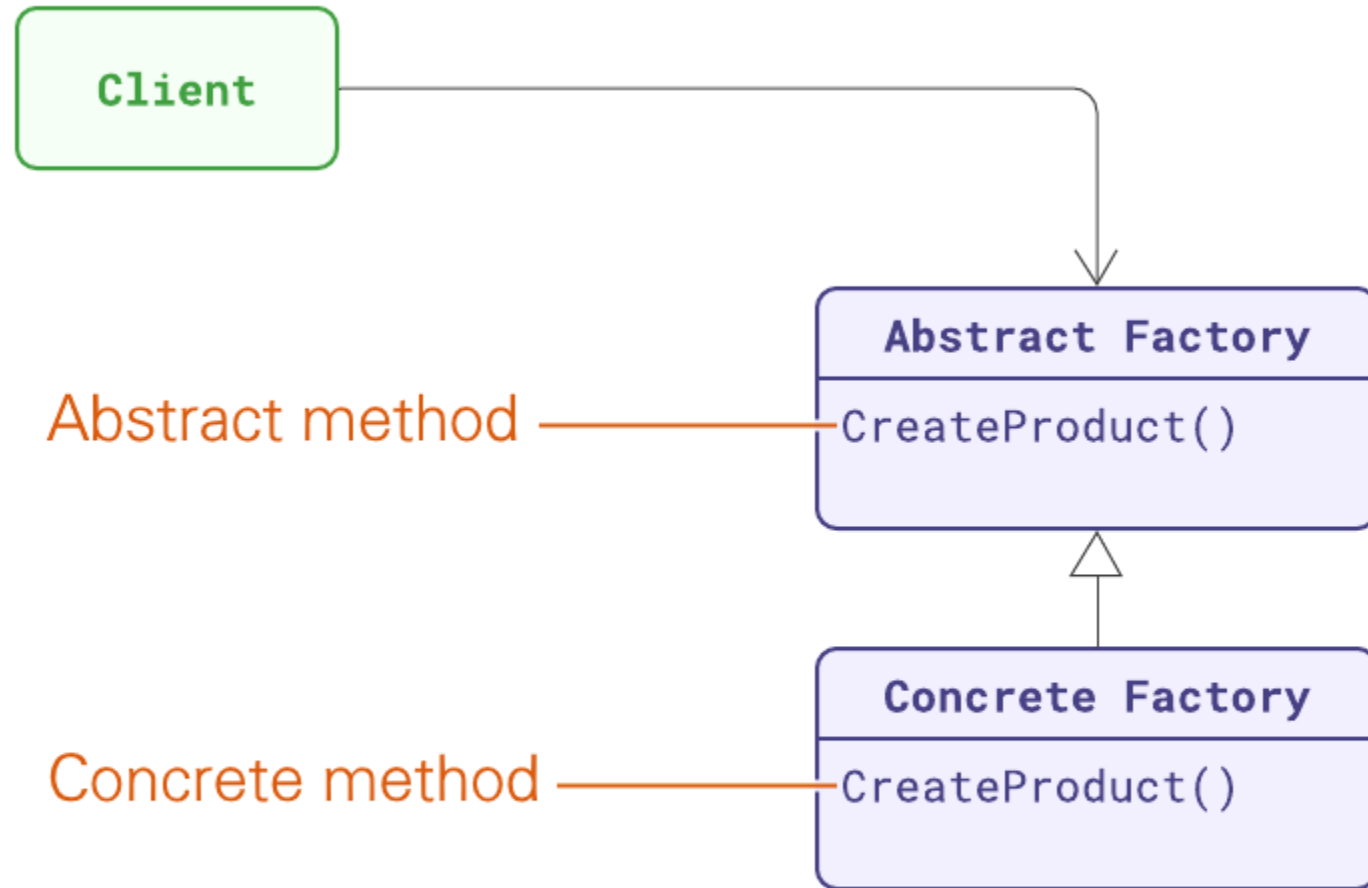


Implementation

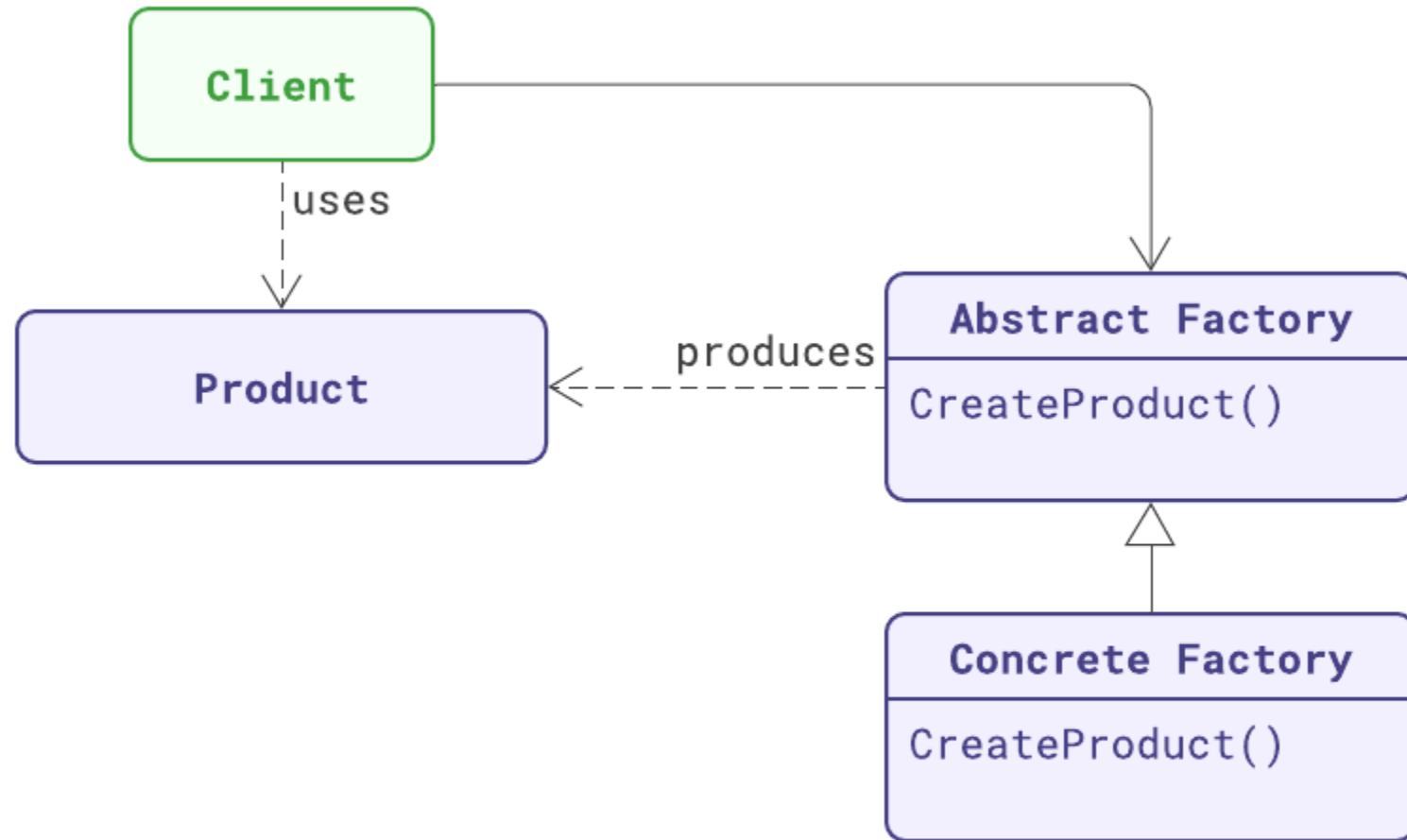


Any patterns here, yet?
What patterns!?

