

# Refactoring to Patterns with ReSharper

Zoran Horvat

Principal consultant at Coding Helmet

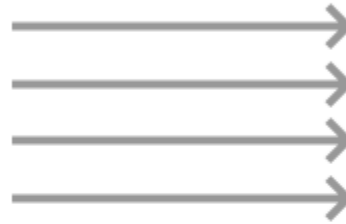
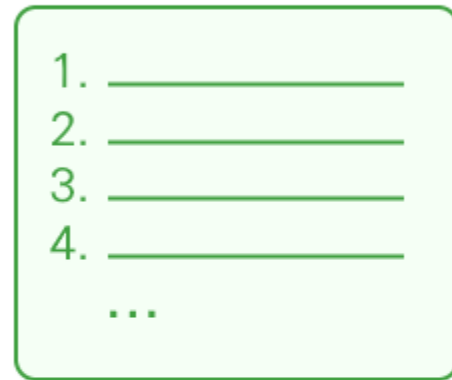
<http://codinghelmet.com>

[zh@codinghelmet.com](mailto:zh@codinghelmet.com)

 [zoranh75](#)

# Programmer at Work

Requirements

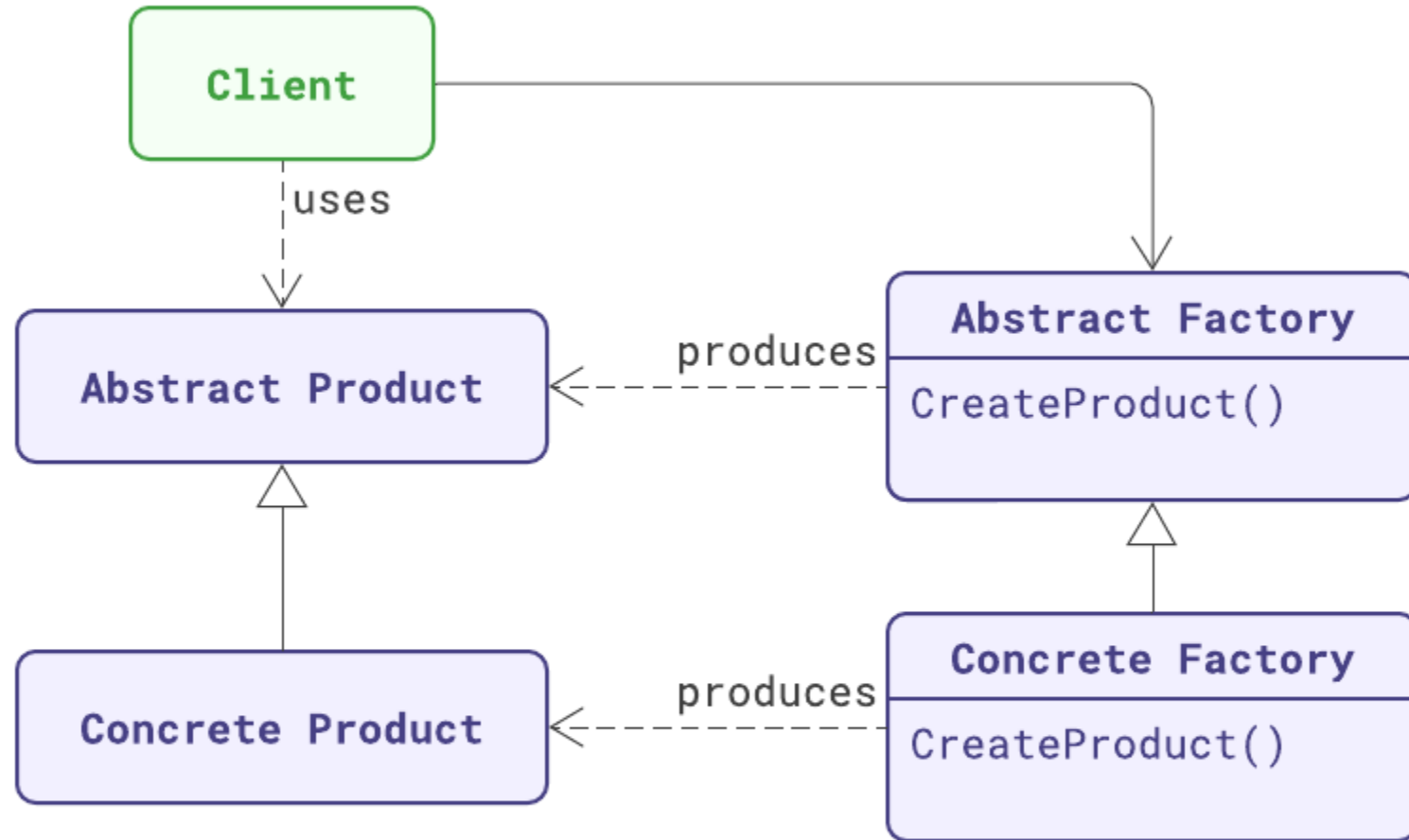


Implementation

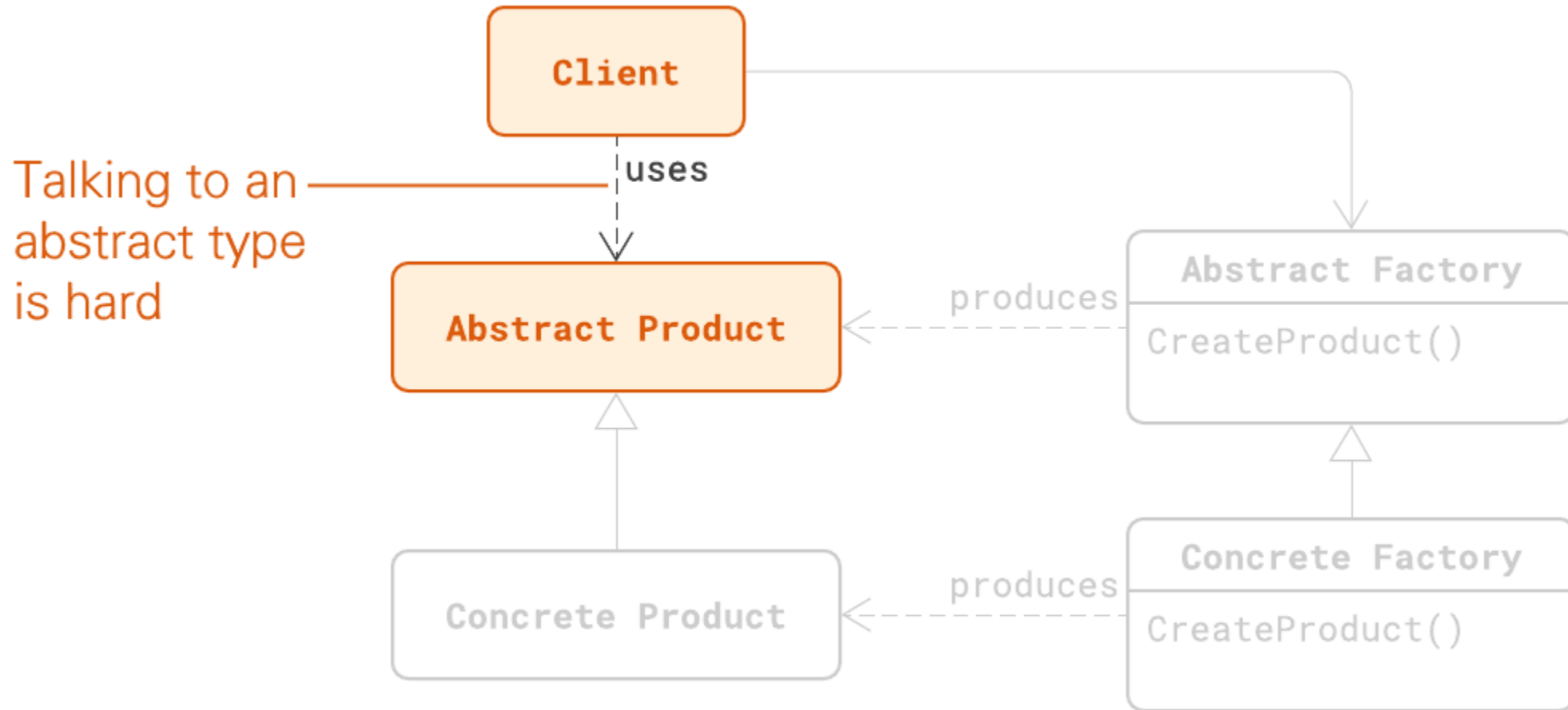


Any patterns here, yet?  
*What patterns!?*

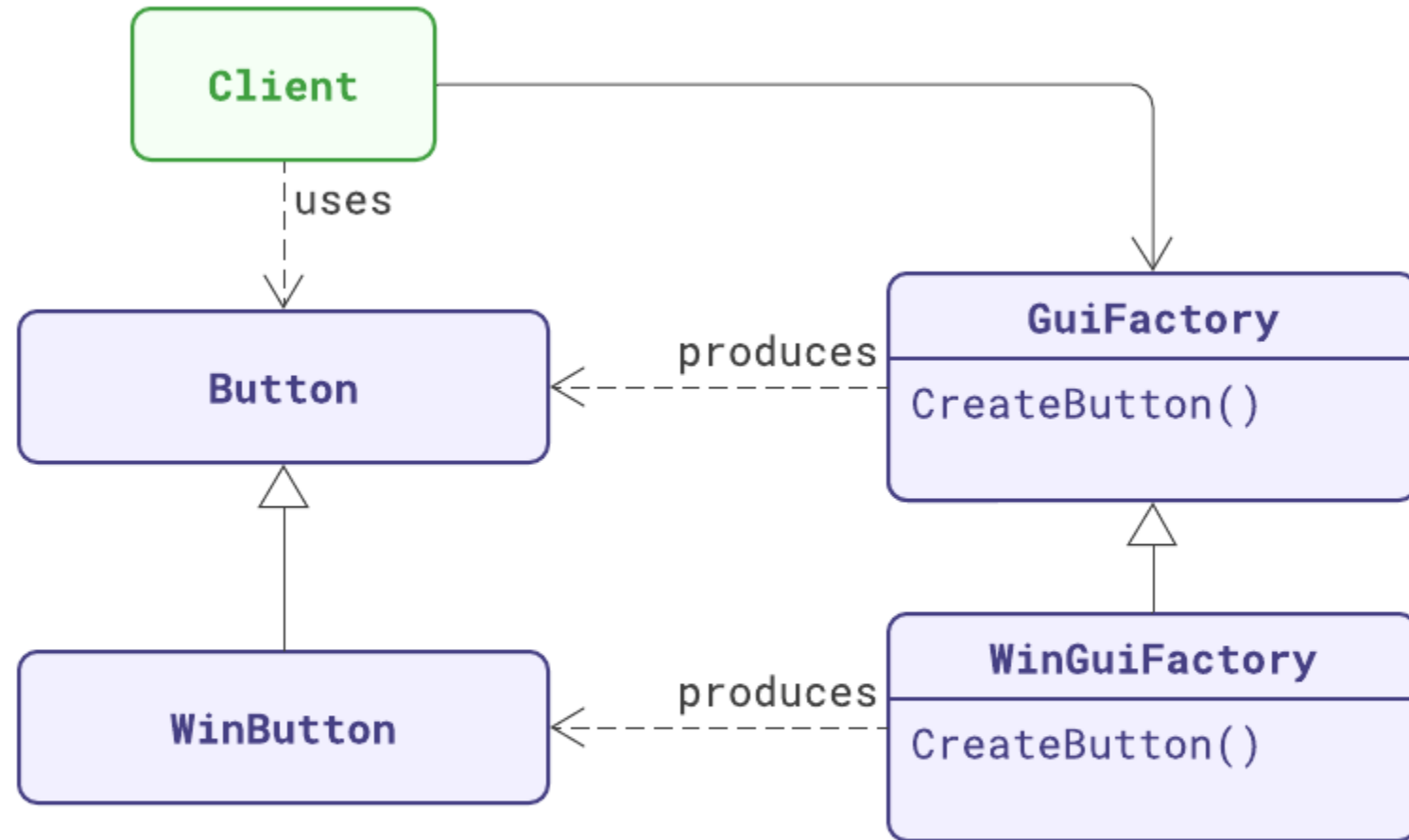
# Example: Abstract Factory



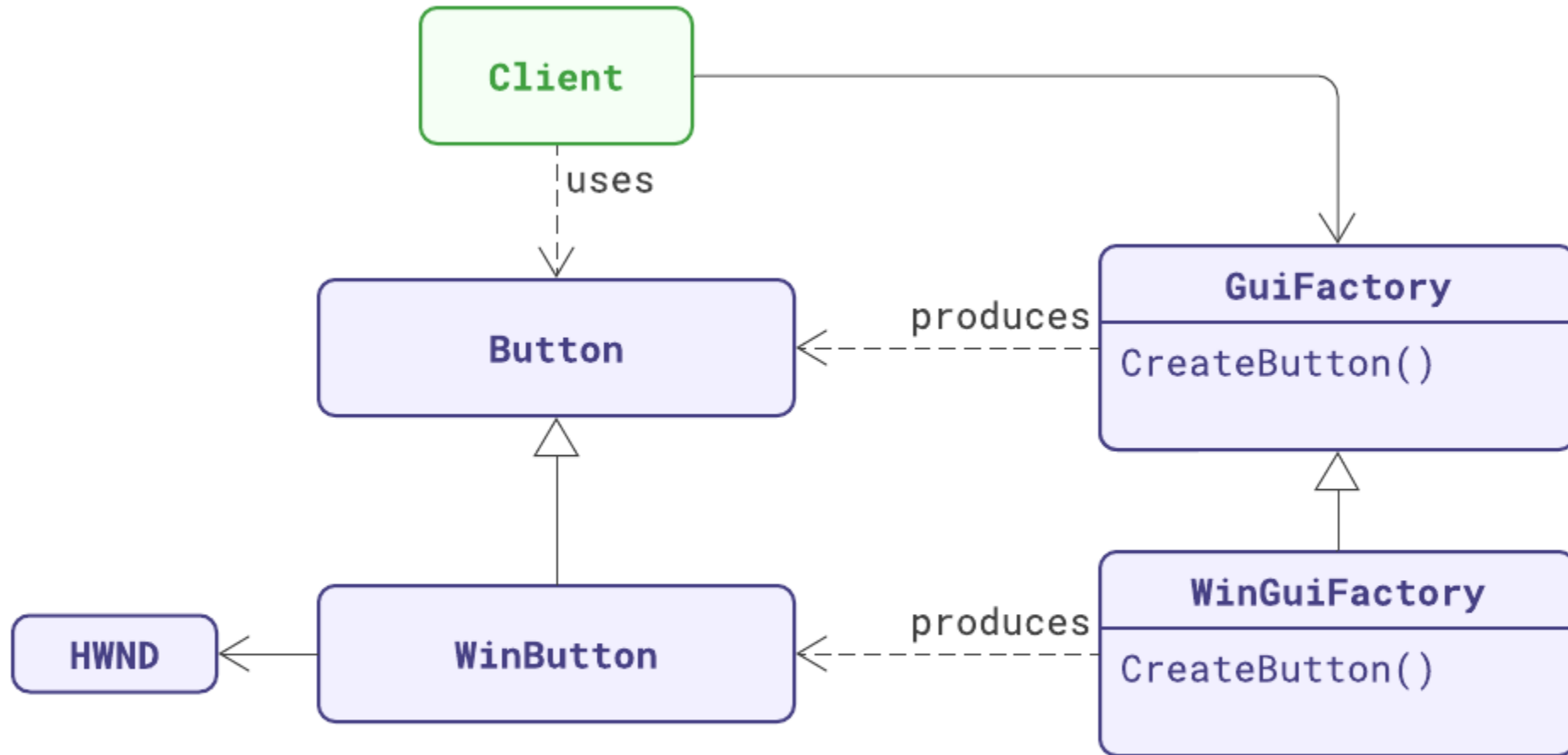