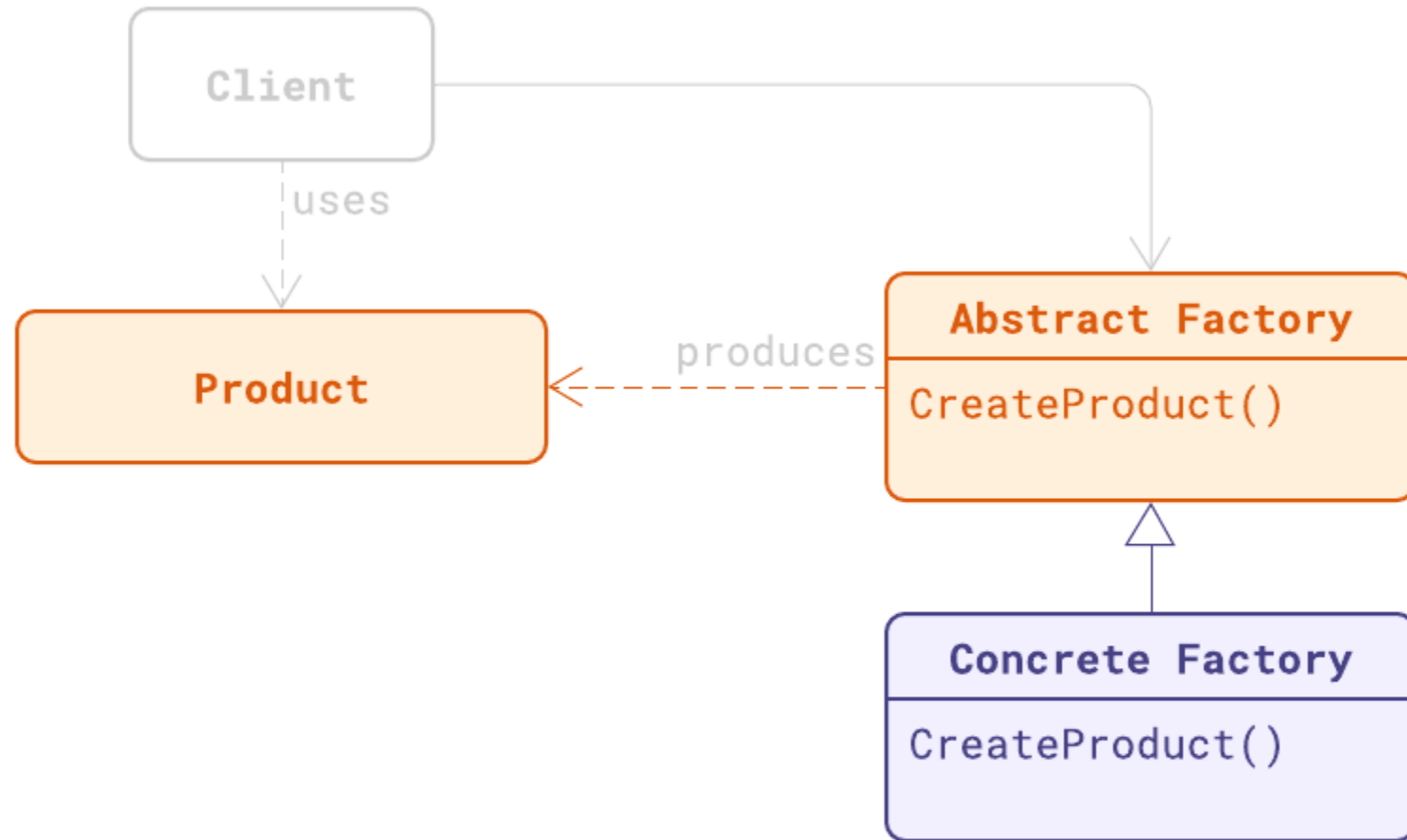
Example: Abstract Factory



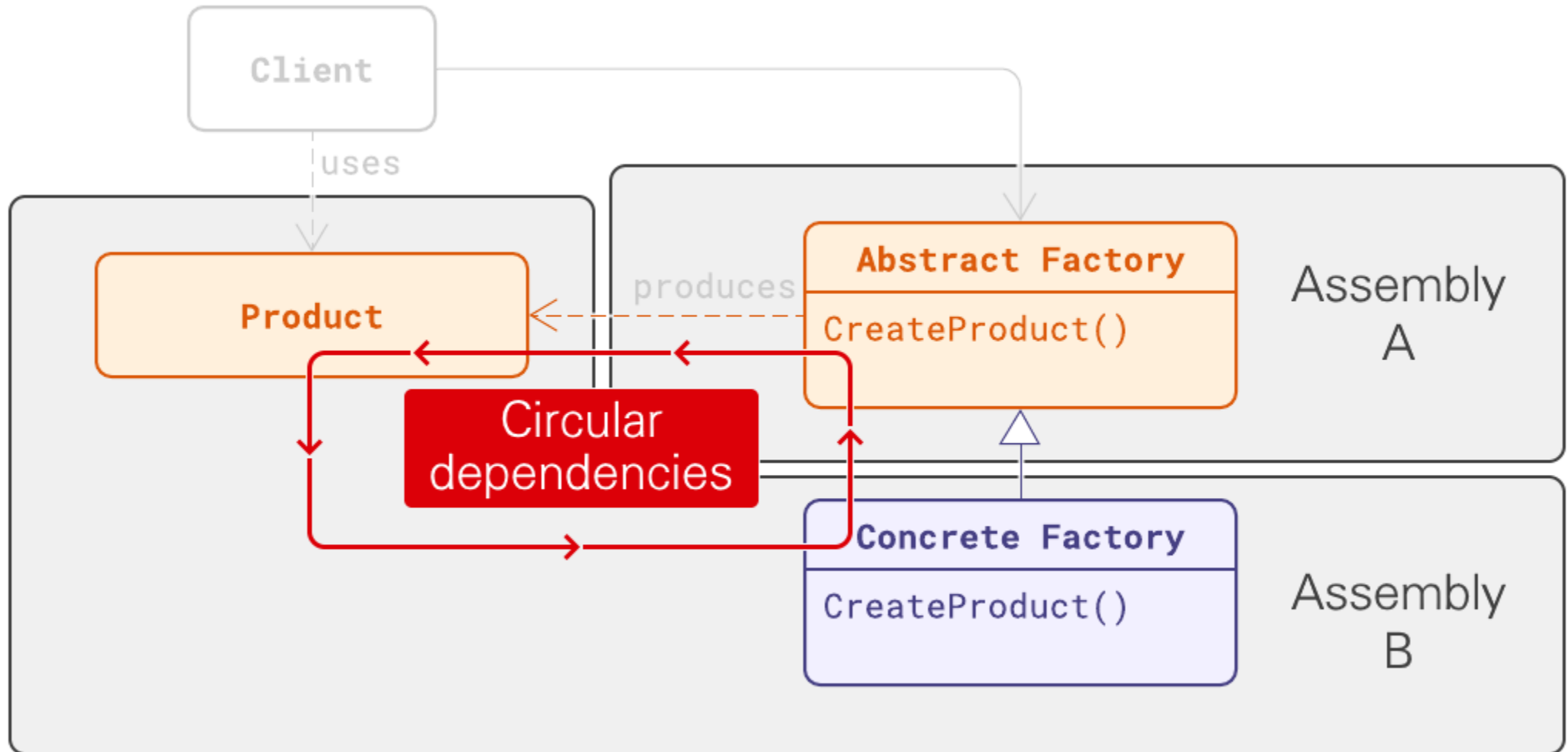
Example: Abstract Factory



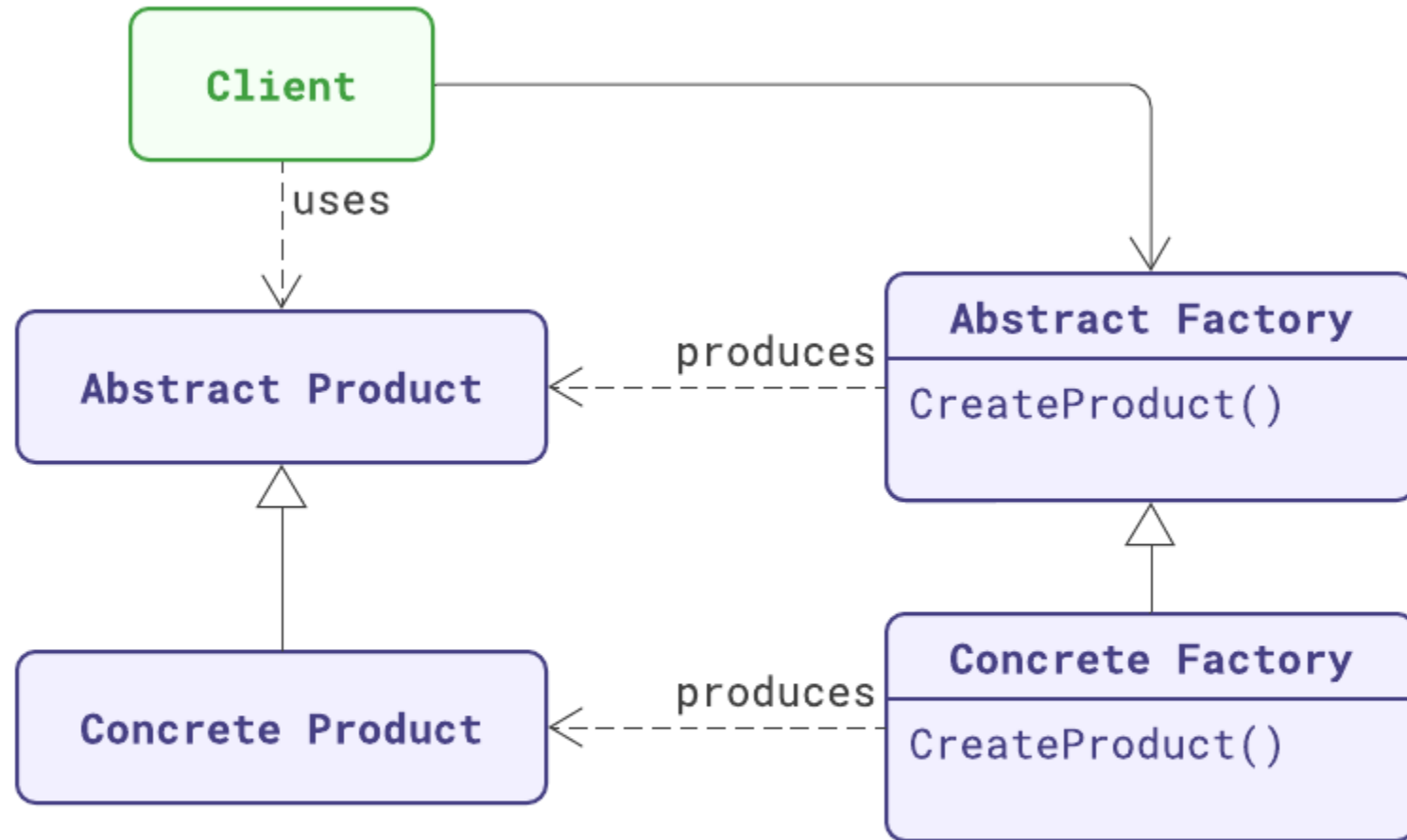
Example: Abstract Factory



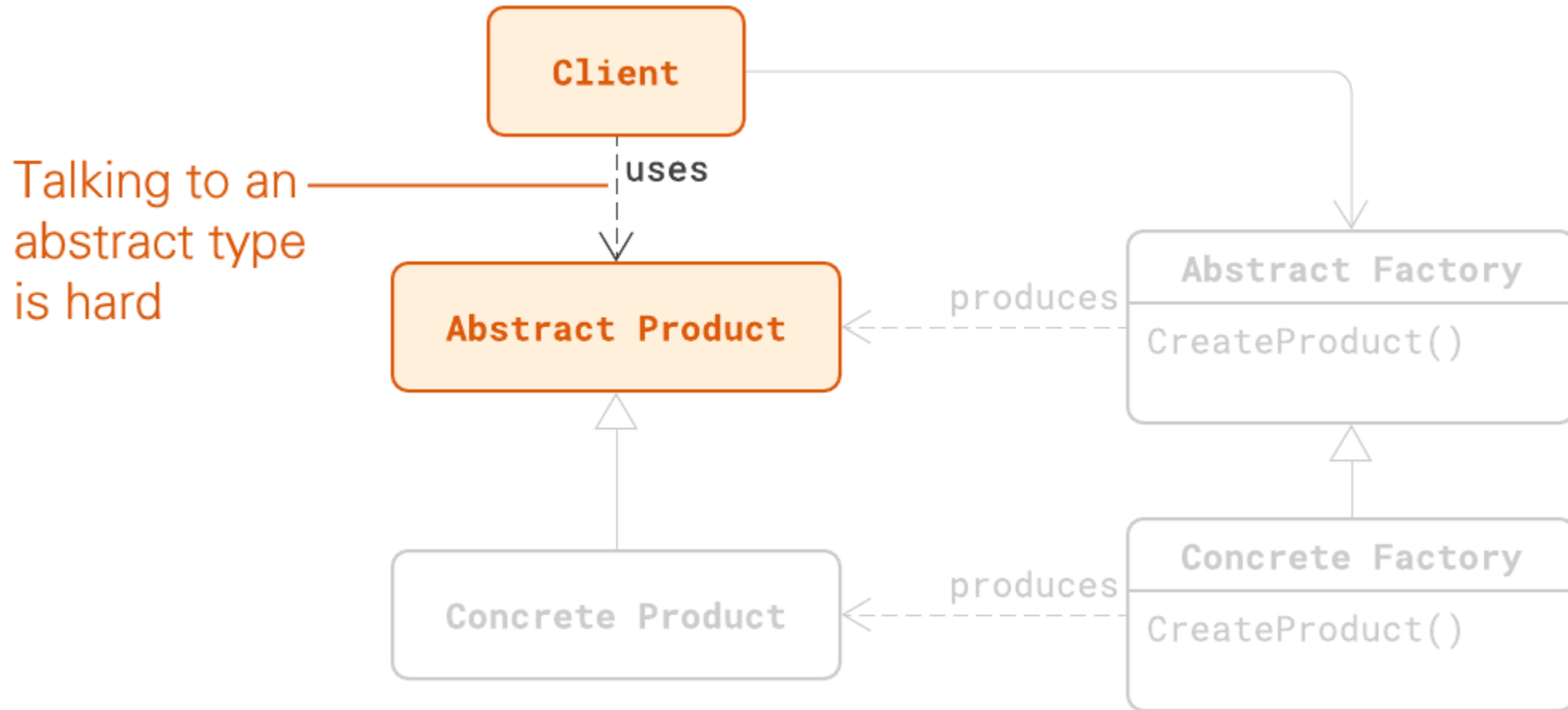
Example: Abstract Factory



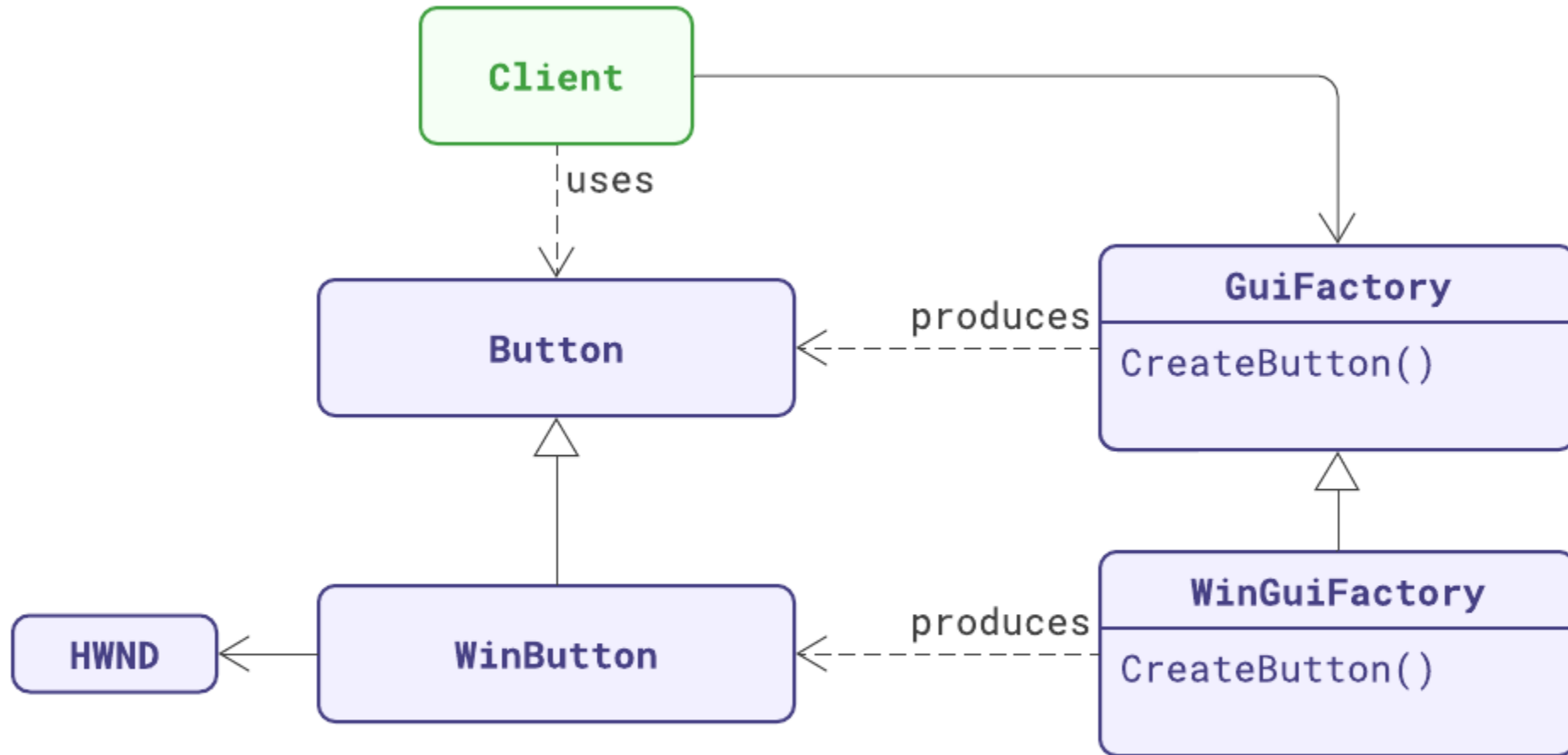
Example: Abstract Factory



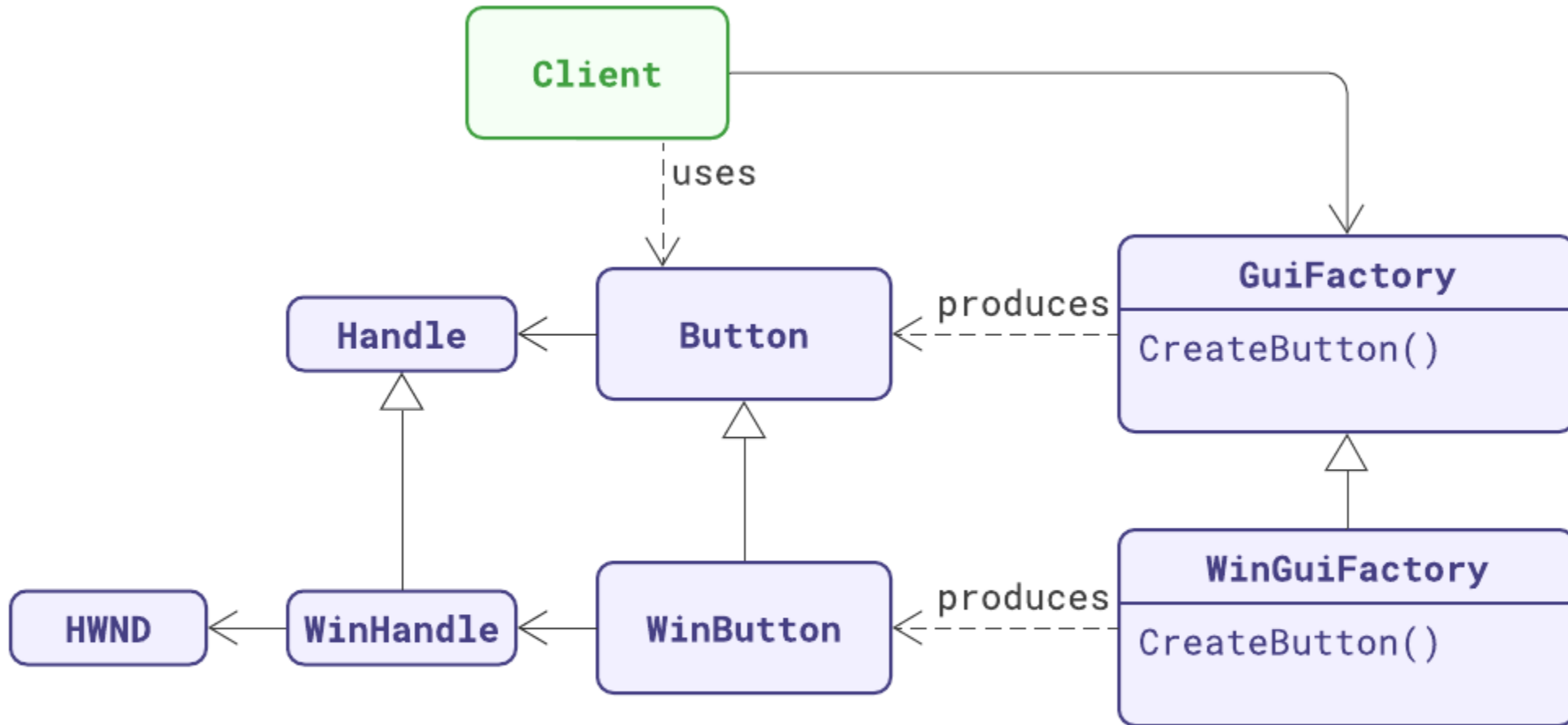
Example: Abstract Factory



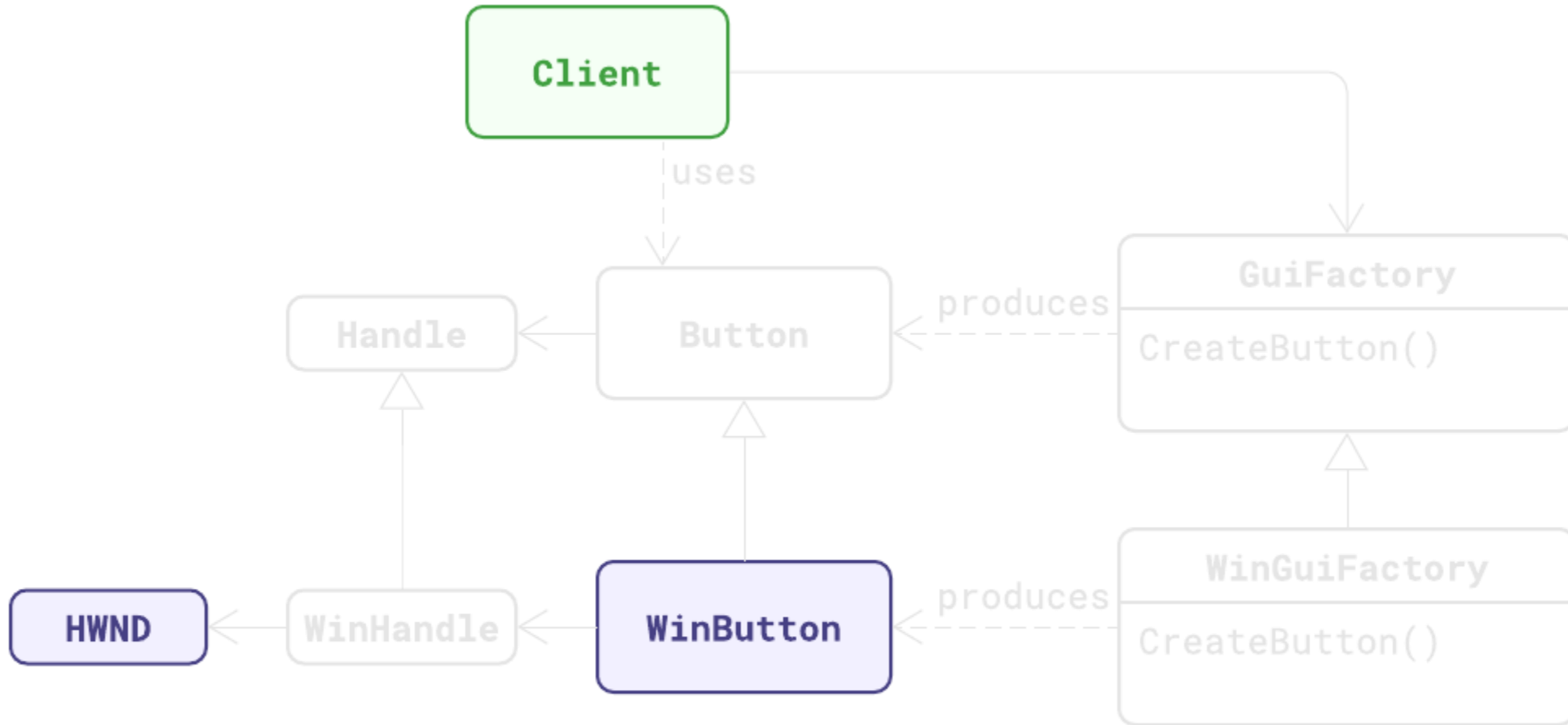
Example: GUI Factory



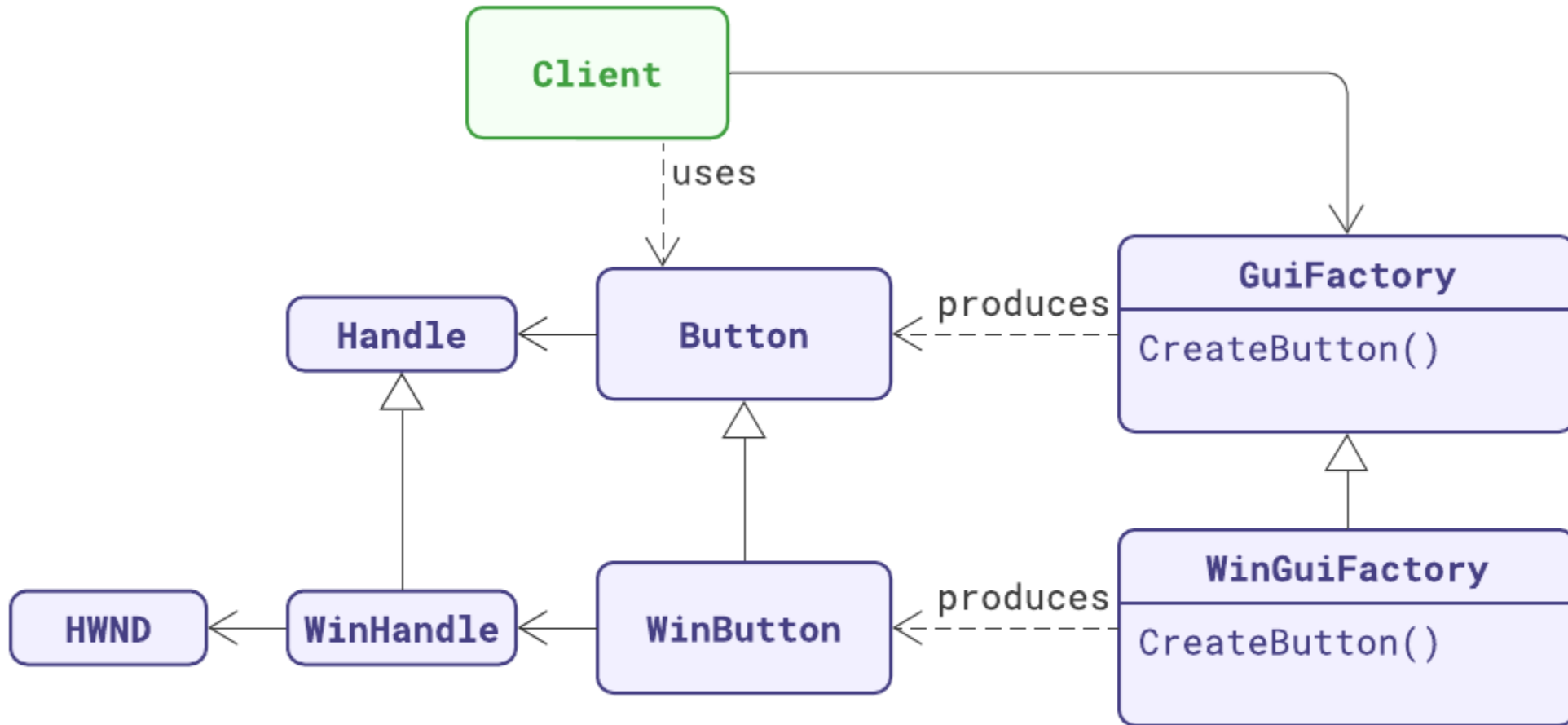
Example: GUI Factory



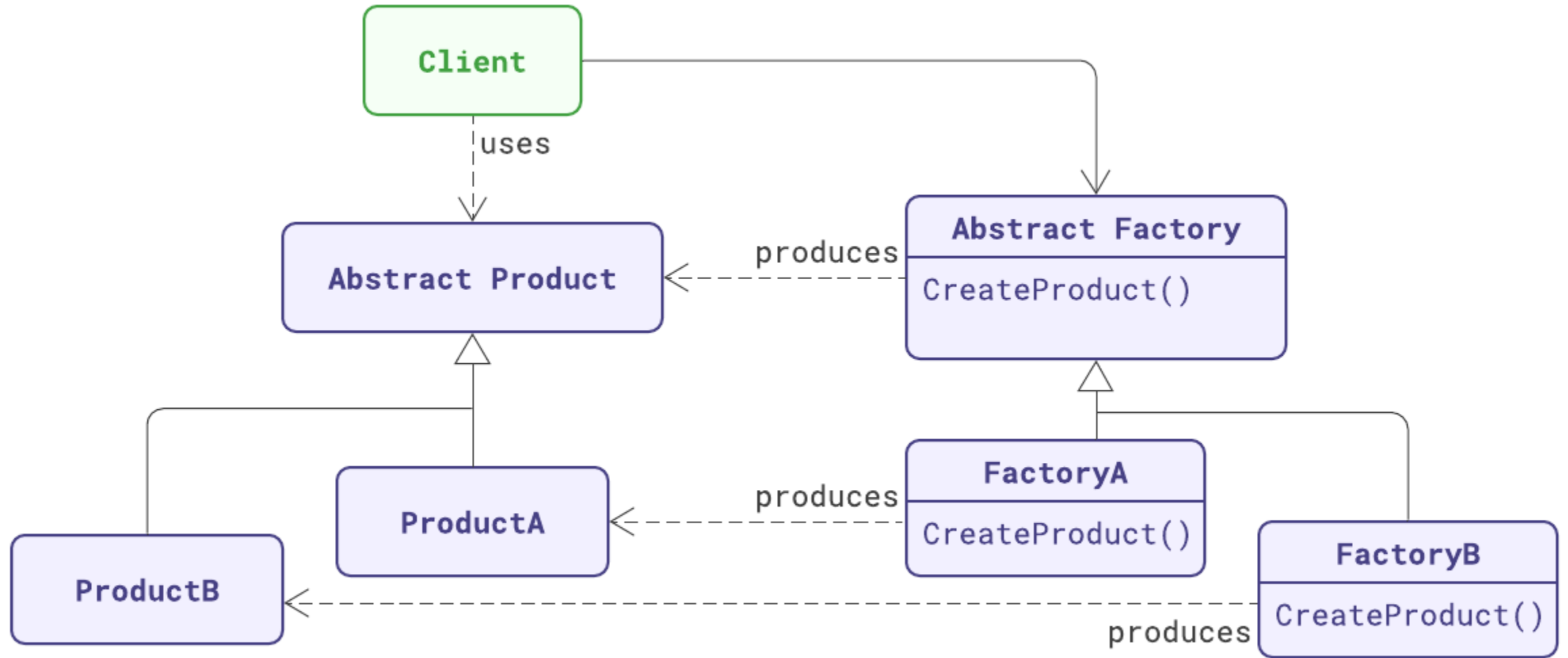
Example: GUI Factory



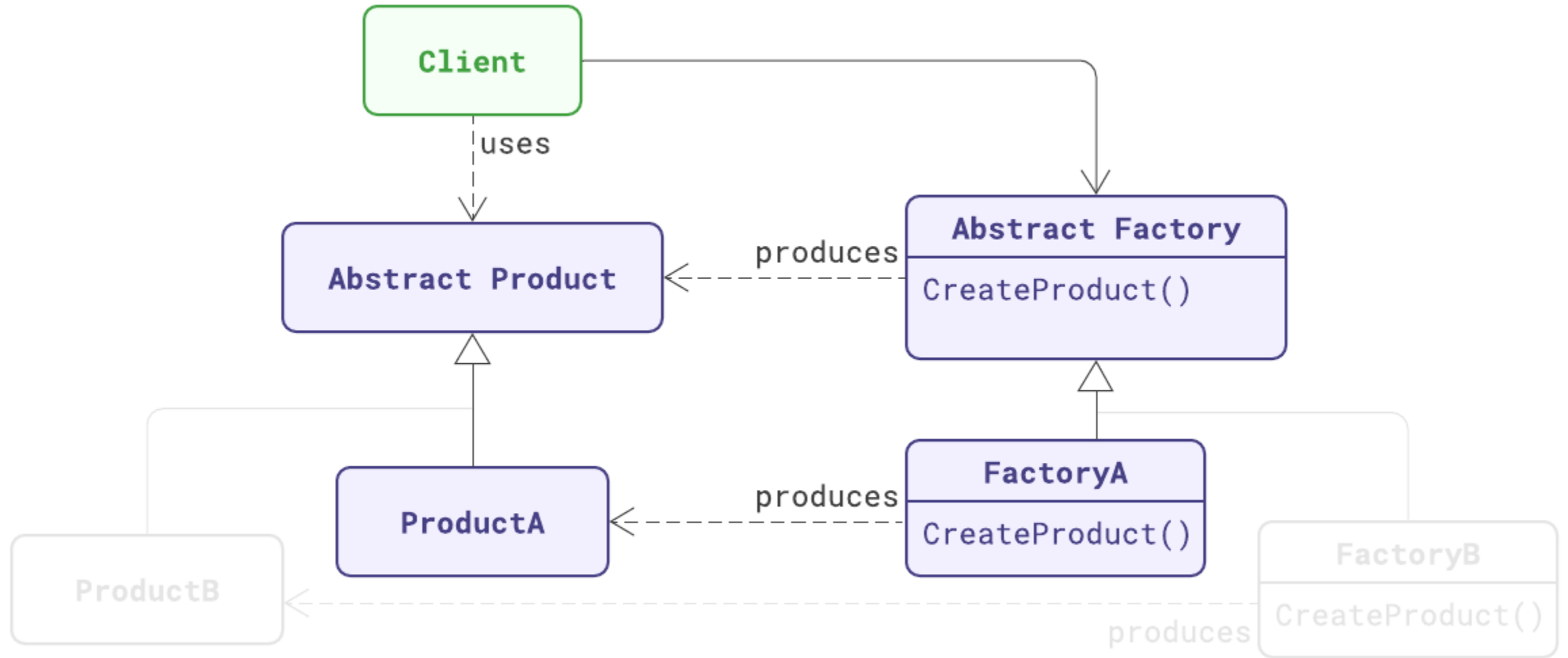
Example: GUI Factory



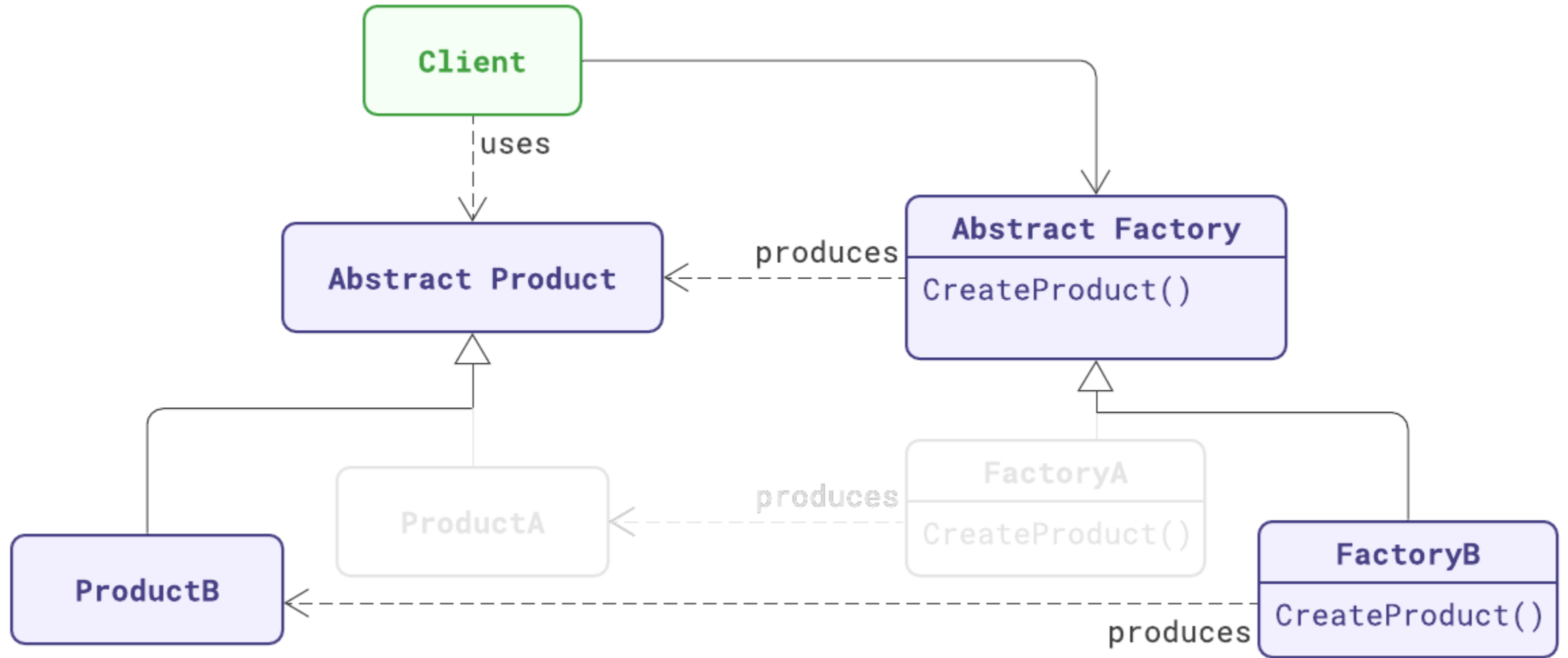
Example: Abstract Factory



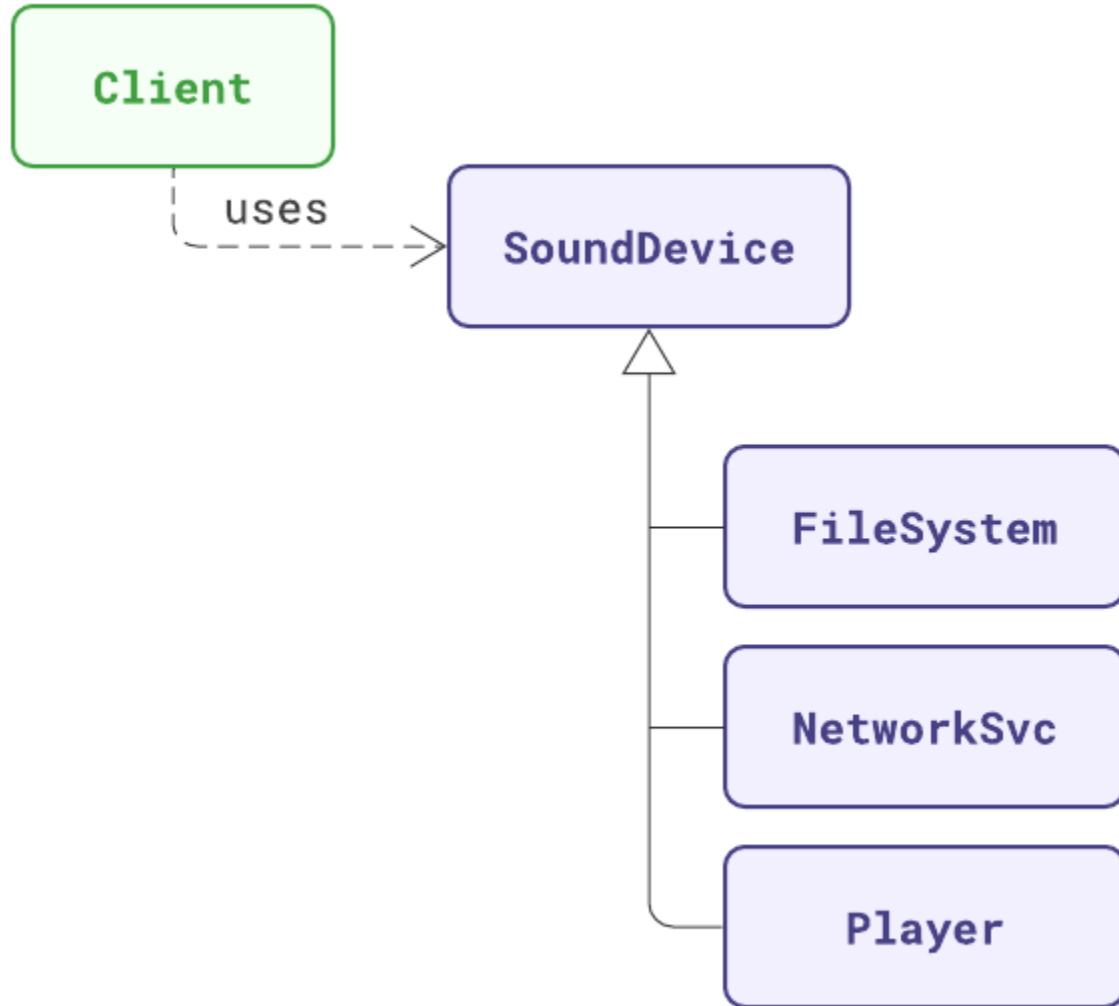
Example: Abstract Factory



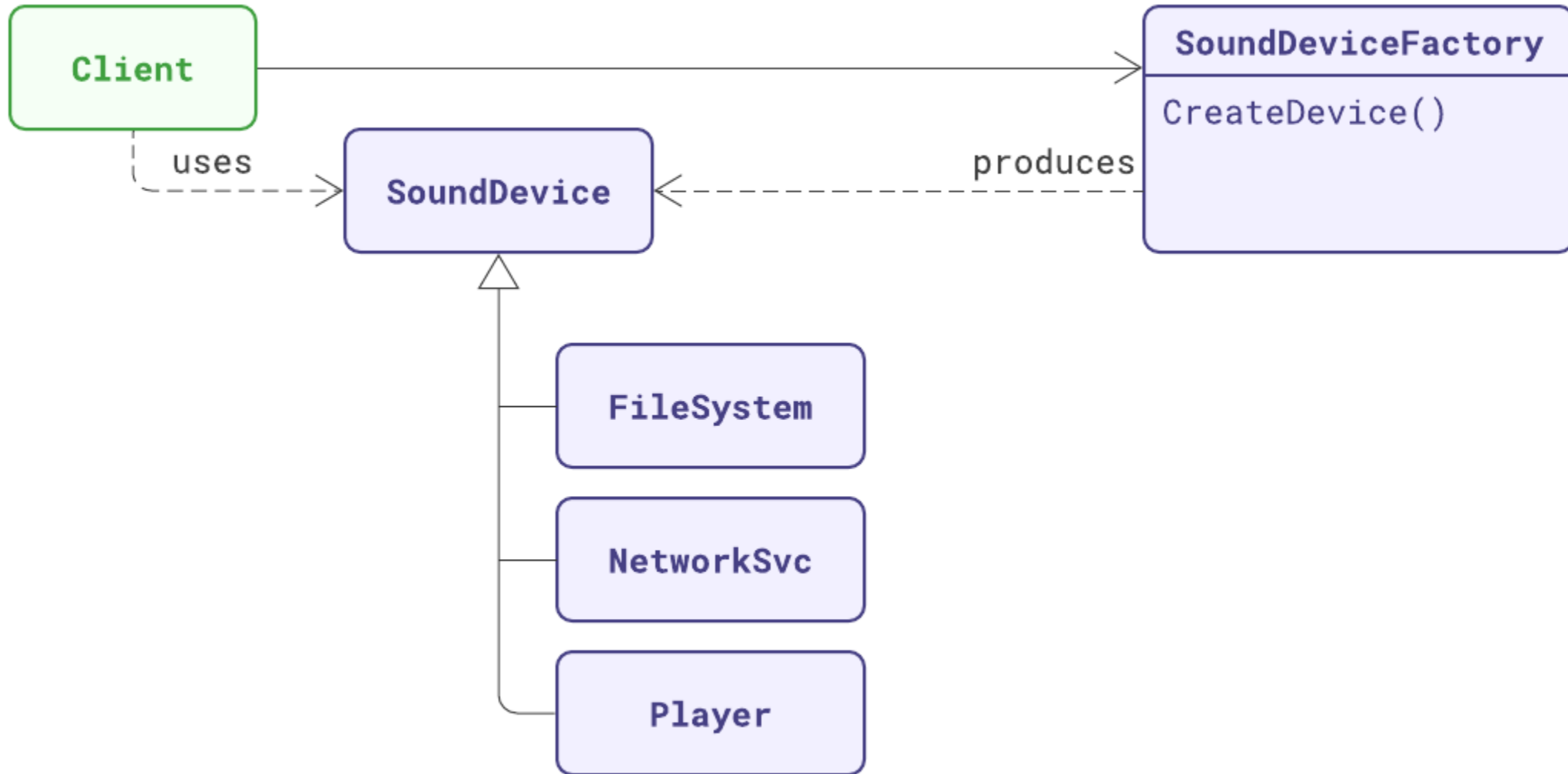
Example: Abstract Factory



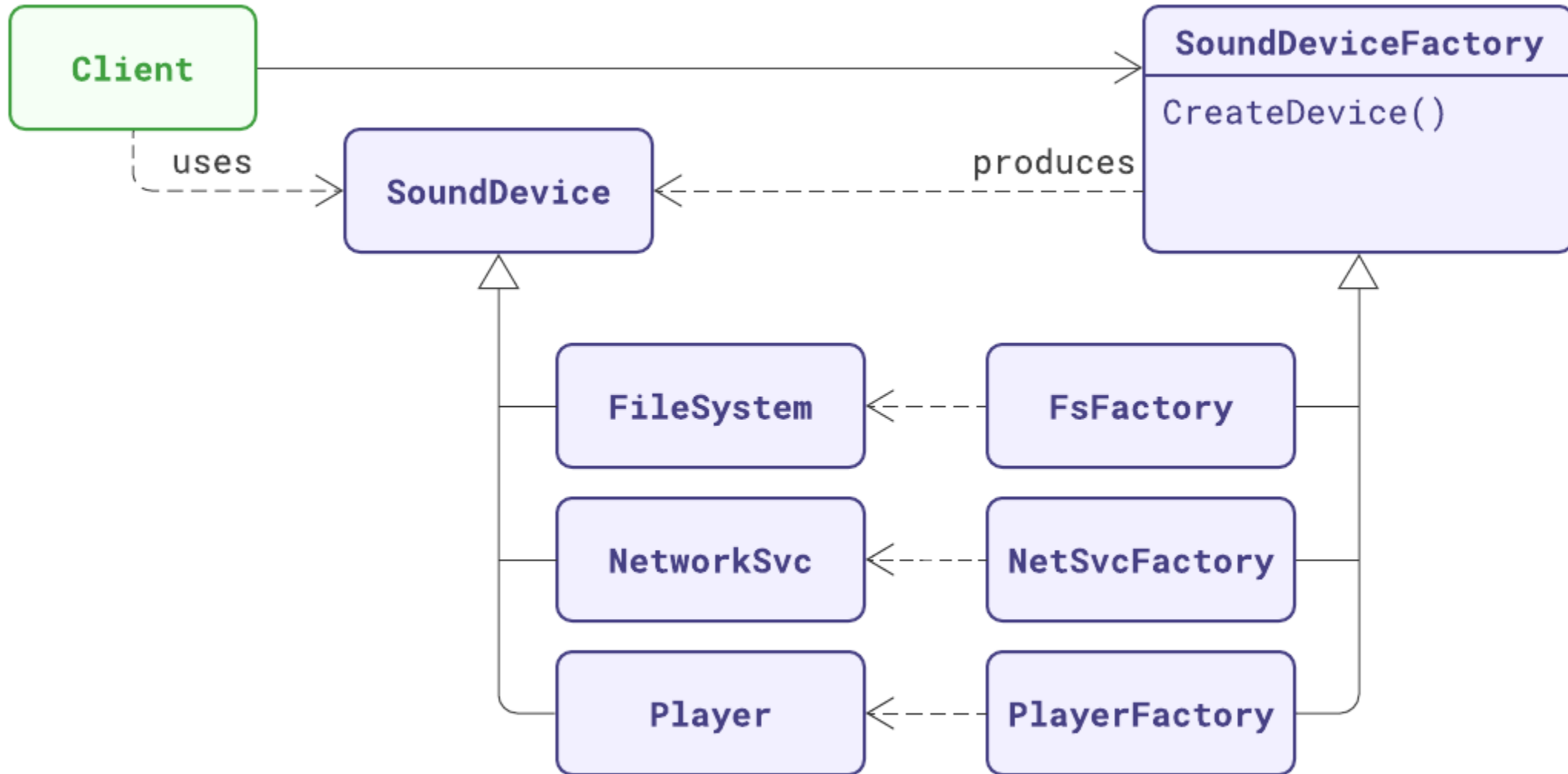
Example: Sound Processing



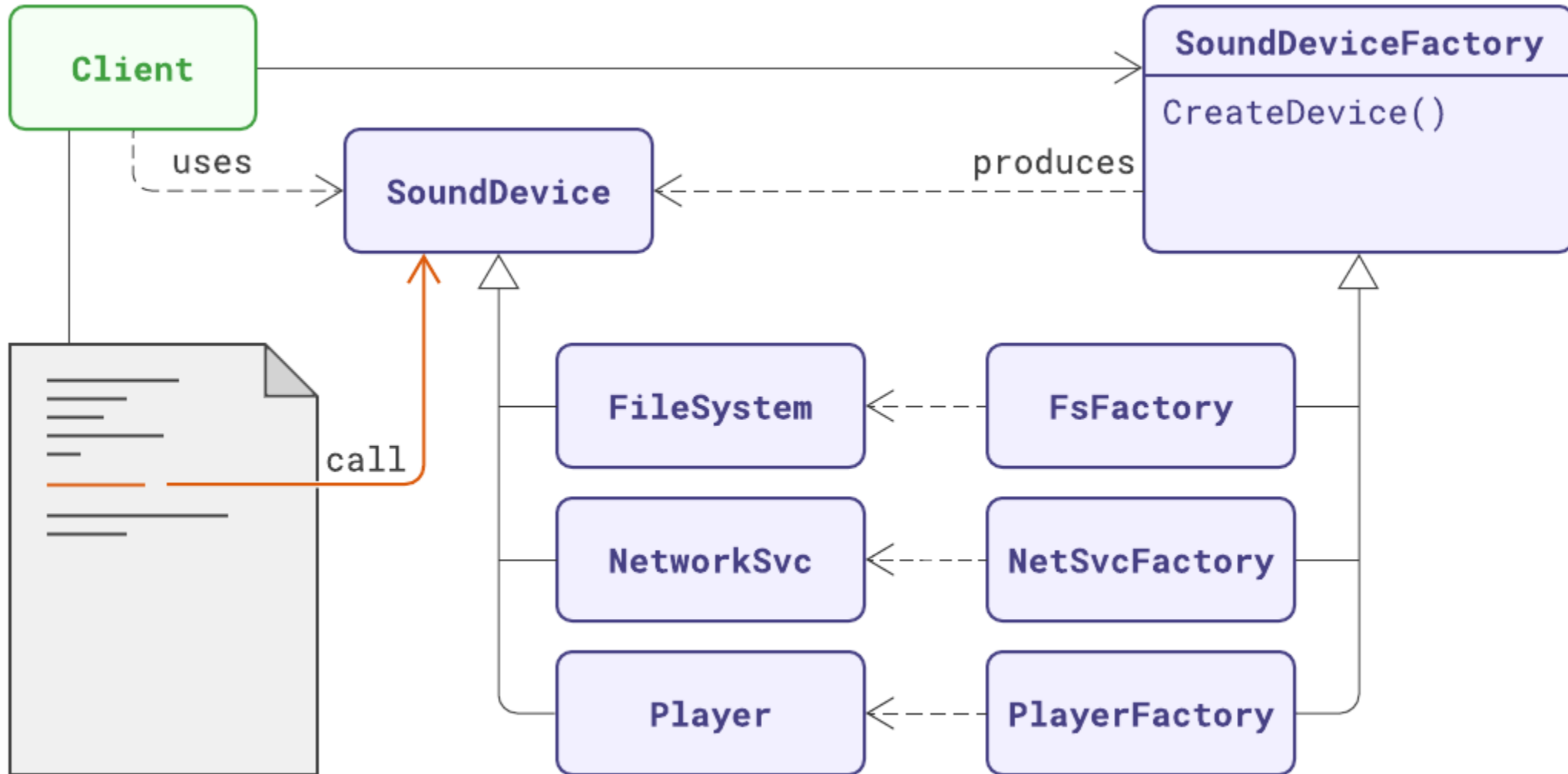
Example: Sound Processing



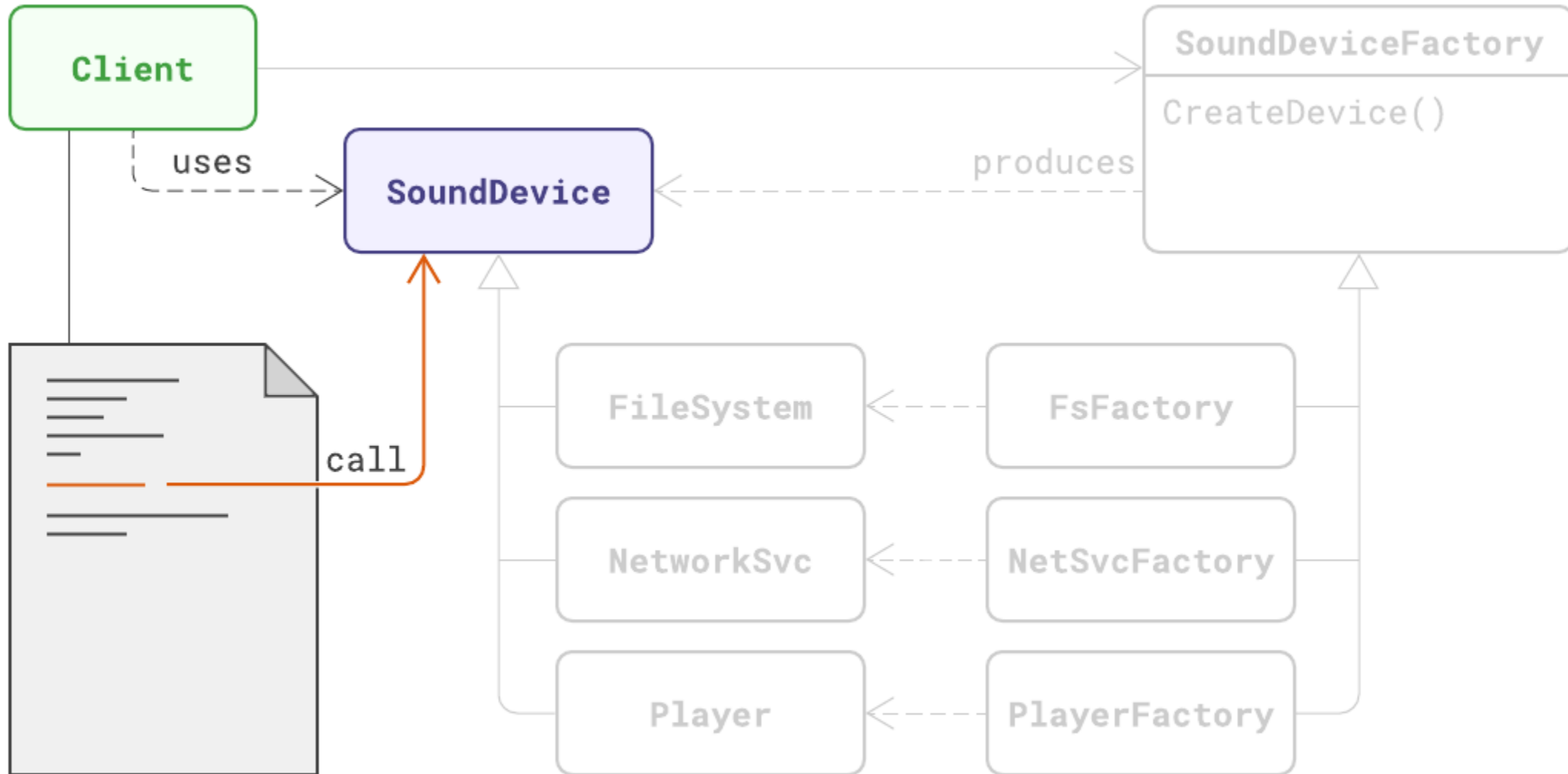
Example: Sound Processing



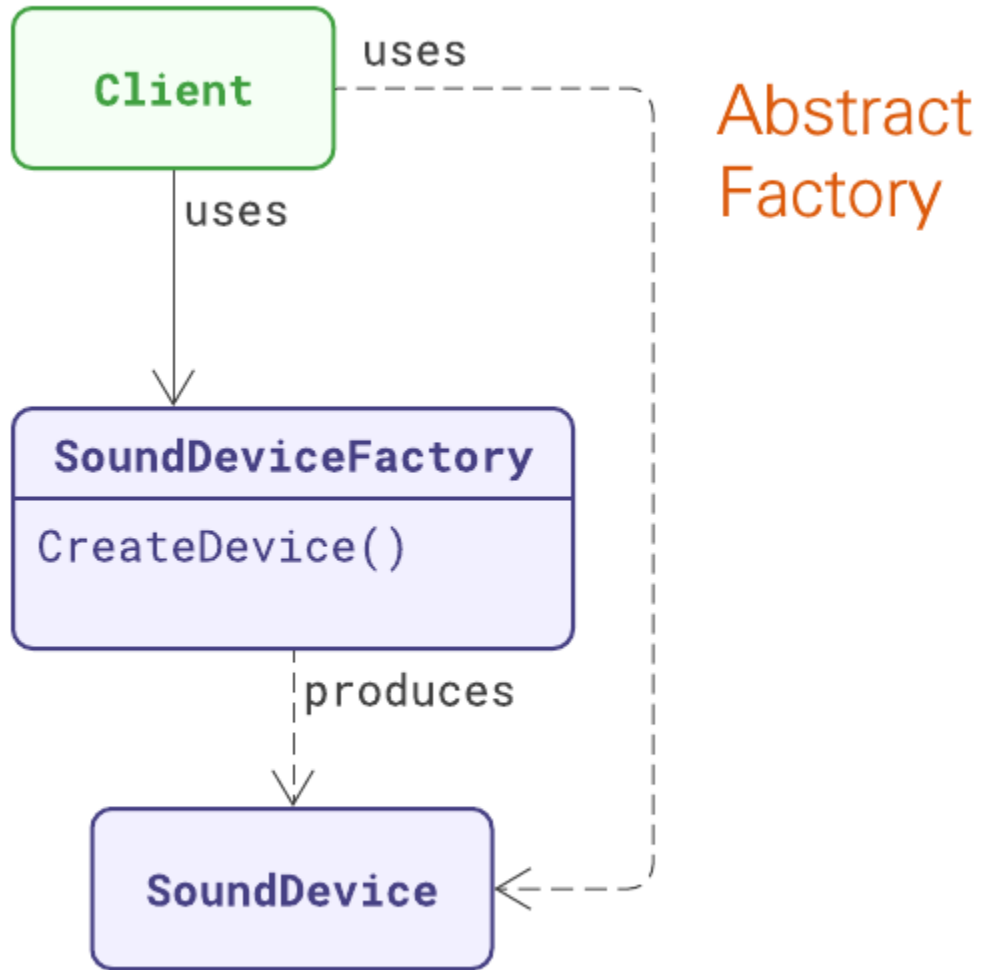
Example: Sound Processing



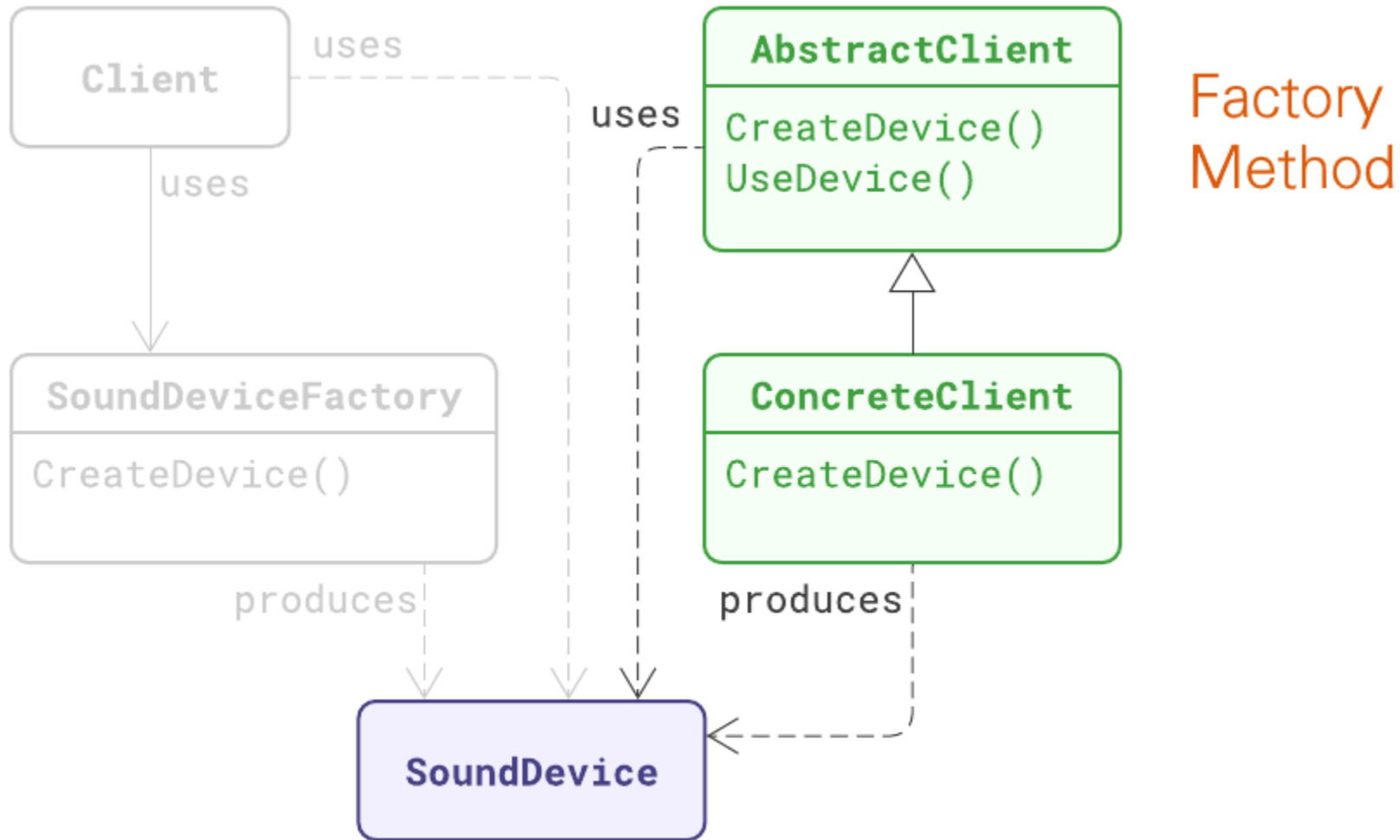
Example: Sound Processing



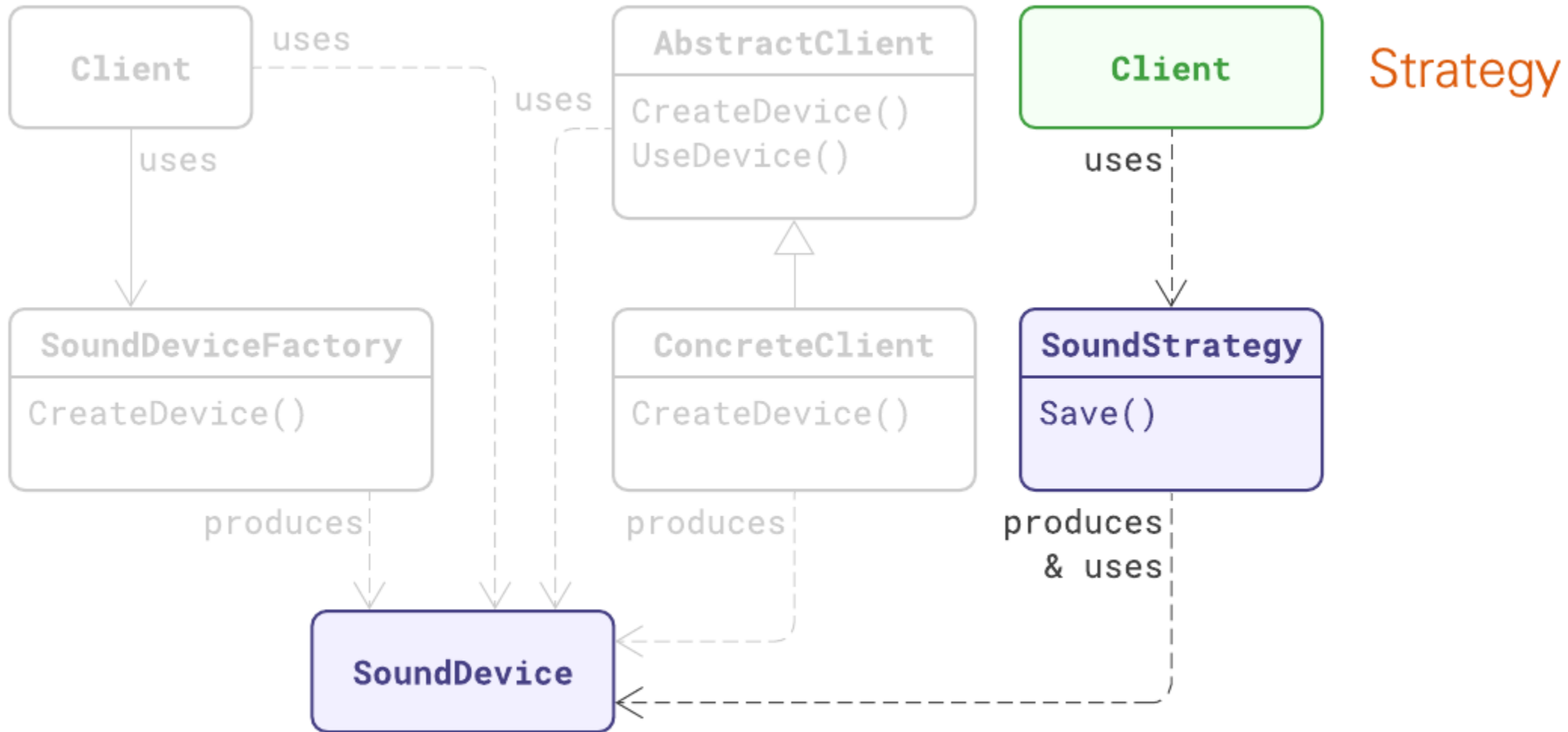
Example: Choosing a Pattern



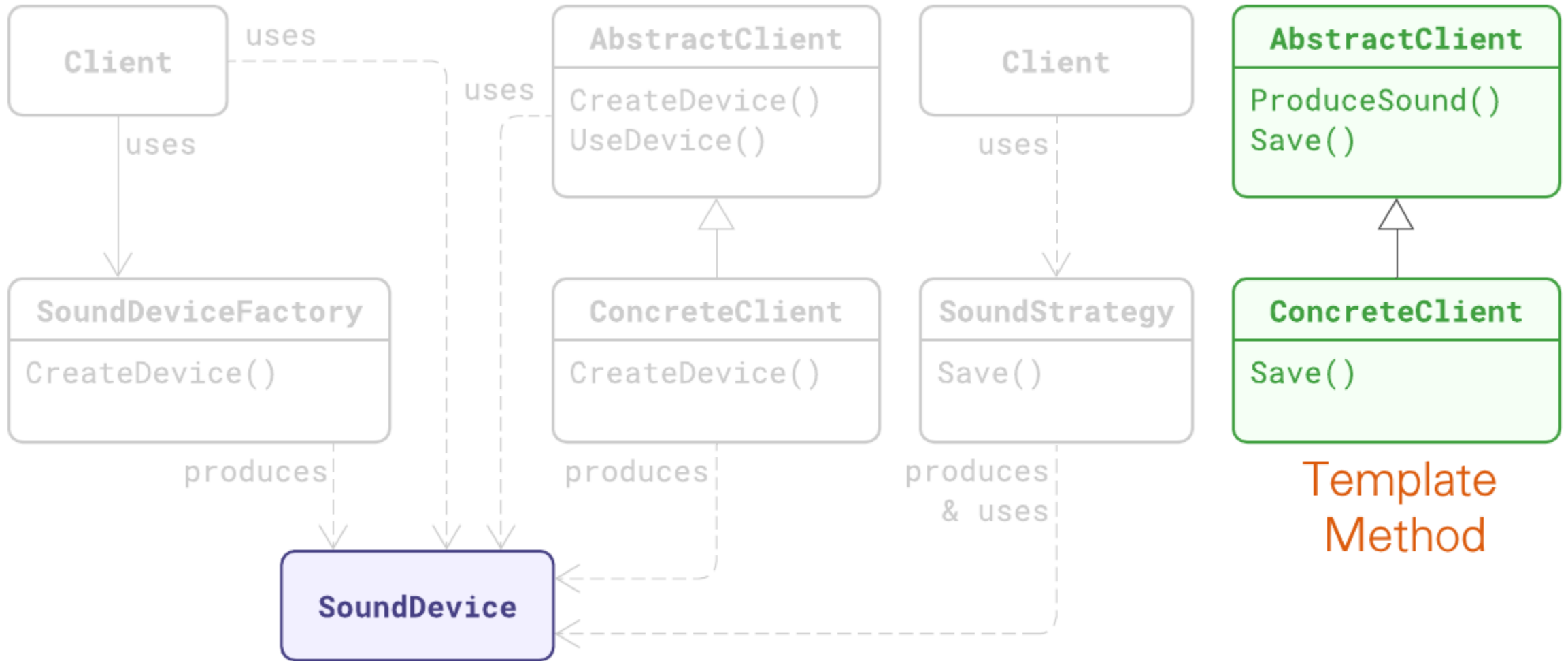
Example: Choosing a Pattern



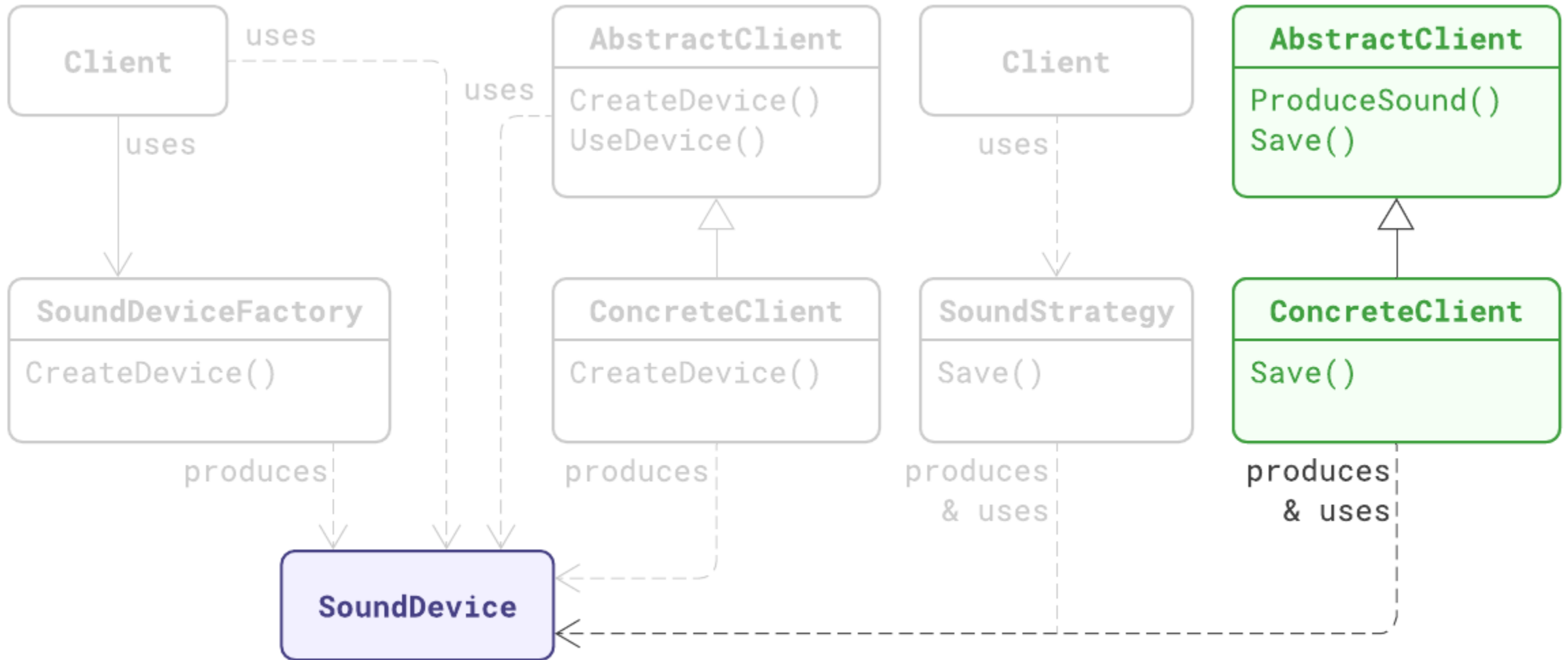
Example: Choosing a Pattern



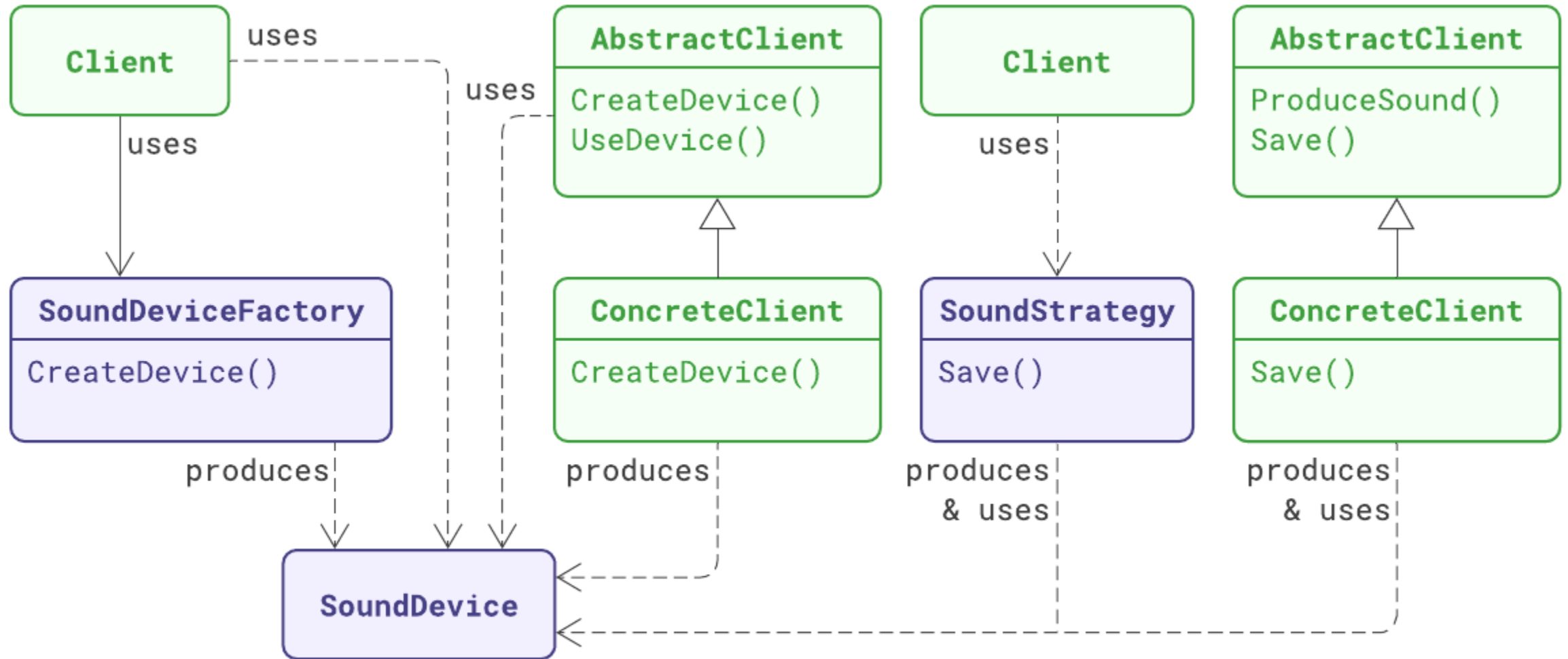
Example: Choosing a Pattern



Example: Choosing a Pattern



Example: Choosing a Pattern



Patterns in Agile Development

User story #1

As a composer,
I want to compose sound.

User story #2

As a composer,
I want to save sound to disk.

User story #3

As a composer,
I want to hear sound.

User story #4

As a composer,
I want to send sound to online library.

Patterns in Agile Development

User story #1

As a composer,
I want to compose sound.

```
void ProduceSound()  
{  
    ...  
    SaveDebugging(uncompressed);  
}  
  
void SaveDebugging(byte[] raw)  
{  
    fileName = "debug.wav";  
    ...  
}
```

Patterns in Agile Development

User story #1

As a composer,
I want to compose sound.

User story #2

As a composer,
I want to save sound to disk.

```
void ProduceSound()  
{  
    ...  
    Save(uncompressed);  
}  
  
void Save(byte[] raw, string fileName)  
{  
    compressed = Compress(raw);  
    ...  
}  
  
byte[] Compress(byte[] raw) { ... }
```

Patterns in Agile Development

User story #1

As a composer,
I want to compose sound.

User story #2

As a composer,
I want to save sound to disk.

```
void ProduceSound()
```

```
{
```

```
...
```

```
    Save(uncompressed);
```

```
}
```

```
void Save(byte[] raw, string fileName)
```

```
{
```

```
    compressed = Compress(raw);
```

```
...
```

```
}
```

```
byte[] Compress(byte[] raw) { ... }
```

Algorithm
parameter

Patterns in Agile Development

User story #1

As a composer,
I want to compose sound.

User story #2

As a composer,
I want to save sound to disk.

User story #3

As a composer,
I want to hear sound.

```
void ProduceSound()  
{  
    ...  
    Save(uncompressed);  
}  
  
void Save(byte[] raw, string fileName)  
{  
    compressed = Compress(raw);  
    ...  
}  
  
void Play(byte[] raw, int deviceId) { ... }  
  
byte[] Compress(byte[] raw) { ... }
```


Patterns in Agile Development

User story #1

As a composer,
I want to compose sound.

User story #2

As a composer,
I want to save sound to disk.

User story #3

As a composer,
I want to hear sound.

```
void ProduceSound()  
{  
    ...  
    Save(uncompressed);  
}
```

```
void Save(byte[] raw, string fileName)  
{  
    compressed = Compress(raw);  
    ...  
}
```

```
void Play(byte[] raw, int deviceId) { ... }
```

```
byte[] Compress(byte[] raw) { ... }
```

Incompatible
algorithm
parameters



Patterns in Agile Development

User story #1

As a composer,
I want to compose sound.

User story #2

As a composer,
I want to save sound to disk.

User story #3

As a composer,
I want to hear sound.

```
void Save(byte[] raw, string fileName)
```

```
void Play(byte[] raw, int deviceId)
```

Patterns in Agile Development

User story #1

As a composer,
I want to compose sound.

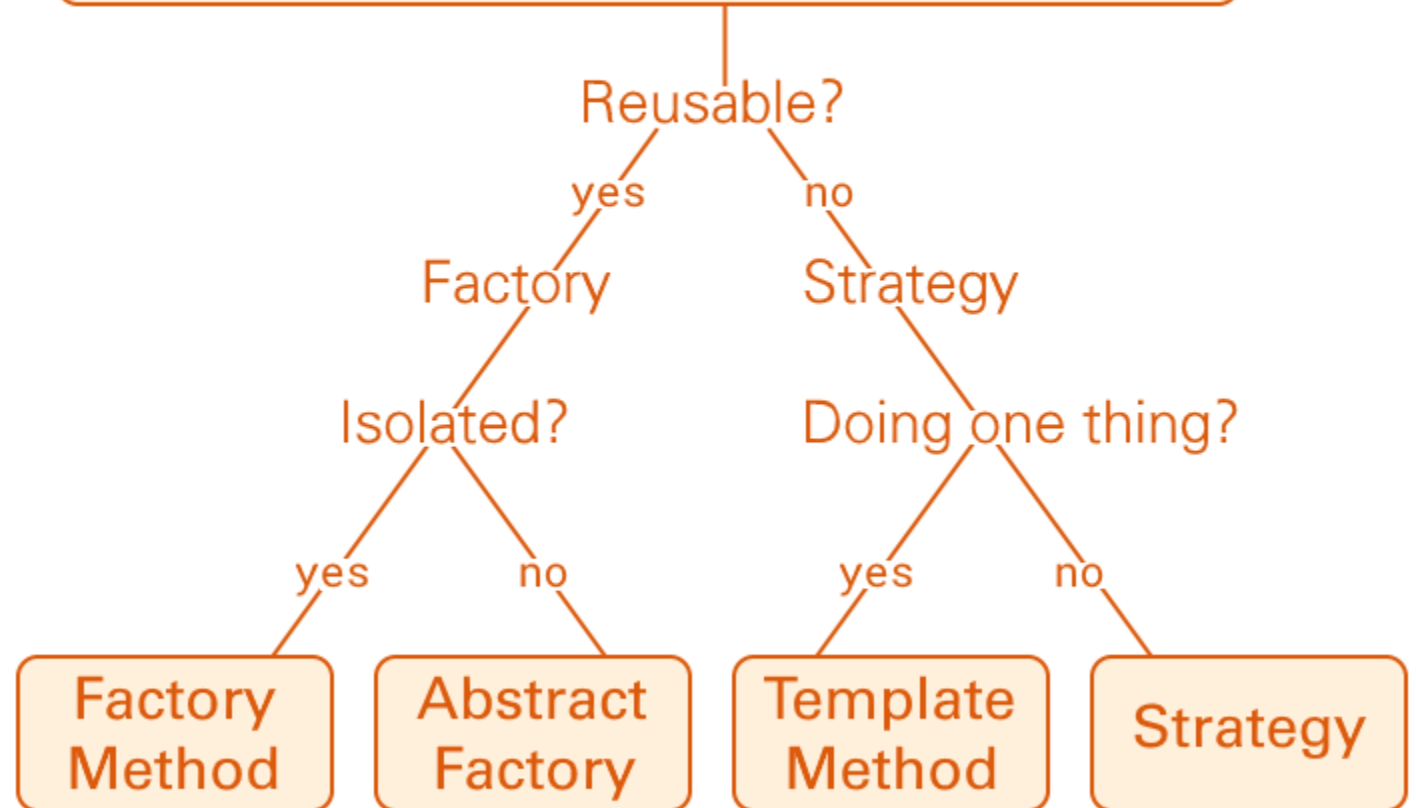
User story #2

As a composer,
I want to save sound to disk.

User story #3

As a composer,
I want to hear sound.

```
void Save(byte[] raw, string fileName)
void Play(byte[] raw, int deviceId)
```



Patterns in Agile Development

User story #1

As a composer,
I want to compose sound.

User story #2

As a composer,
I want to save sound to disk.

User story #3

As a composer,
I want to hear sound.

```
void Save(byte[] raw, string fileName)
void Play(byte[] raw, int deviceId)
```

Right time to apply a pattern

When there is enough information