# Example: Abstract Factory



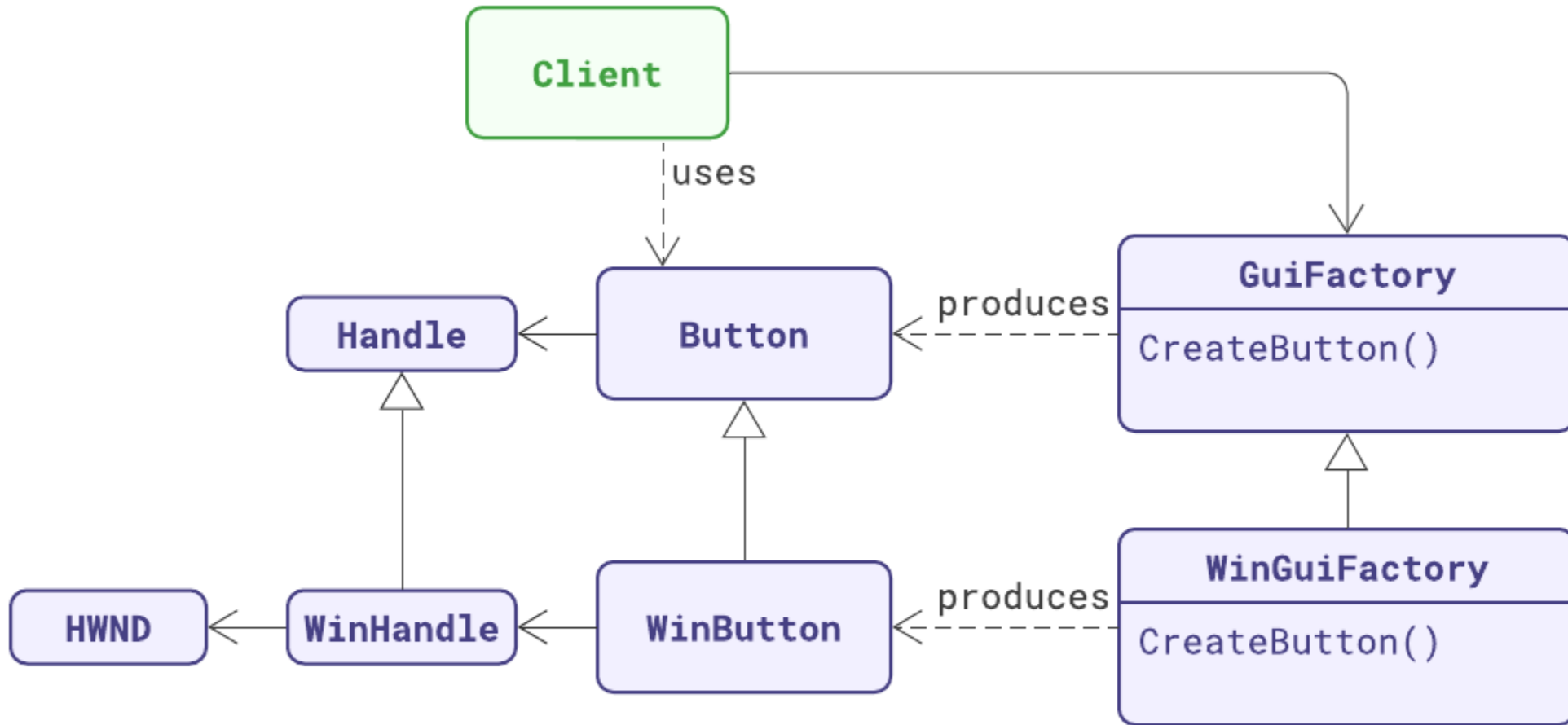
# Example: GUI Factory



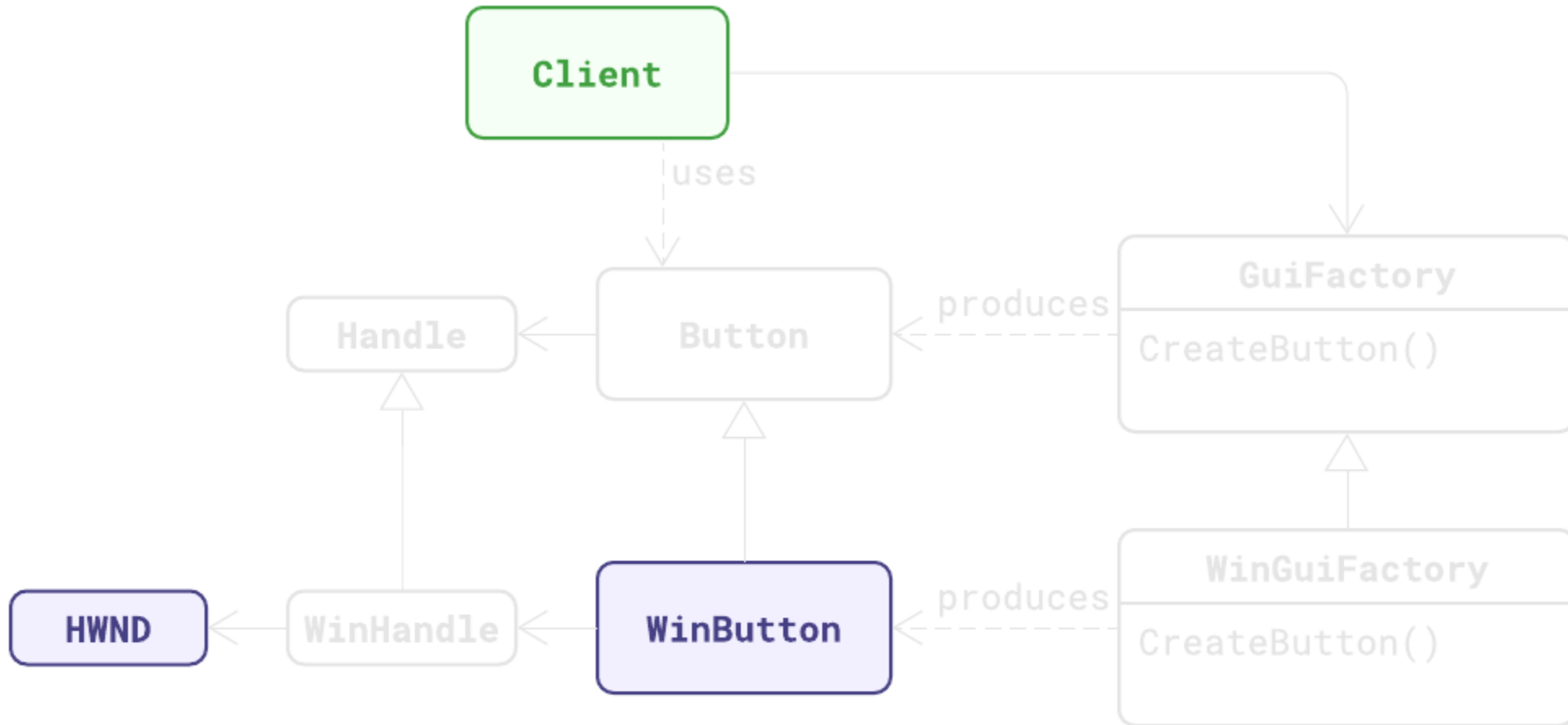
# Example: GUI Factory



# Example: GUI Factory

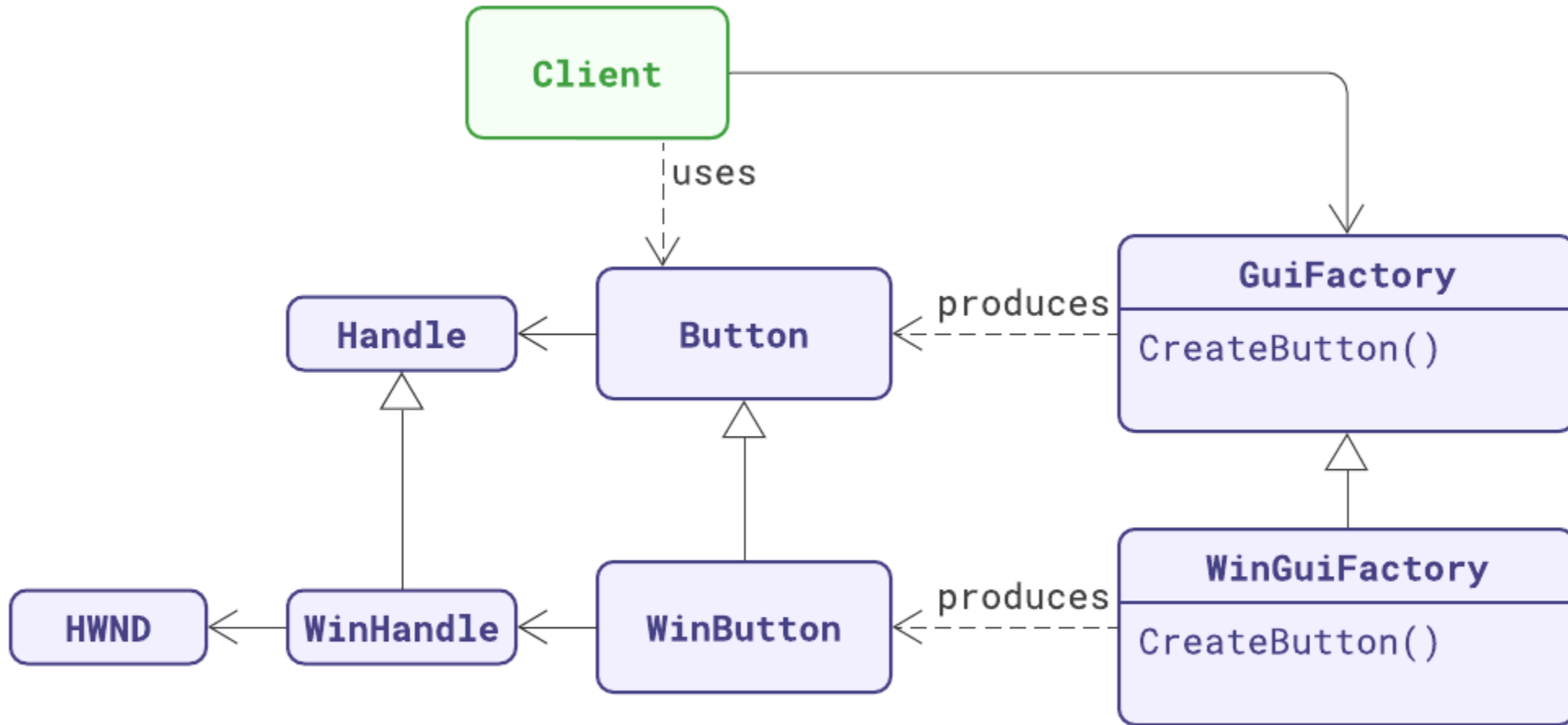


# Example: GUI Factory

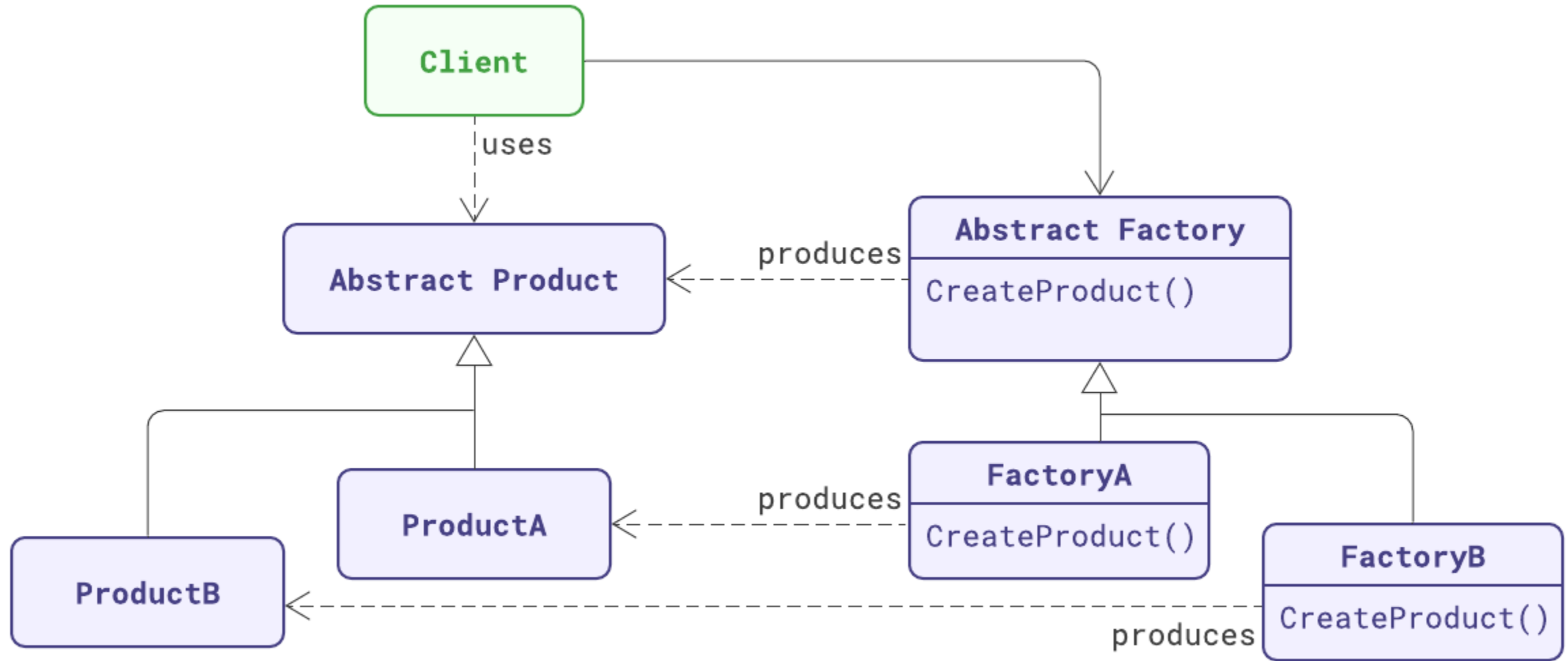