Effects of applying a pattern

Simplified design

Lots of code removed to helping types

Patterns in Agile Development

User story #1

As a composer,
I want to compose sound.

User story #2

As a composer,
I want to save sound to disk.

User story #3

As a composer,
I want to hear sound.

User story #4

As a composer,
I want to send sound to online library.

```
void Save(byte[] raw, string fileName)
void Play(byte[] raw, int deviceId)
void Send(byte[] raw, NetSvc service)
```

+1 component



Patterns in Agile Development

User story #1

As a composer,
I want to compose sound.

User story #2

As a composer,
I want to save sound to disk.

User story #3

As a composer,
I want to hear sound.

User story #4

As a composer,
I want to send sound to online library.

```
void Save(byte[] raw, string fileName)
void Play(byte[] raw, int deviceId)
void Send(byte[] raw, NetSvc service)
```

In this Module

Structure of this course

- Where to find code and slides
- How to play course videos

Relation to other domains

- Code refactoring
- Evolutionary design

Refactoring to Design Patterns

<http://codinghelmet.com/go/refactoring-to-patterns>

Mastering Iterative Object-oriented Design

<http://codinghelmet.com/go/mastering-iterative-ood>

Structure of This Course

Decorator

Adapter

Strategy

Abstract Factory

Factory Method

Builder

Composite

Structure of This Course

Decorator
Adapter
Strategy
Abstract Factory
Factory Method
Builder
Composite



Please, watch the modules linearly

Structure of This Course

Decorator
Adapter
Strategy
Abstract Factory
Factory Method
Builder
Composite



Please, watch the modules linearly

There is more in them than just patterns!

Structure of This Course

Decorator

Adapter

Strategy

Abstract Factory

Factory Method

Builder

Composite

Structure of This Course

Decorator

Adapter

Strategy

Abstract Factory

Factory Method

Builder

Composite

How to apply the Decorator and Adapter

Structure of This Course

Decorator

Adapter

Strategy

Abstract Factory

Factory Method

Builder

Composite

How to apply the Decorator and Adapter

When and ***where*** to inject a pattern

Structure of This Course

Decorator

Adapter

Strategy

Abstract Factory

Factory Method

Builder

Composite

How to apply the Decorator and Adapter

When and *where* to inject a pattern

Recognizing the *need* to change

Structure of This Course

Decorator

Adapter

Strategy

Abstract Factory

Factory Method

Builder

Composite

Every object must be constructed

Structure of This Course

Decorator

Adapter

Strategy

Abstract Factory

Factory Method

Builder

Composite

Every object must be constructed

Creational patterns are fundamental

Structure of This Course

Decorator
Adapter
Strategy
Abstract Factory
Factory Method
Builder
Composite

Section 2: Adding Behavior to an Object with the Decorator Pattern

34min

3. Motivation for the Decorator Pattern

▶ 7min

4. The First Attempt to Decorate a Class

▶ 9min

5. The Downfall of the Subclassing Decorator

▶ 7min

6. Using Object Composition Instead of Inheritance

▶ 9min

7. Summary

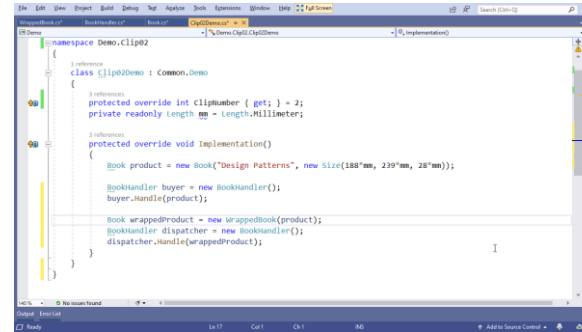
▶ 2min

 Resources ▾



Structure of This Course

Decorator
Adapter
Strategy
Abstract Factory
Factory Method
Builder
Composite



```
namespace Demo.Clip02
{
    [reference]
    class Clip02Demo : Common.Demo
    {
        [reference]
        protected override int clipNumber { get; } = 2;
        private readonly length mm = length.Millimeter;

        [reference]
        protected override void Implementation()
        {
            Book product = new Book("Design Patterns", new Size(188"mm", 239"mm", 28"mm));
            BookHandler buyer = new BookHandler();
            buyer.Handle(product);

            Book wrappedProduct = new WrappedBook(product);
            BookHandler dispatcher = new BookHandler();
            dispatcher.Handle(wrappedProduct);
        }
    }
}
```

Resources ▾

demos.zip
slides.pdf



Structure of This Course

Decorator
Adapter
Strategy
Abstract Factory
Factory Method
Builder
Composite

demos.zip

▼ M04

▼ Initial

Clip01

Clip02

Clip03

Clip04

Clip05

Clip06

Clip07

Solution #1

▼ Final

Clip01

Clip02

Clip03

Clip04

Clip05

Clip06

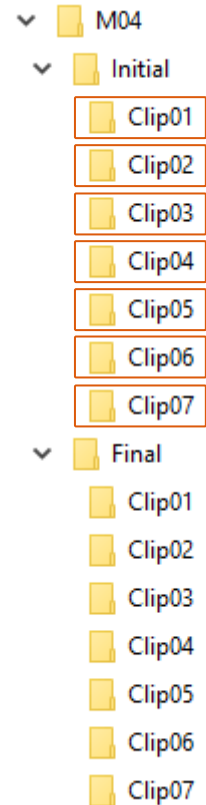
Clip07

Solution #2

Structure of This Course

Decorator
Adapter
Strategy
Abstract Factory
Factory Method
Builder
Composite

demos.zip

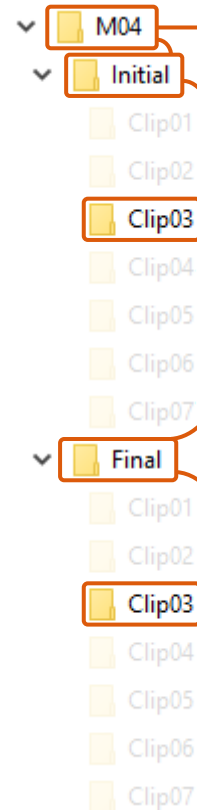


Separate projects
in the solution

Structure of This Course

Decorator
Adapter
Strategy
Abstract Factory
Factory Method
Builder
Composite

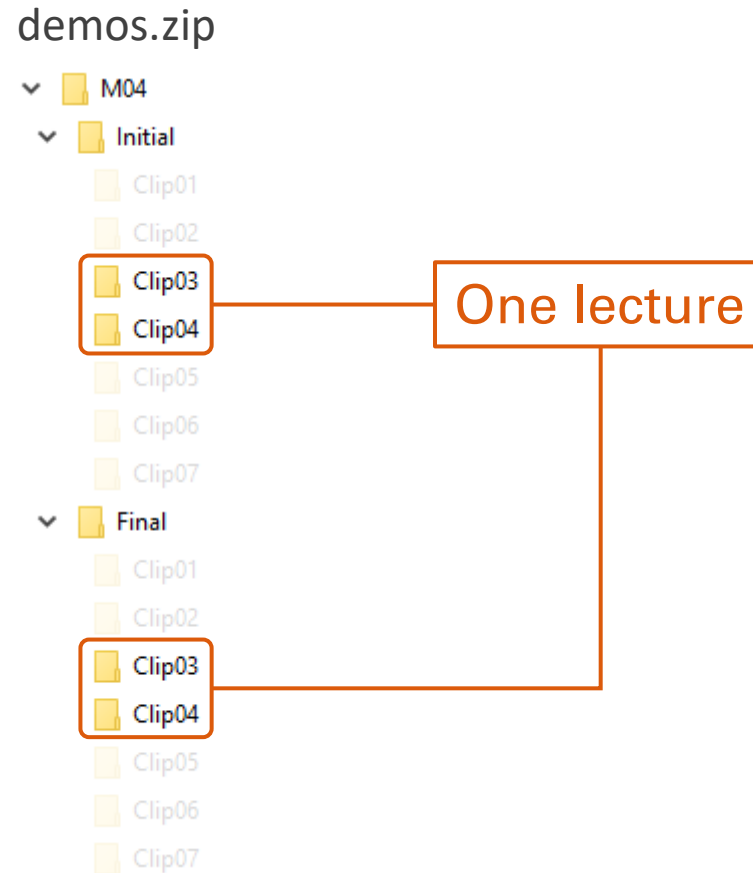
demos.zip



Finding initial code
for clip 3 in module 4

Structure of This Course

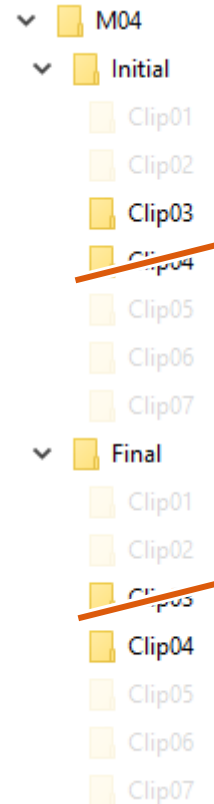
Decorator
Adapter
Strategy
Abstract Factory
Factory Method
Builder
Composite