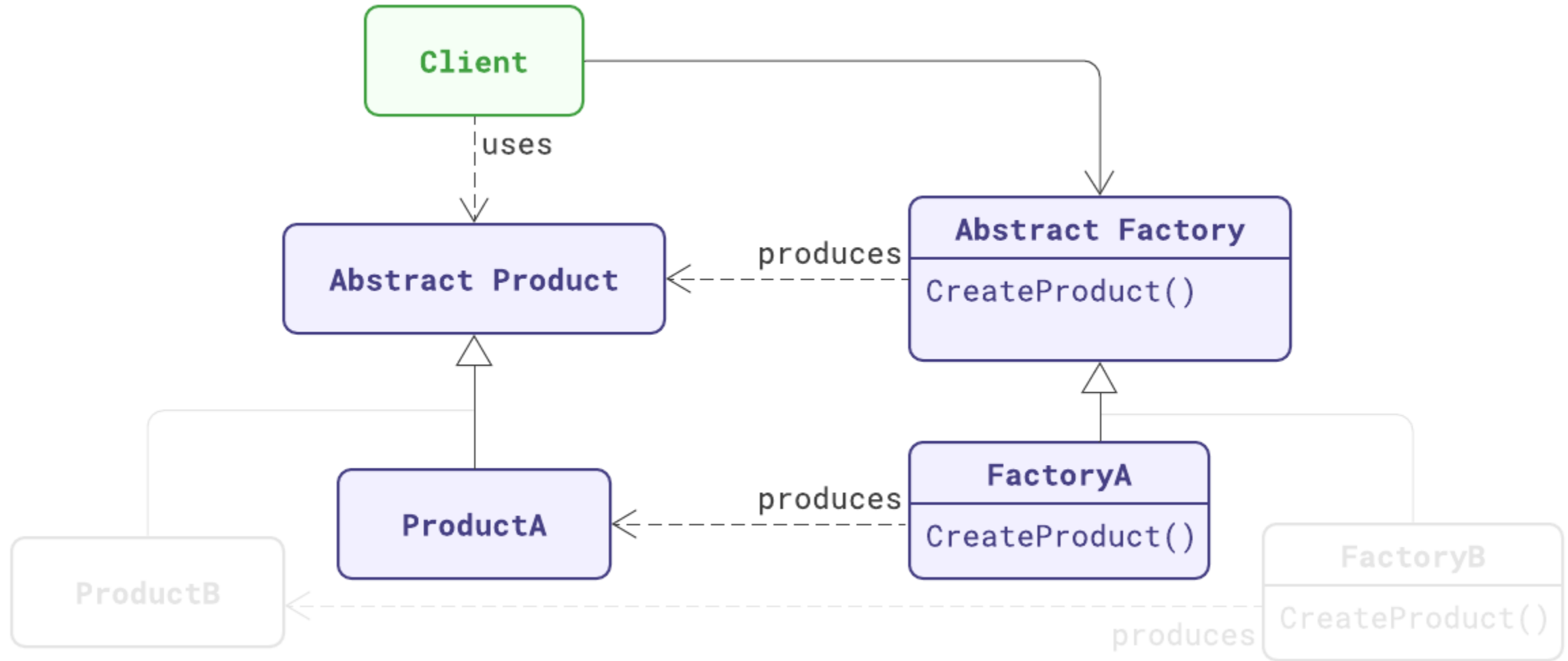
# Example: GUI Factory



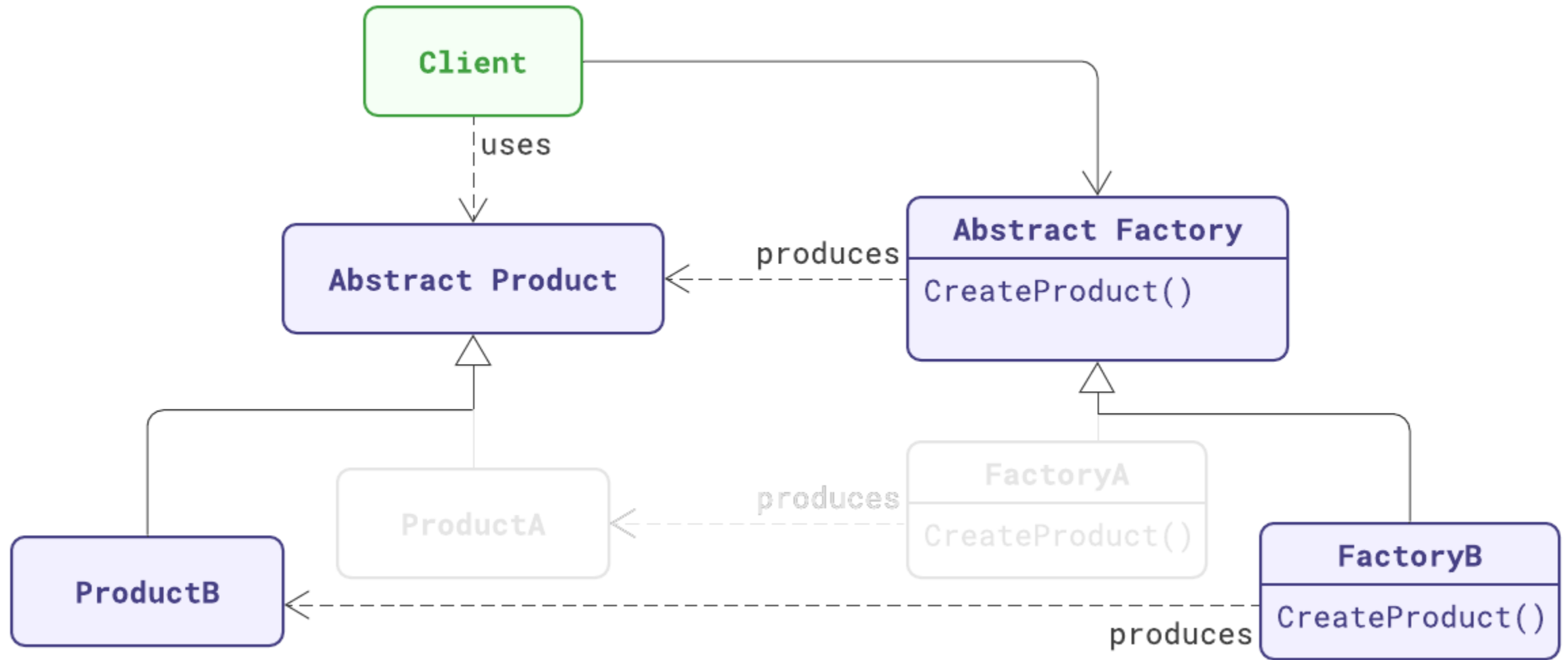
# Example: Abstract Factory



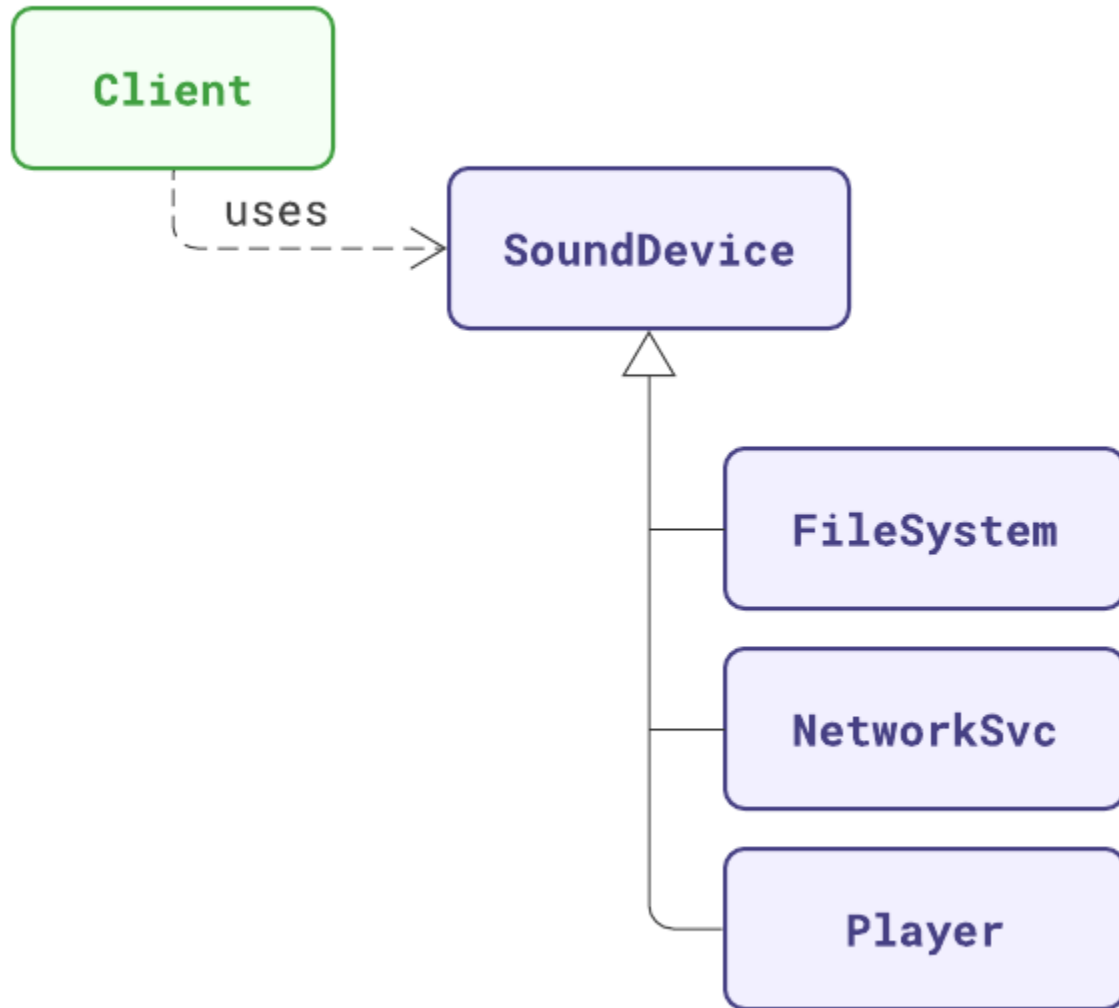
# Example: Abstract Factory



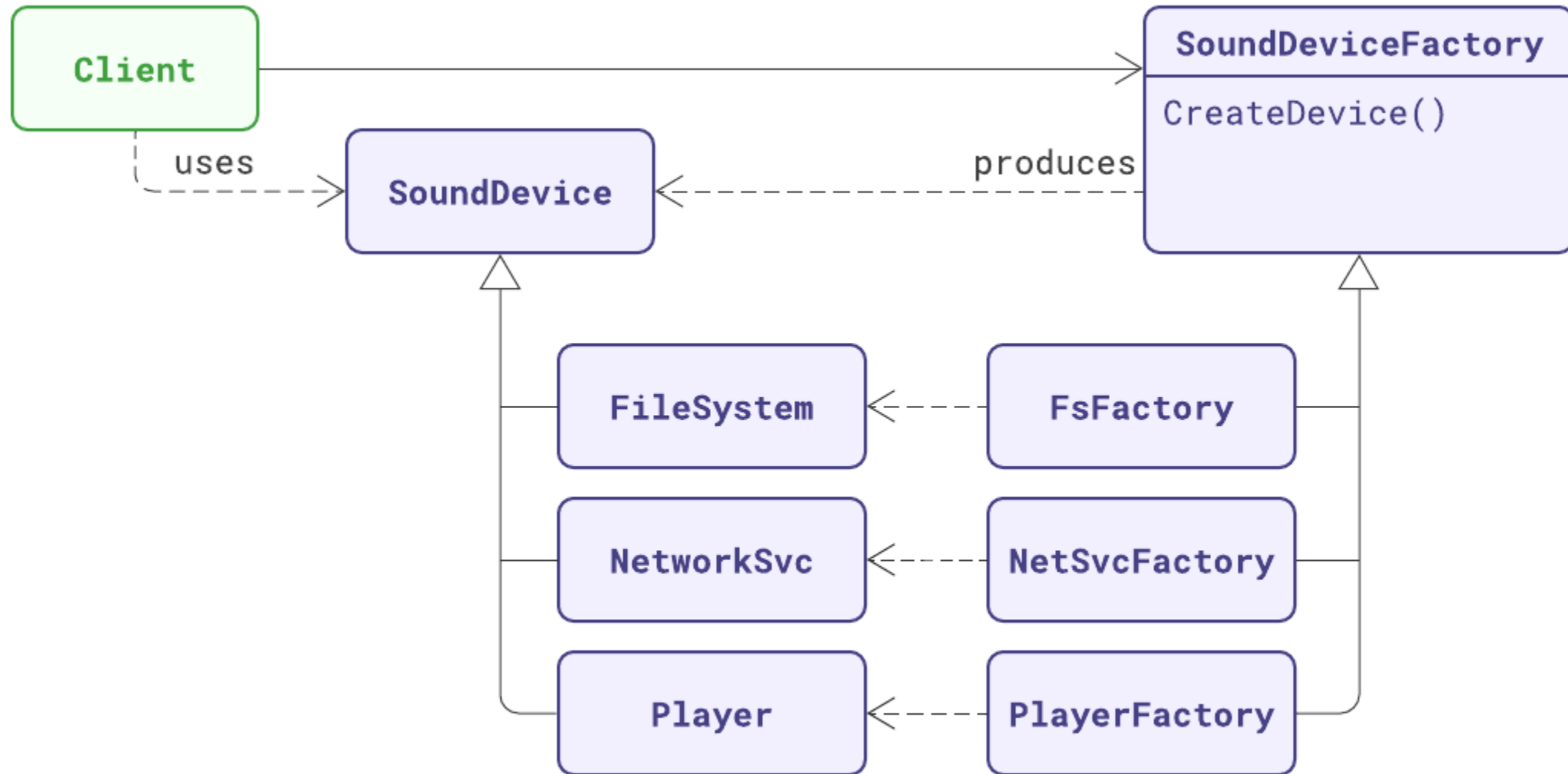
# Example: Abstract Factory



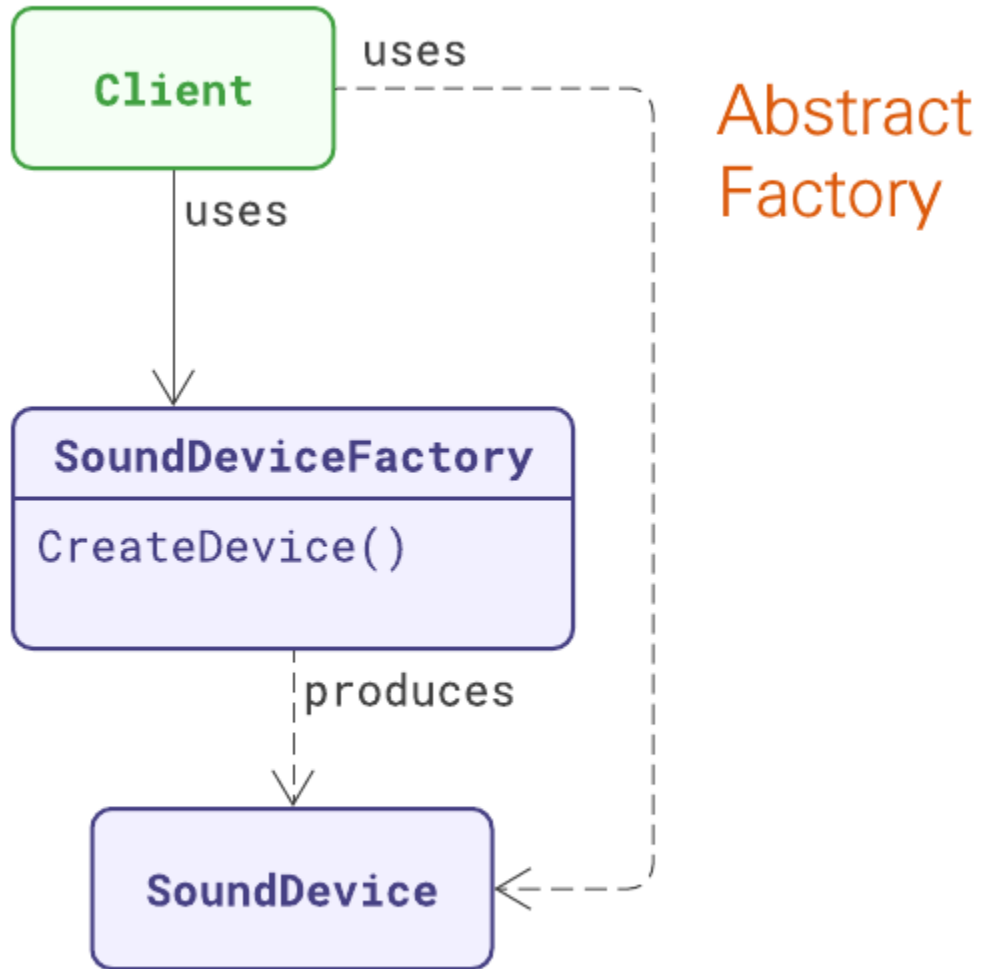