Structure of This Course

Decorator
Adapter
Strategy
Abstract Factory
Factory Method
Builder
Composite

demos.zip



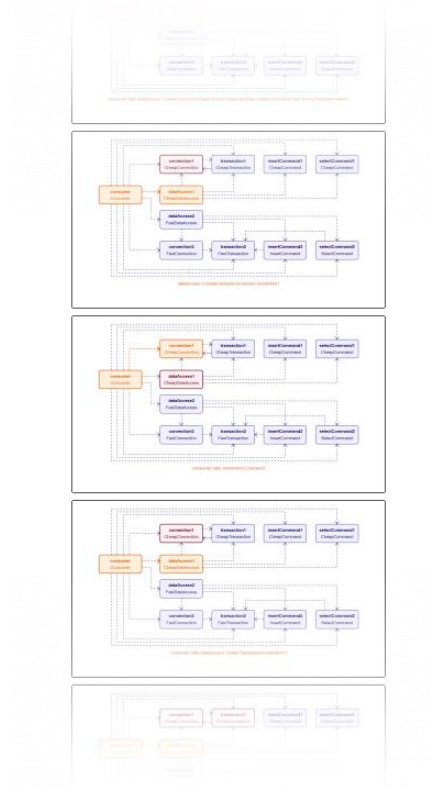
Structure of This Course

Decorator
Adapter
Strategy
Abstract Factory
Factory Method
Builder
Composite

demos.zip

- ▼ M04
 - ▼ Initial
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07
 - ▼ Final
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07

slides.pdf



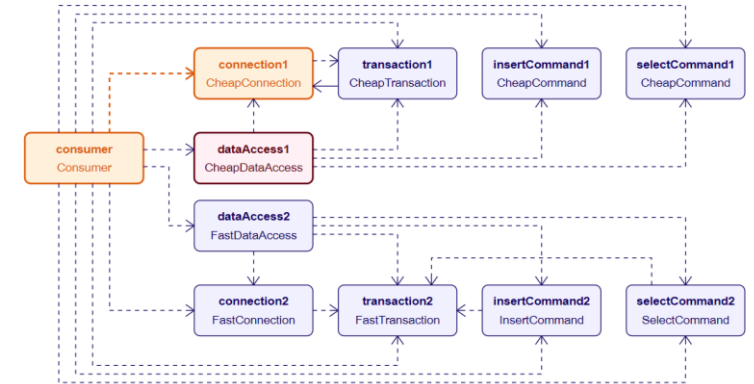
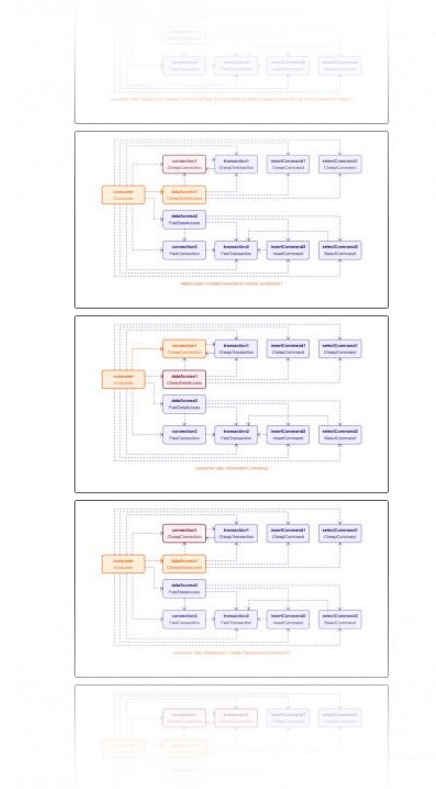
Structure of This Course

Decorator
Adapter
Strategy
Abstract Factory
Factory Method
Builder
Composite

demos.zip

- ▼ M04
 - ▼ Initial
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07
 - ▼ Final
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07

slides.pdf



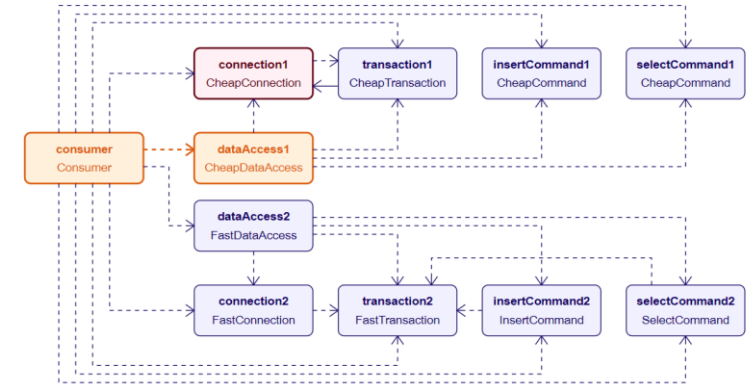
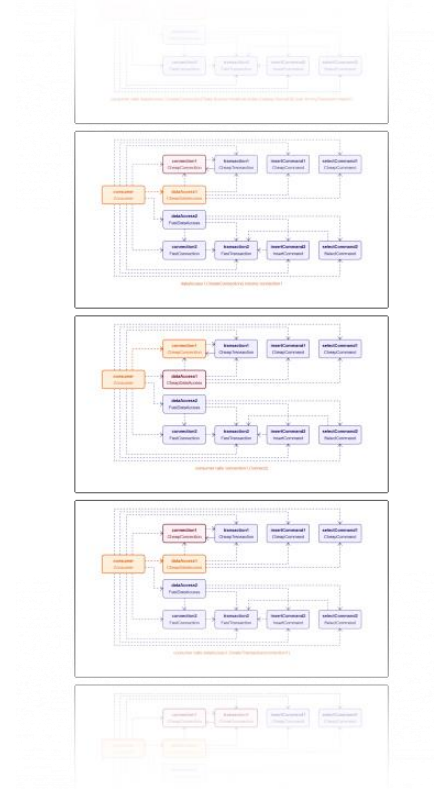
Structure of This Course

Decorator
Adapter
Strategy
Abstract Factory
Factory Method
Builder
Composite

demos.zip

- ▼ M04
 - ▼ Initial
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07
 - ▼ Final
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07

slides.pdf



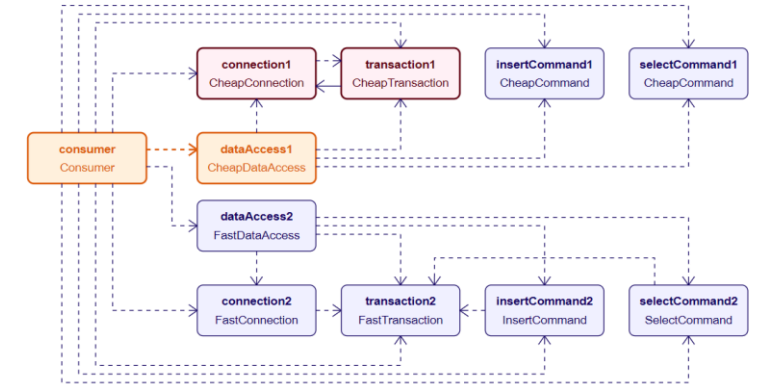
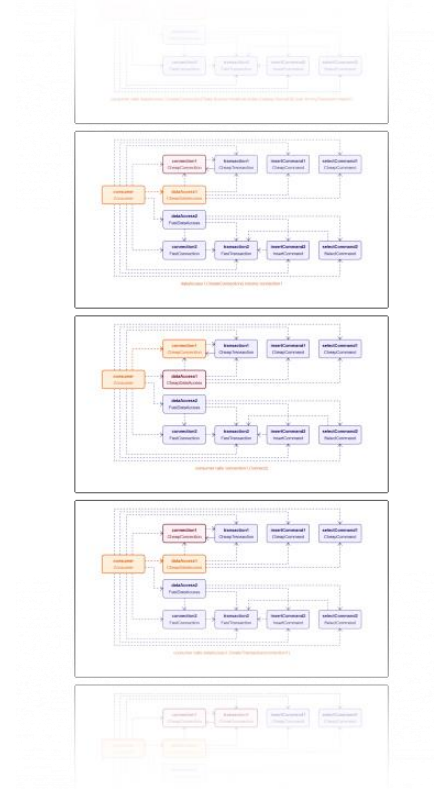
Structure of This Course

Decorator
Adapter
Strategy
Abstract Factory
Factory Method
Builder
Composite

demos.zip

- ▼ M04
 - ▼ Initial
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07
 - ▼ Final
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07

slides.pdf



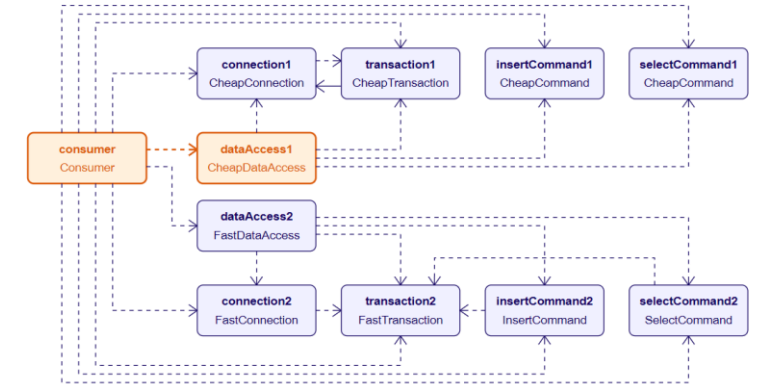
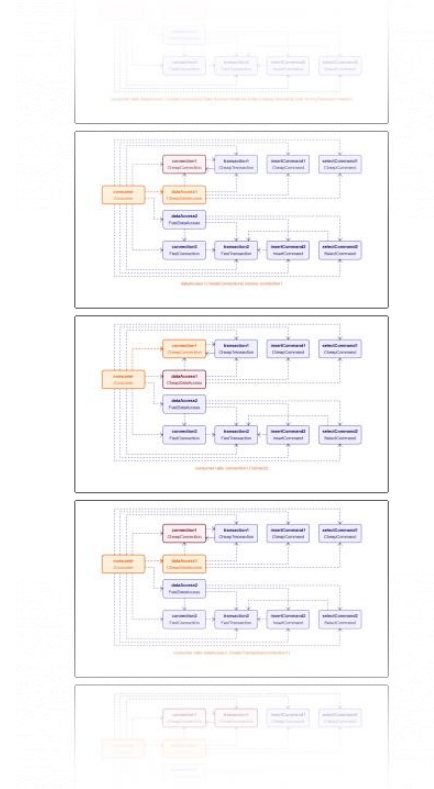
Structure of This Course

Decorator
Adapter
Strategy
Abstract Factory
Factory Method
Builder
Composite

demos.zip

- ▼ M04
 - ▼ Initial
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07
 - ▼ Final
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07

slides.pdf



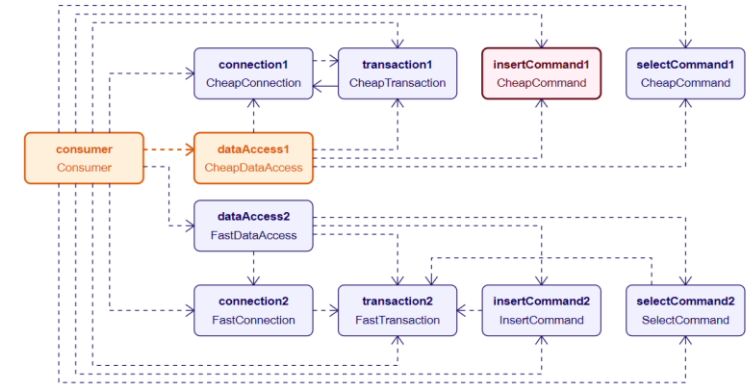
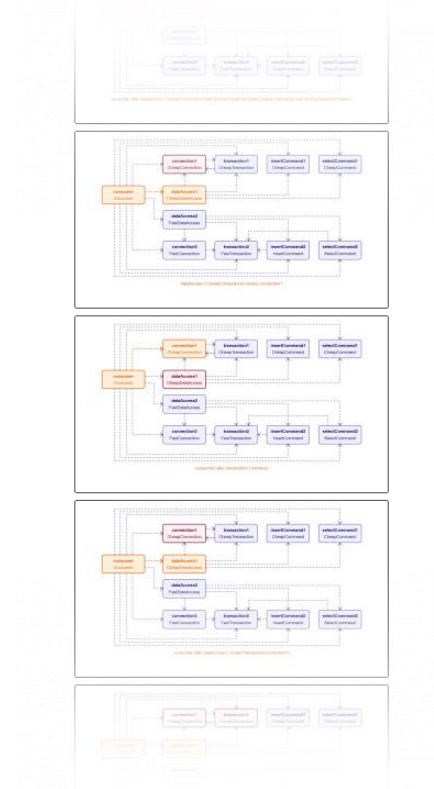
Structure of This Course

Decorator
Adapter
Strategy
Abstract Factory
Factory Method
Builder
Composite

demos.zip

- ▼ M04
 - ▼ Initial
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07
 - ▼ Final
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07

slides.pdf



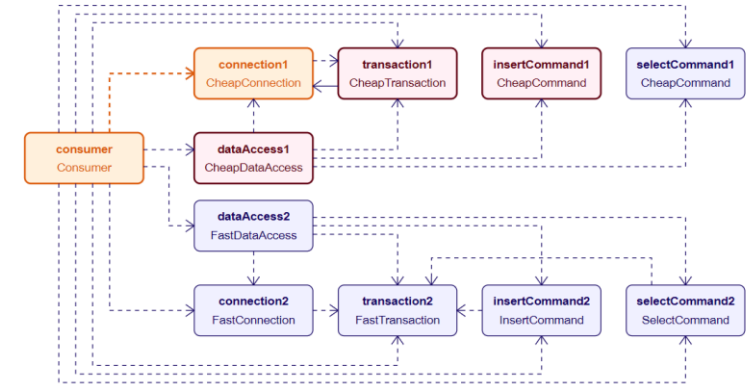
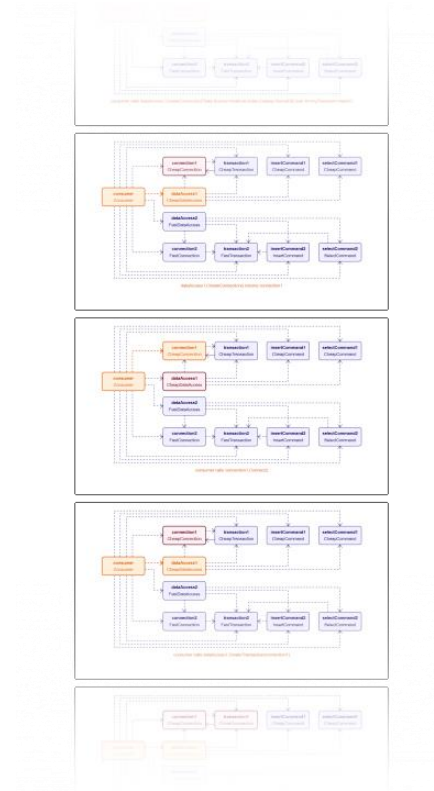
Structure of This Course

Decorator
Adapter
Strategy
Abstract Factory
Factory Method
Builder
Composite

demos.zip

- ▼ M04
 - ▼ Initial
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07
 - ▼ Final
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07

slides.pdf



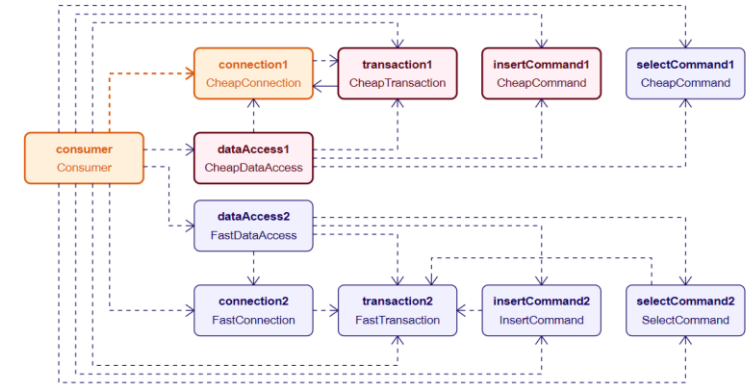
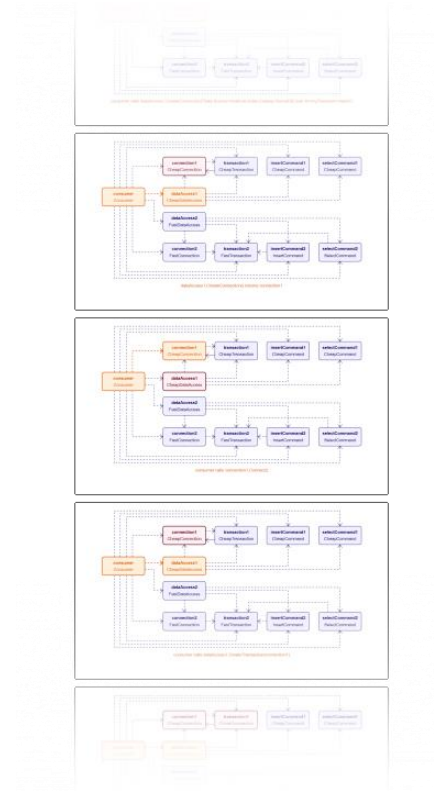
Structure of This Course

Decorator
Adapter
Strategy
Abstract Factory
Factory Method
Builder
Composite

demos.zip

- ▼ M04
 - ▼ Initial
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07
 - ▼ Final
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07

slides.pdf



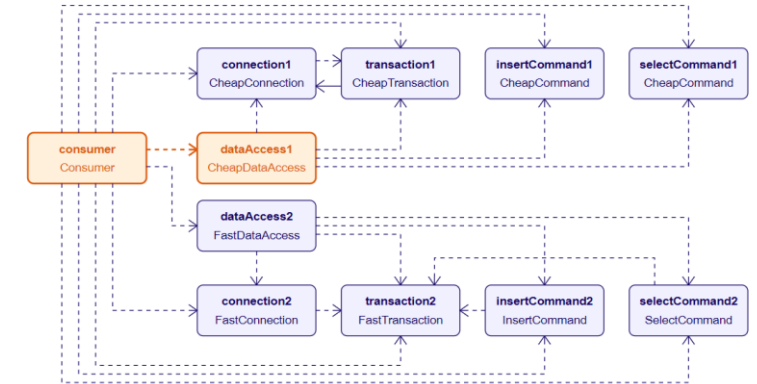
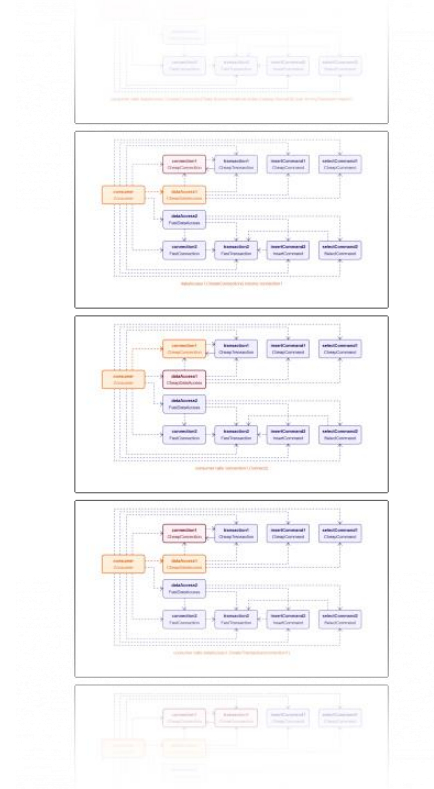
Structure of This Course

Decorator
Adapter
Strategy
Abstract Factory
Factory Method
Builder
Composite

demos.zip

- ▼ M04
 - ▼ Initial
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07
 - ▼ Final
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07

slides.pdf



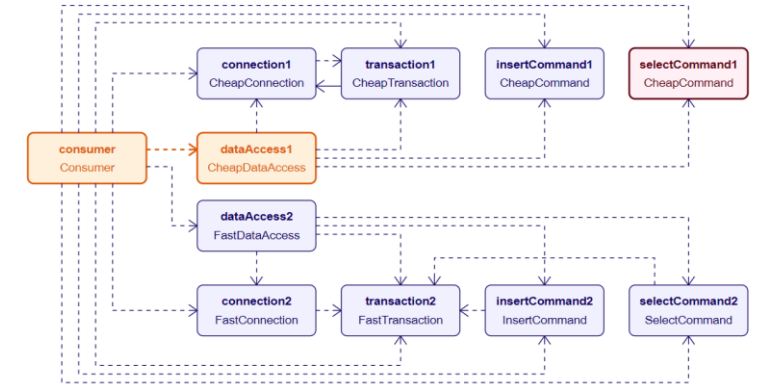
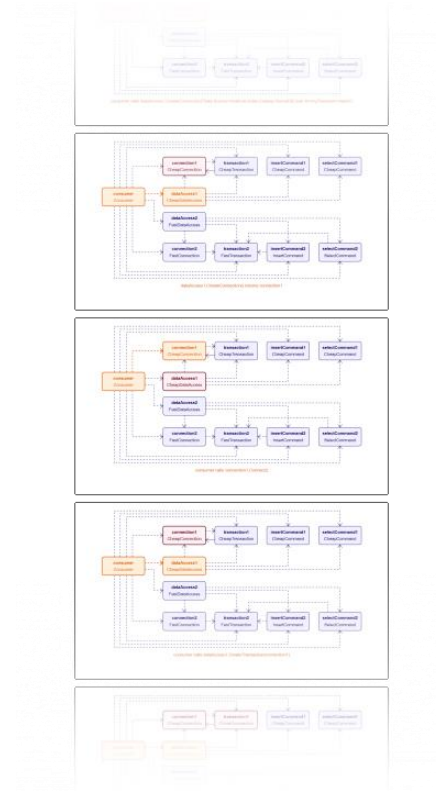
Structure of This Course

Decorator
Adapter
Strategy
Abstract Factory
Factory Method
Builder
Composite

demos.zip

- ▼ M04
 - ▼ Initial
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07
 - ▼ Final
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07

slides.pdf



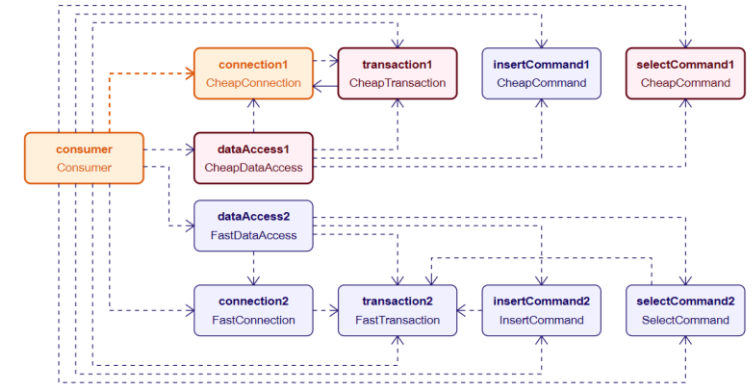
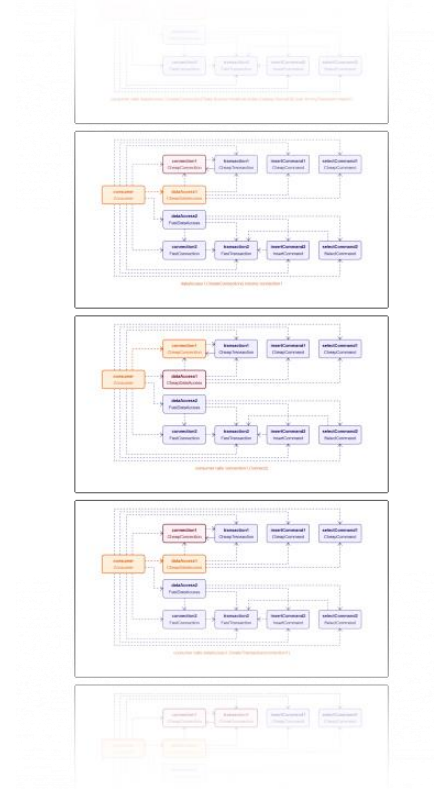
Structure of This Course

Decorator
Adapter
Strategy
Abstract Factory
Factory Method
Builder
Composite

demos.zip

- ▼ M04
 - ▼ Initial
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07
 - ▼ Final
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07

slides.pdf



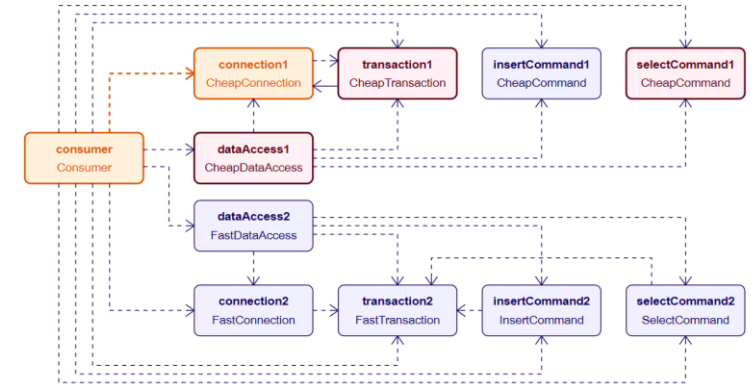
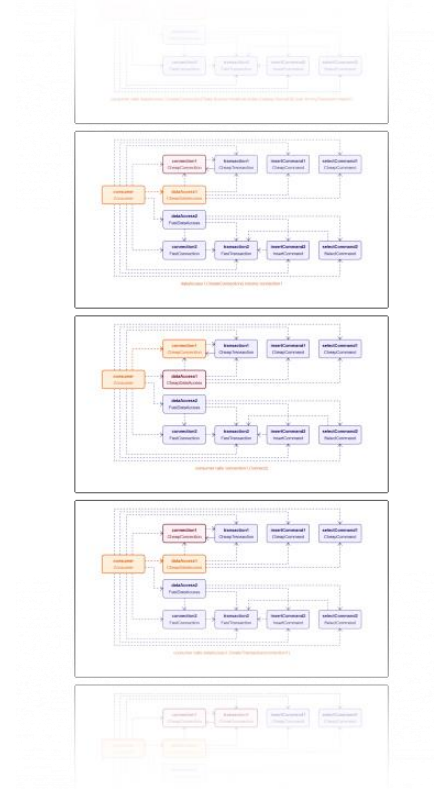
Structure of This Course

Decorator
Adapter
Strategy
Abstract Factory
Factory Method
Builder
Composite

demos.zip

- ▼ M04
 - ▼ Initial
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07
 - ▼ Final
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07

slides.pdf



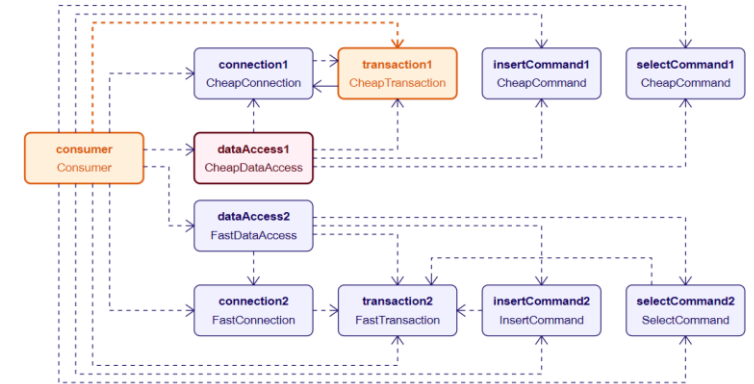
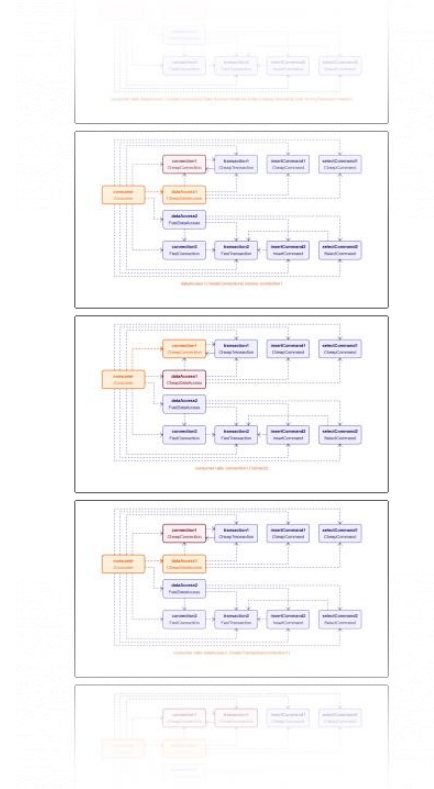
Structure of This Course

Decorator
Adapter
Strategy
Abstract Factory
Factory Method
Builder
Composite

demos.zip

- ▼ M04
 - ▼ Initial
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07
 - ▼ Final
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07

slides.pdf



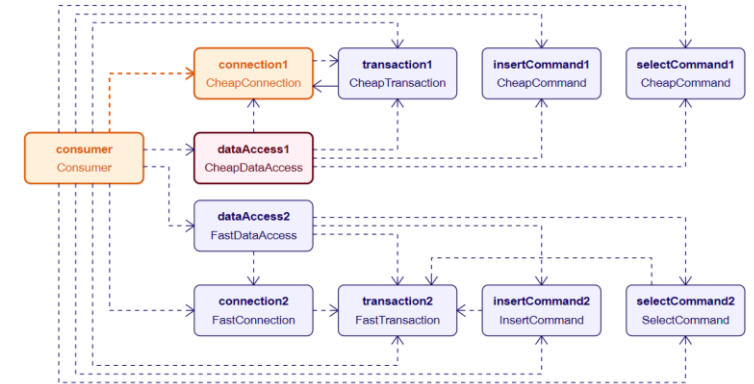
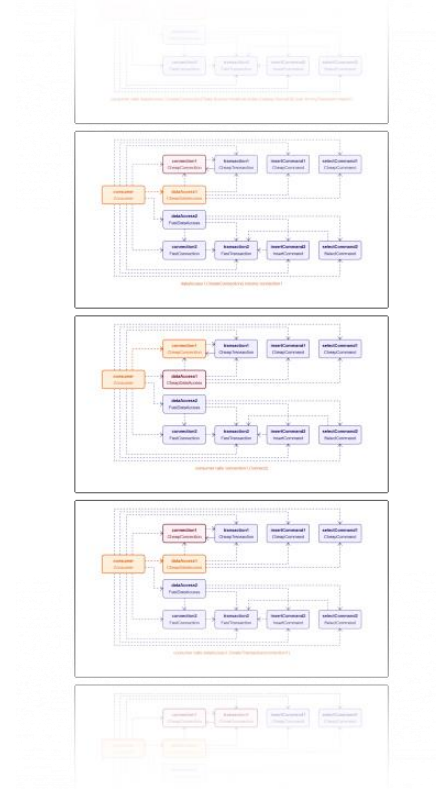
Structure of This Course

Decorator
Adapter
Strategy
Abstract Factory
Factory Method
Builder
Composite

demos.zip

- ▼ M04
 - ▼ Initial
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07
 - ▼ Final
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07

slides.pdf



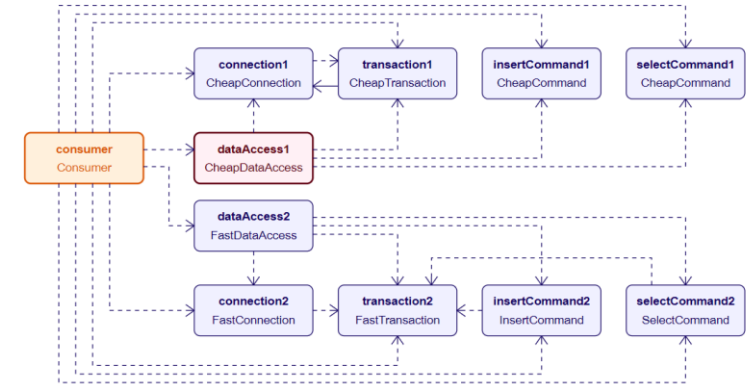
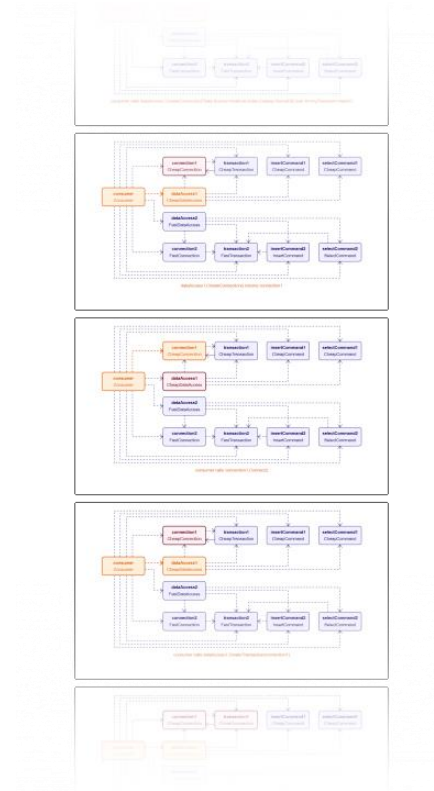
Structure of This Course

Decorator
Adapter
Strategy
Abstract Factory
Factory Method
Builder
Composite

demos.zip

- ▼ M04
 - ▼ Initial
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07
 - ▼ Final
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07

slides.pdf



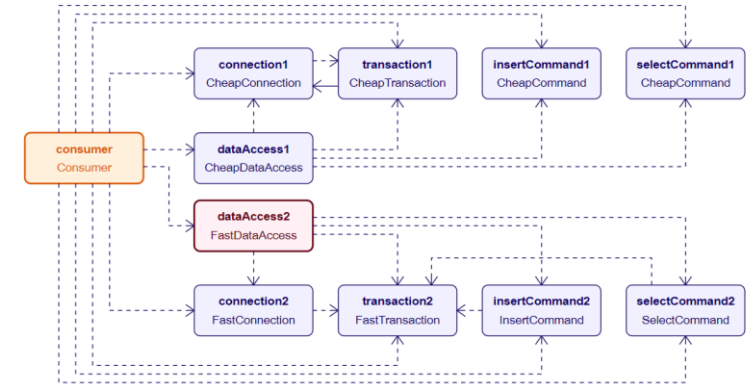
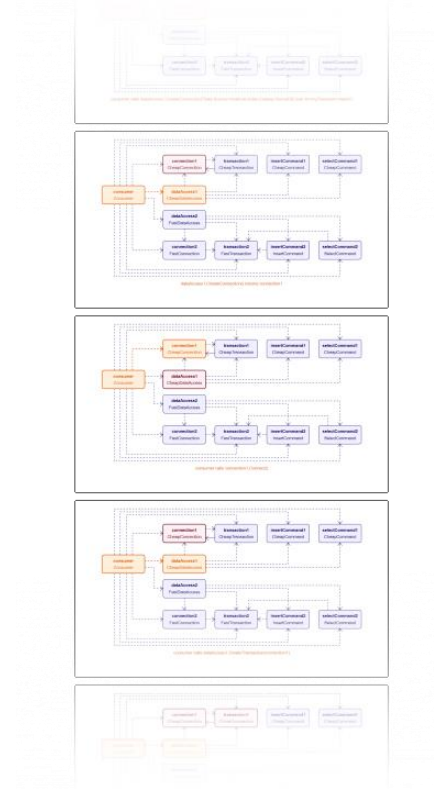
Structure of This Course

Decorator
Adapter
Strategy
Abstract Factory
Factory Method
Builder
Composite

demos.zip

- ▼ M04
 - ▼ Initial
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07
 - ▼ Final
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07

slides.pdf



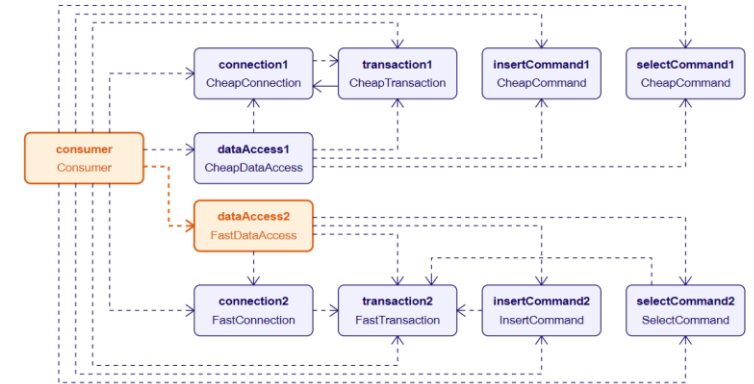
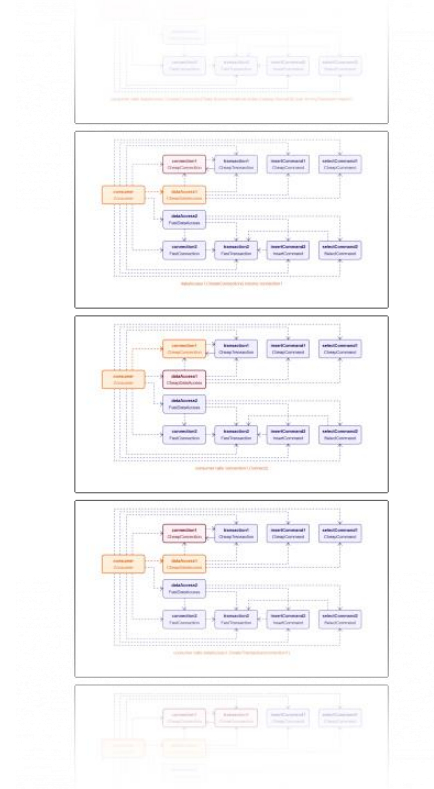
Structure of This Course