# Example: Sound Processing Library



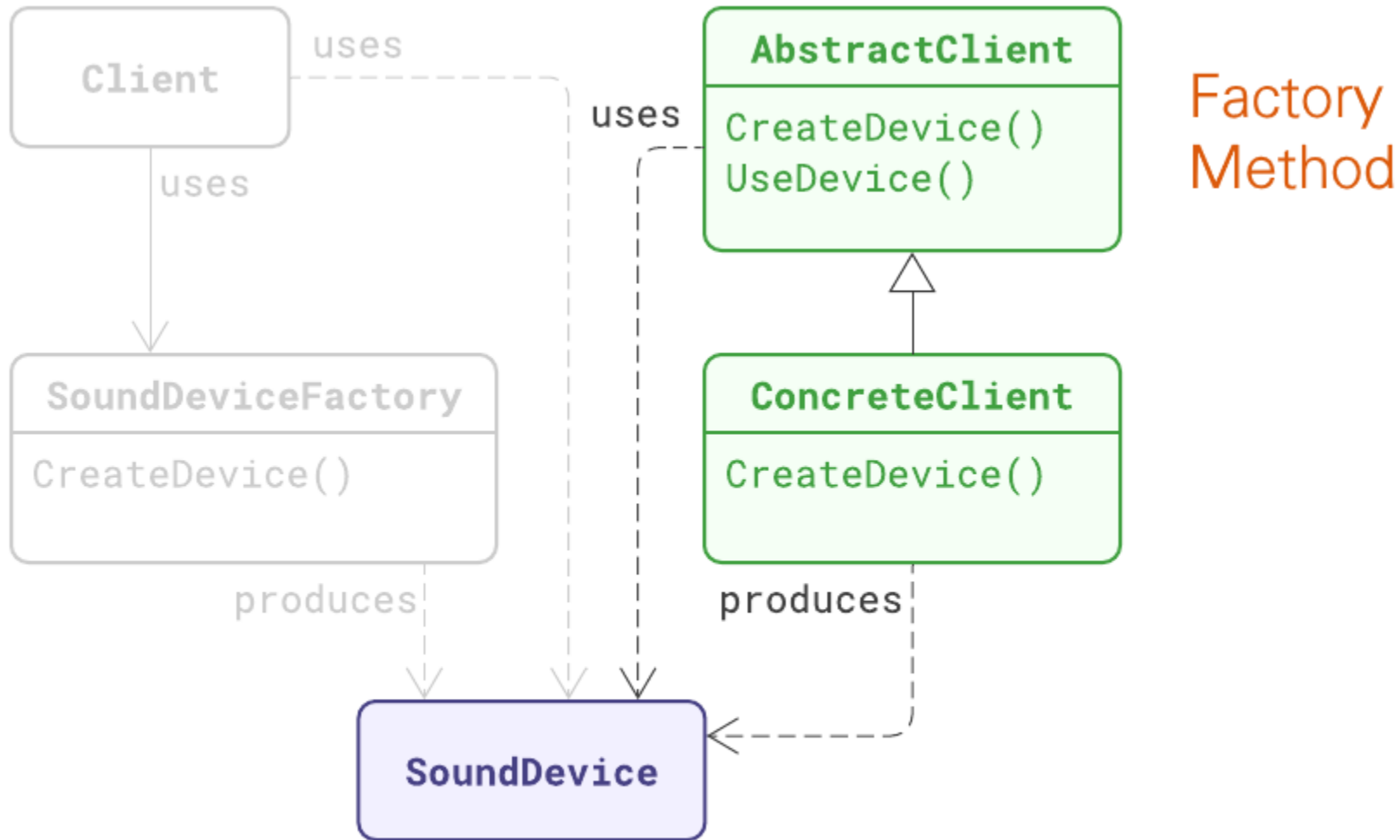
# Example: Sound Processing Library



# Example: Sound Processing Library

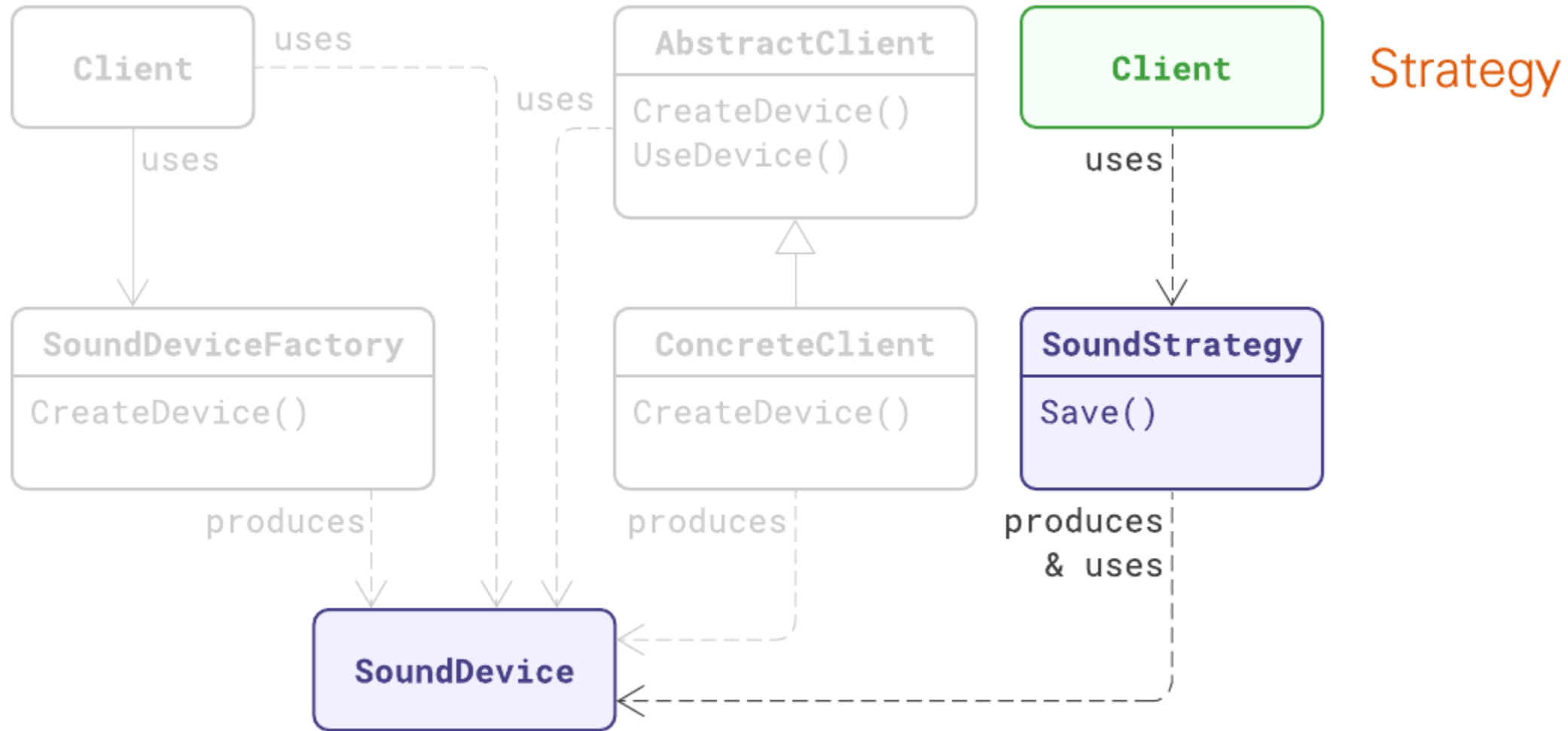


# Example: Sound Processing Library

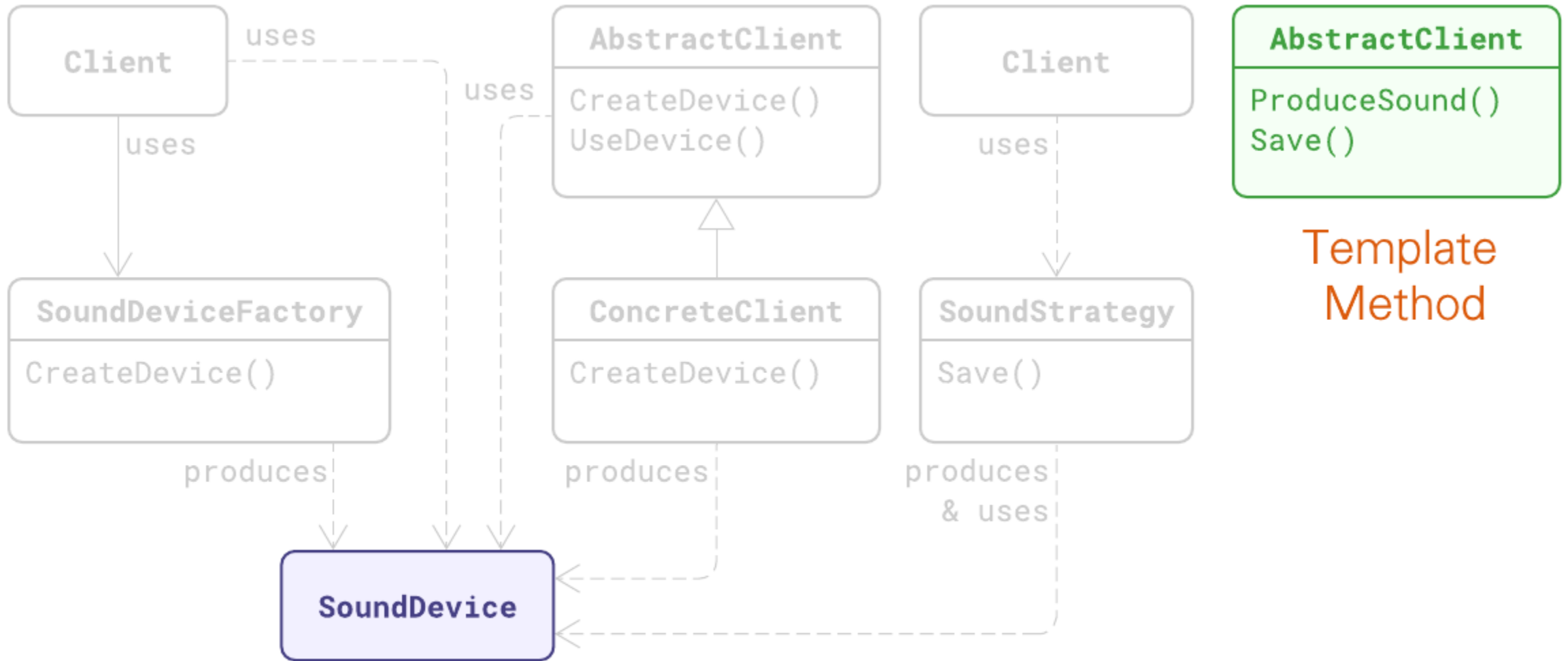