Decorator
Adapter
Strategy
Abstract Factory
Factory Method
Builder
Composite

demos.zip

- ▼ M04
 - ▼ Initial
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07
 - ▼ Final
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07

slides.pdf



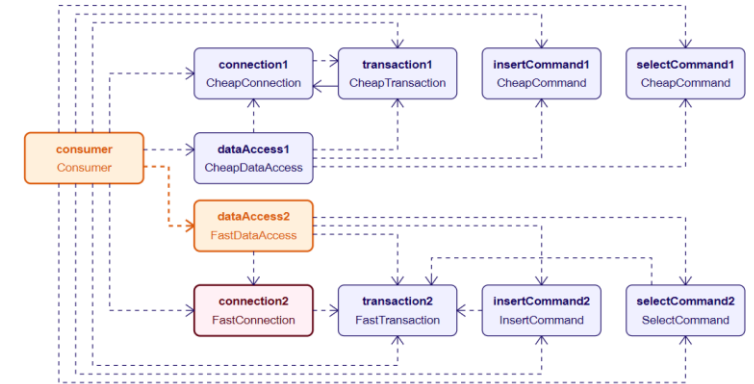
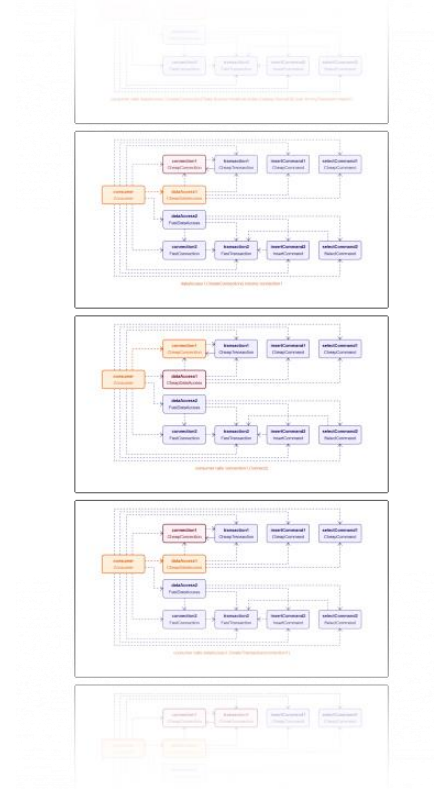
Structure of This Course

Decorator
Adapter
Strategy
Abstract Factory
Factory Method
Builder
Composite

demos.zip

- ▼ M04
 - ▼ Initial
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07
 - ▼ Final
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07

slides.pdf



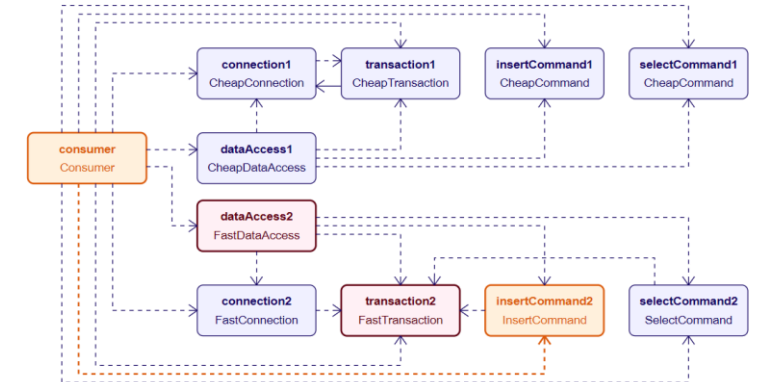
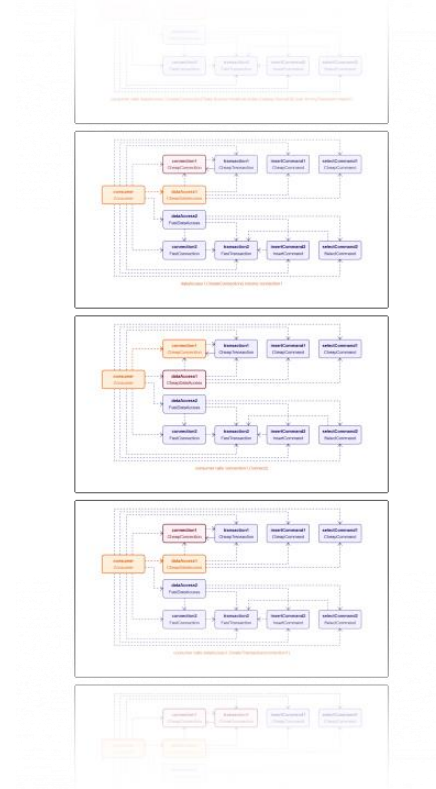
Structure of This Course

Decorator
Adapter
Strategy
Abstract Factory
Factory Method
Builder
Composite

demos.zip

- ▼ M04
 - ▼ Initial
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07
 - ▼ Final
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07

slides.pdf



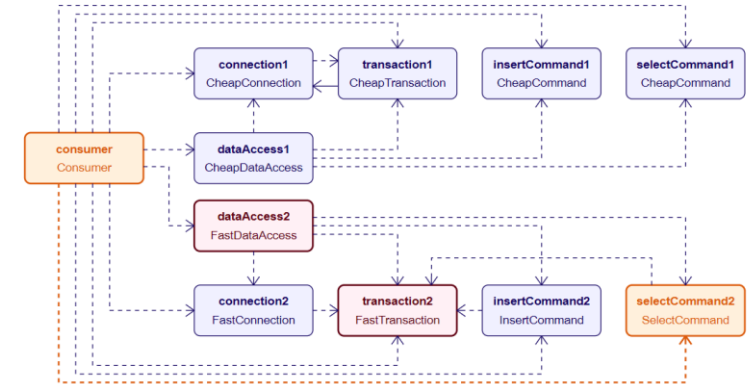
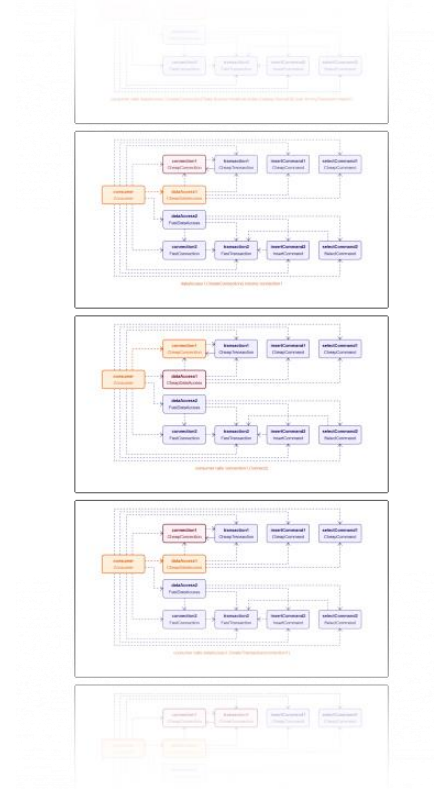
Structure of This Course

Decorator
Adapter
Strategy
Abstract Factory
Factory Method
Builder
Composite

demos.zip

- ▼ M04
 - ▼ Initial
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07
 - ▼ Final
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07

slides.pdf



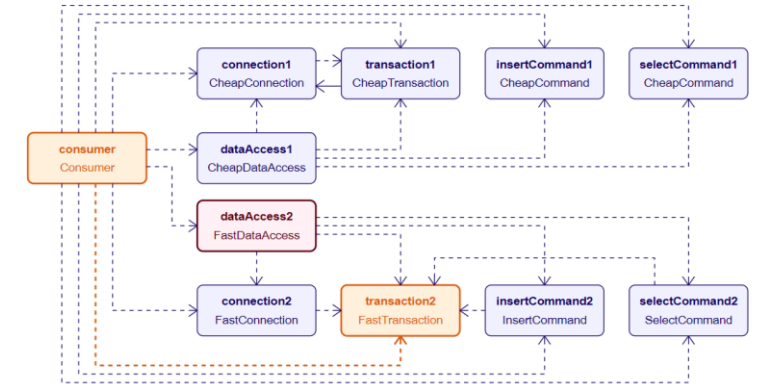
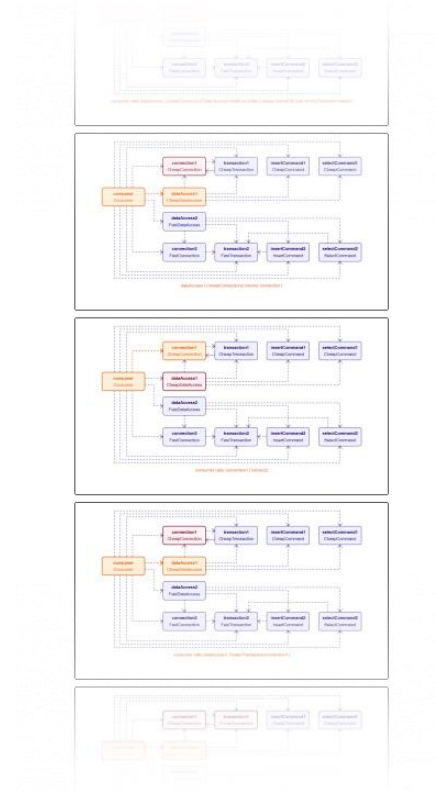
Structure of This Course

Decorator
Adapter
Strategy
Abstract Factory
Factory Method
Builder
Composite

demos.zip

- ▼ M04
 - ▼ Initial
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07
 - ▼ Final
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07

slides.pdf



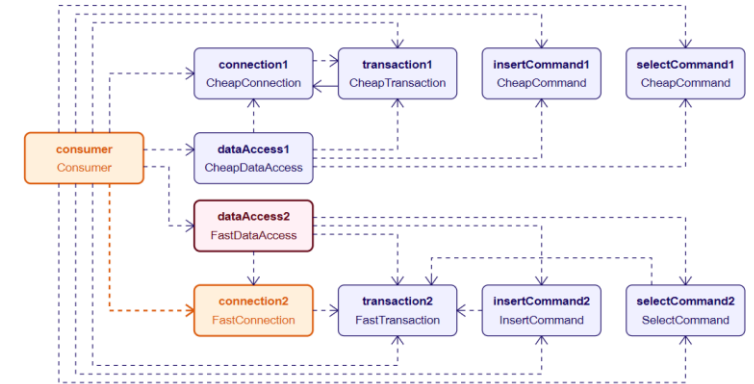
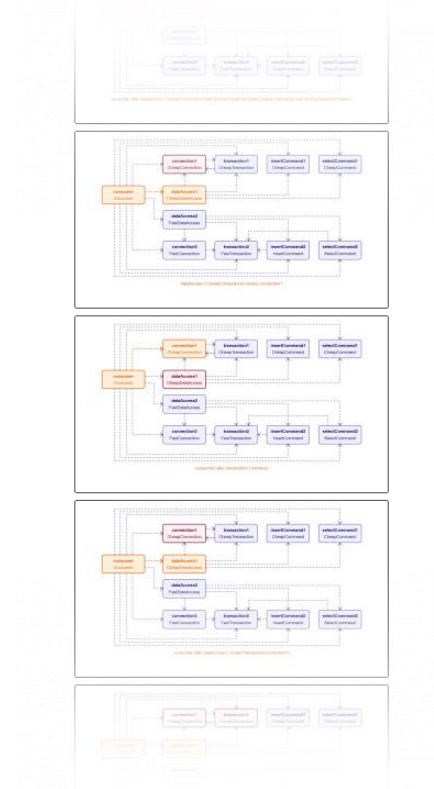
Structure of This Course

Decorator
Adapter
Strategy
Abstract Factory
Factory Method
Builder
Composite

demos.zip

- ▼ M04
 - ▼ Initial
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07
 - ▼ Final
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07

slides.pdf



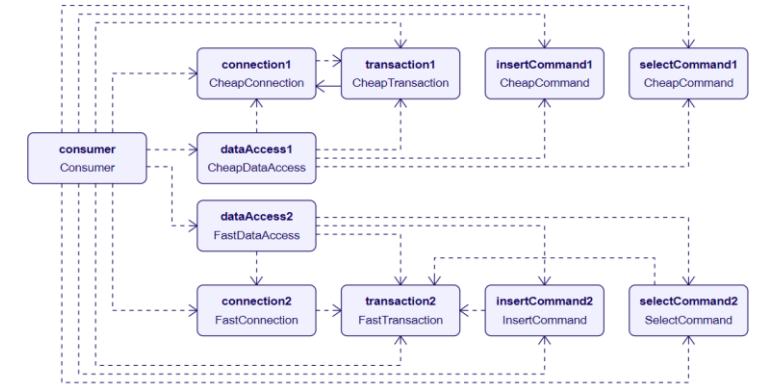
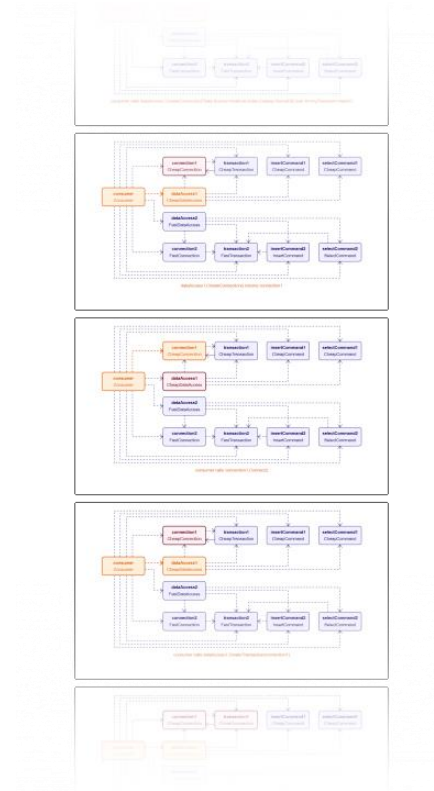
Structure of This Course

Decorator
Adapter
Strategy
Abstract Factory
Factory Method
Builder
Composite

demos.zip

- ▼ M04
 - ▼ Initial
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07
 - ▼ Final
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07

slides.pdf



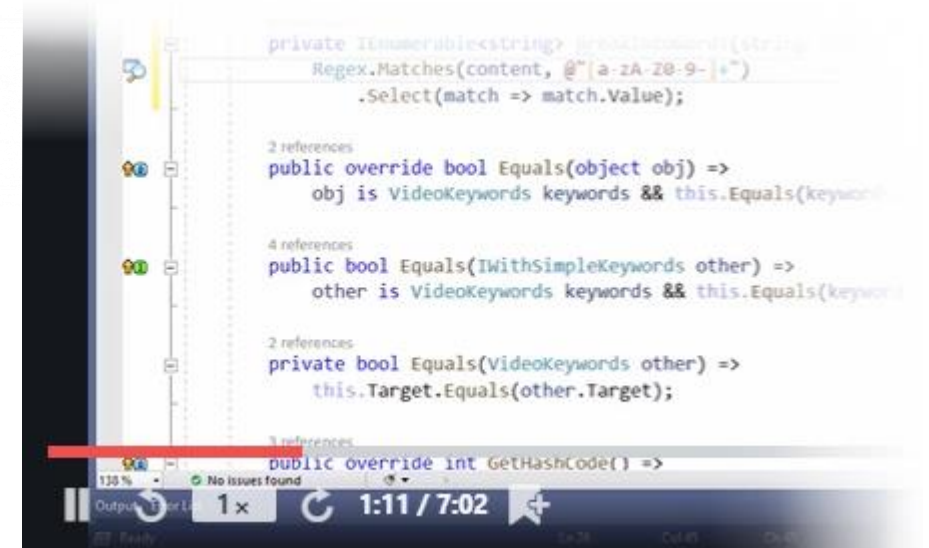
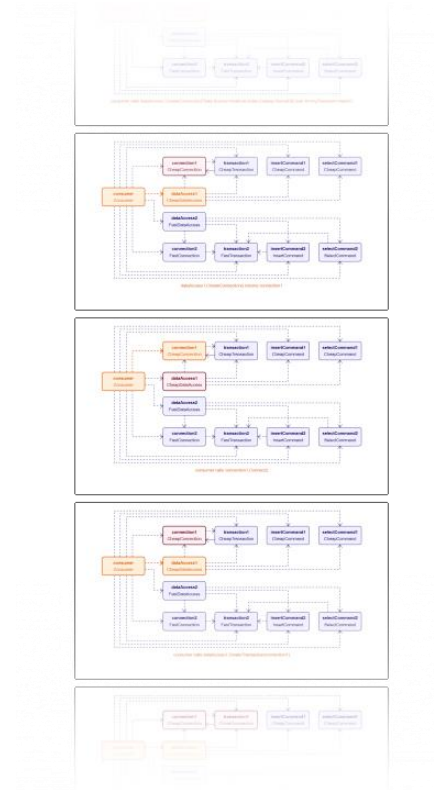
Structure of This Course

Decorator
Adapter
Strategy
Abstract Factory
Factory Method
Builder
Composite

demos.zip

- ▼ M04
 - ▼ Initial
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07
 - ▼ Final
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07

slides.pdf



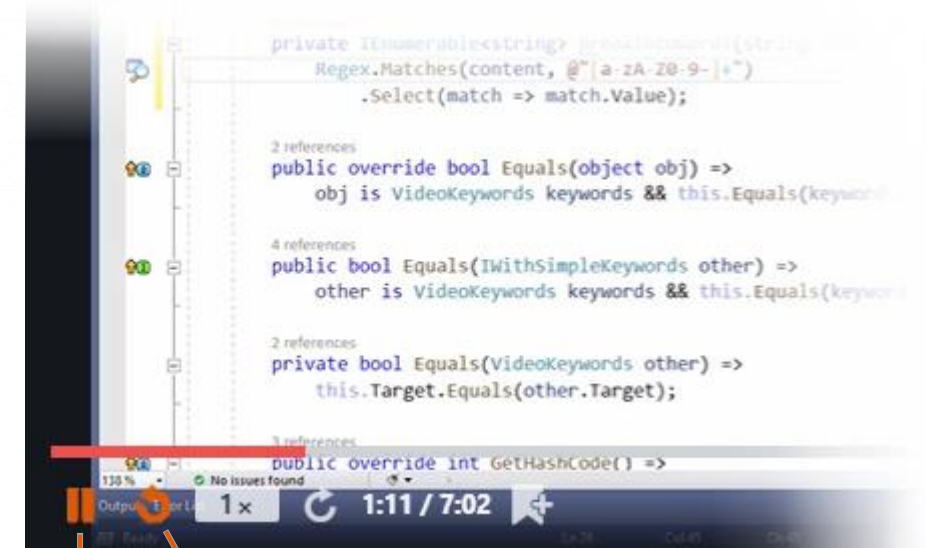
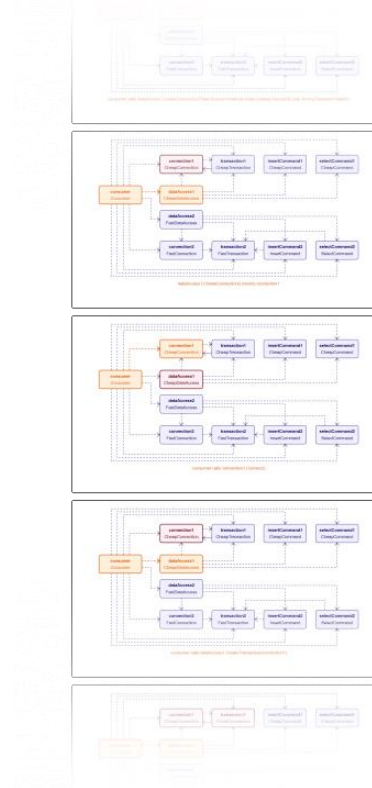
Structure of This Course

Decorator
Adapter
Strategy
Abstract Factory
Factory Method
Builder
Composite

demos.zip

- ▼ M04
 - ▼ Initial
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07
 - ▼ Final
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07

slides.pdf



Replay video
when needed

Pause the video
when needed

Structure of This Course

Decorator
Adapter
Strategy
Abstract Factory
Factory Method
Builder
Composite

demos.zip

- ▼ M04
 - ▼ Initial
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07
 - ▼ Final
 - Clip01
 - Clip02
 - Clip03
 - Clip04
 - Clip05
 - Clip06
 - Clip07

slides.pdf