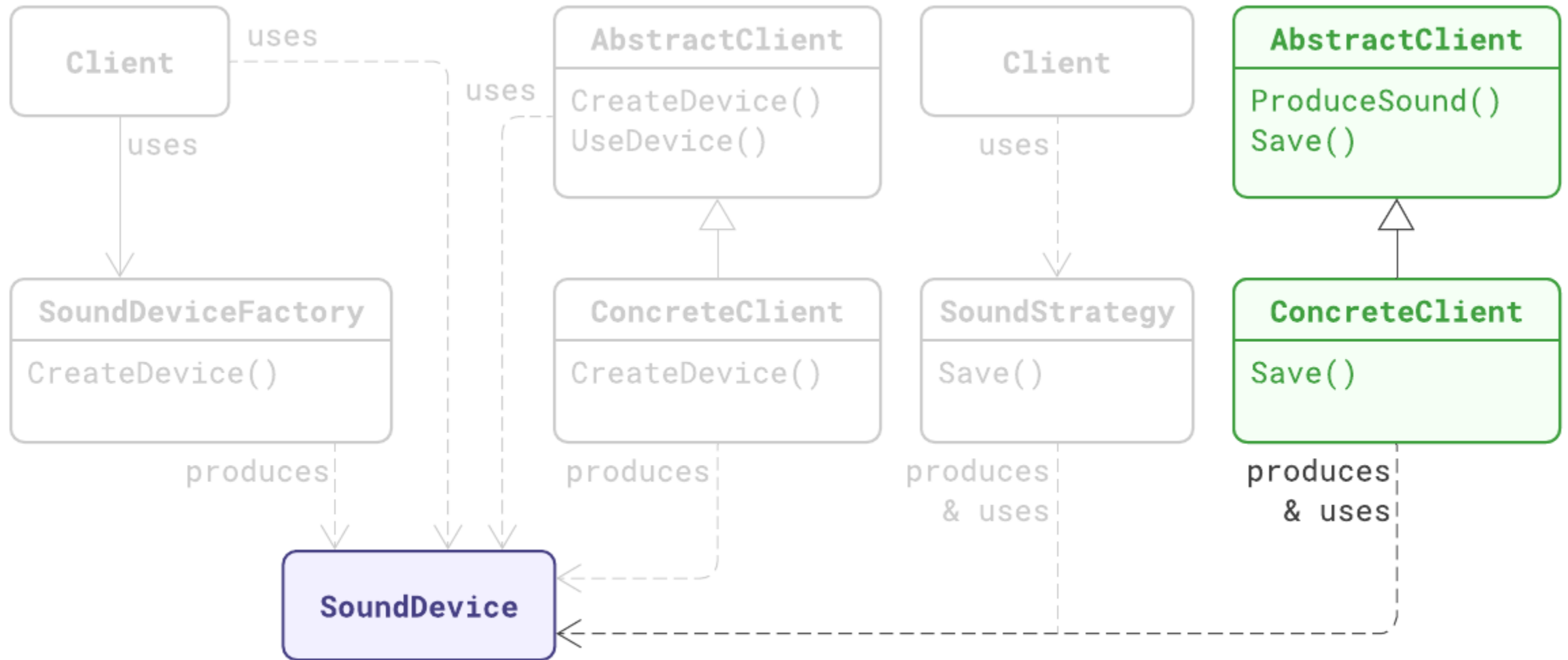
# Example: Sound Processing Library



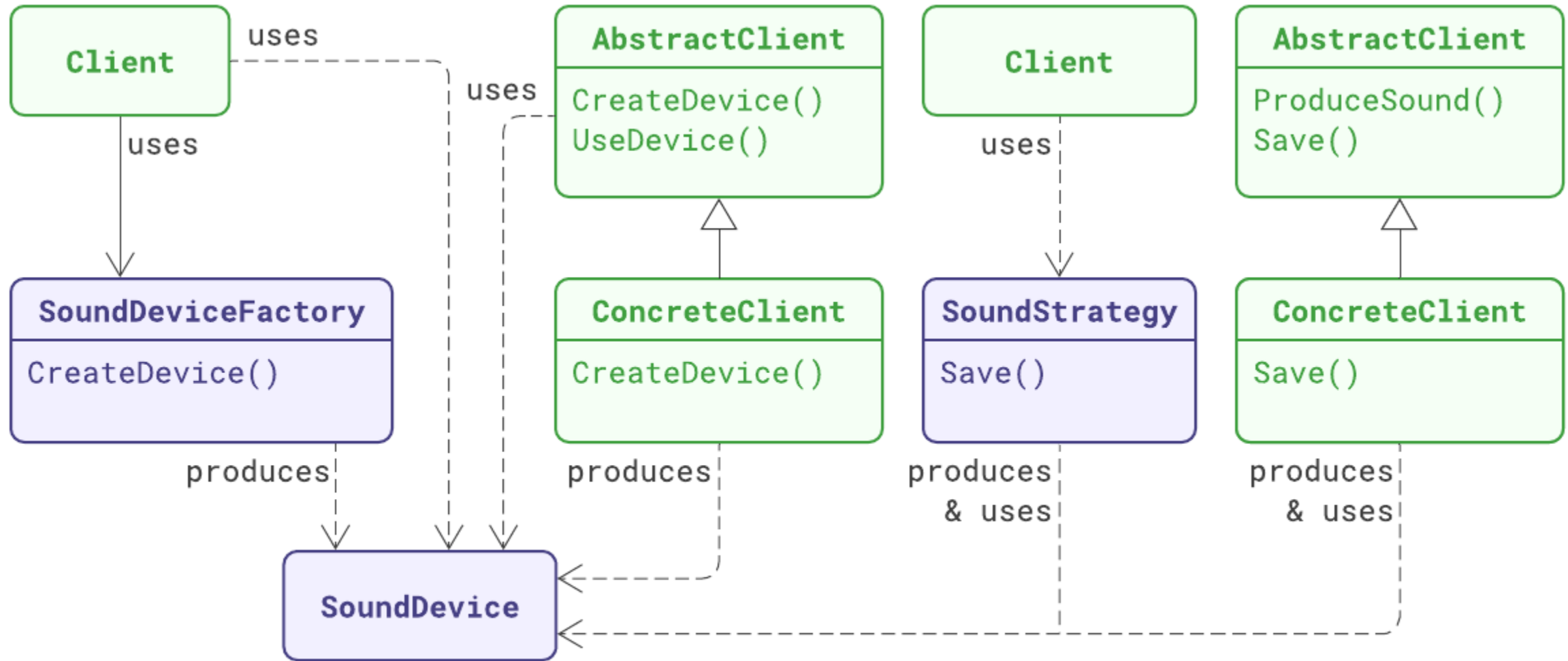
# Example: Sound Processing Library



# Example: Sound Processing Library



# Example: Sound Processing 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.

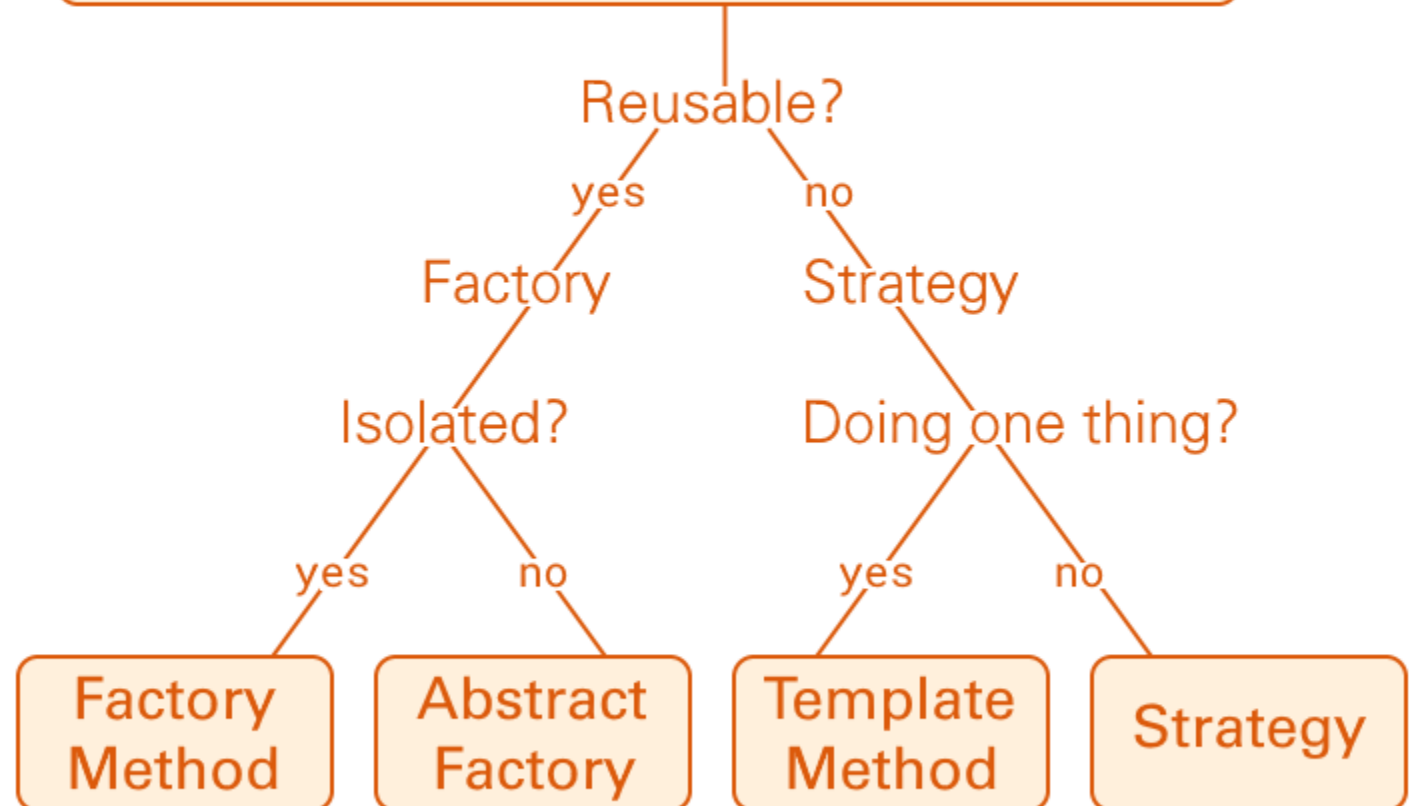
## User story #2

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.

## 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)
```

# More on Design Patterns and Refactoring

Design Patterns in C# Made Simple

<http://codinghelmet.com/go/design-patterns>



Refactoring to Design Patterns by Example

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

Tactical Design Patterns in .NET

- Managing Responsibilities
- Control Flow
- Creating Objects

<http://codinghelmet.com/go/pluralsight>



PLURALSIGHT

## Questions?