

Module 04: "Factory Method"



Agenda

- ▶ Introductory Example: Publications
- ▶ Challenges
- ▶ Implementing the Factory Method Pattern
- ▶ Pattern: Factory Method
- ▶ Overview of Factory Method Pattern
- ▶ Factory Method vs. Abstract Factory



Introductory Example: Publications

```
interface IPart
{
}
```

```
class Chapter : IPart
{
    public int Number { get; }
}
```

```
class Publication : IEnumerable<IPart>
{
    ...
    public IList<IPart> Parts { get; }
    public Publication( string title )
    { ... }
    public void Print() { ... }
}
```

```
Publication book = new Publication("Design Patterns in C#")
{
    new Foreword(),
    new Chapter( 1 ),
    new Index()
};
book.Print();
```

Challenges

- ▶ Clients should be shielded from internal object structure and creation
- ▶ The appropriate subclass should provide appropriate creational logic
- ▶ Additionally:
 - Most challenges from Builder
 - Some of the challenges from Abstract Factory

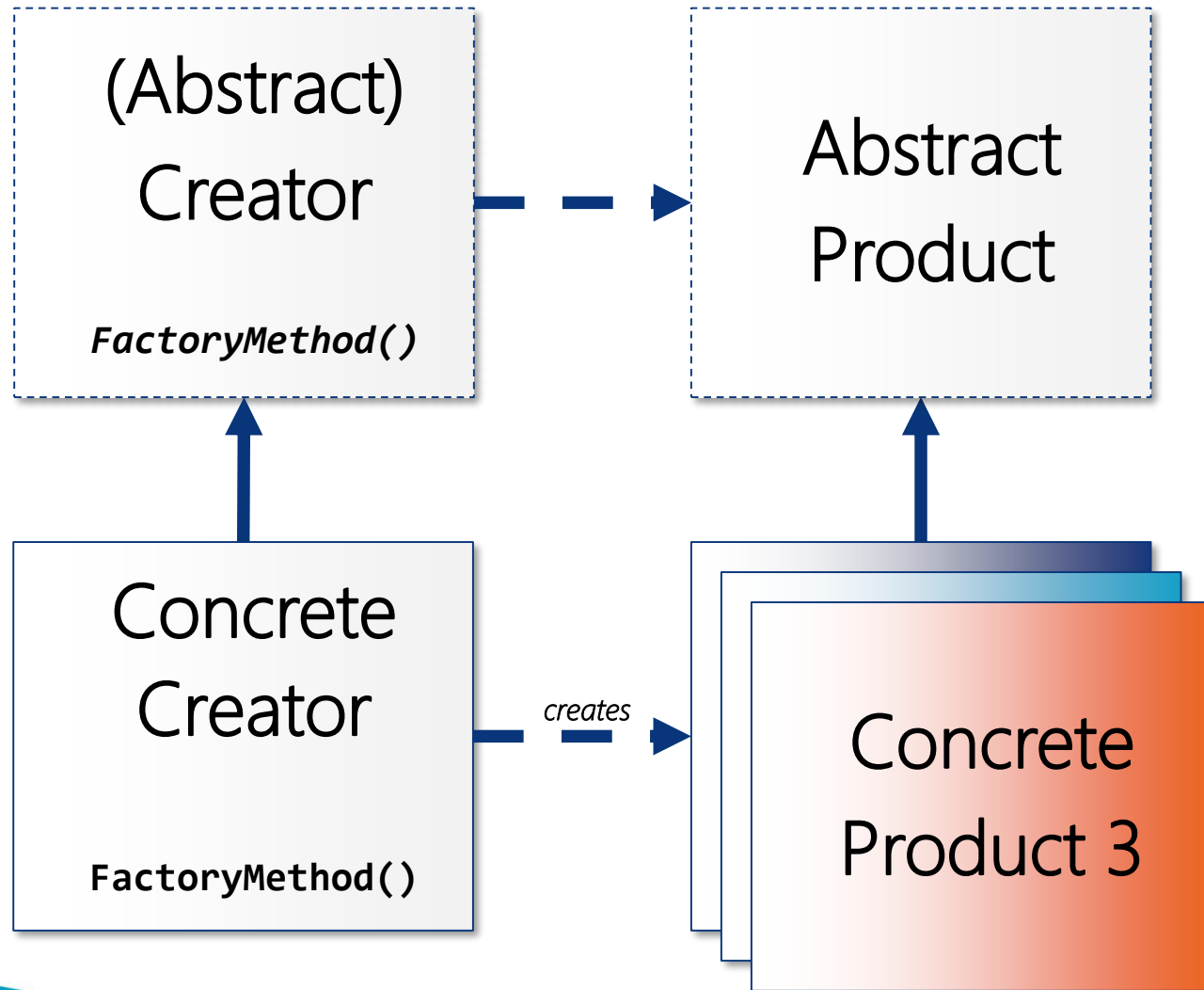


Pattern: Factory Method

- ▶ *Define an interface for creating an object, but let subclasses decide which class to instantiate. Factory Method lets a class defer instantiation to subclasses.*
- ▶ Purpose
 - Define a separate operation (factory method) for creating objects
 - Create objects by invoking factory method
- ▶ Origin: Gang of Four



Overview of Factory Method Pattern



Overview of Factory Method Pattern

- ▶ Product
 - Interface or abstract class capturing a generic product
- ▶ Concrete Product
 - Concrete class implementing the Product interface
- ▶ (Abstract) Creator
 - Usually abstract class providing abstract factory method
 - Could be concrete class and/or provide default implementation
 - May or may not call factory method
- ▶ Concrete Creators
 - Overrides factory method to provide specialized object creation for subclasses
- ▶ Note: Creators and Products might not be in one-to-one relationship



Factory Method vs. Abstract Factory

- ▶ Very often confused and used interchangeably
- ▶ Factory Method
 - A single method for object instantiation
 - Uses inheritance and relies on subclasses to instantiate
 - Calls own factory method
 - Handles a single product hierarchy
- ▶ Abstract Factory
 - A distinct other object with multiple factory methods
 - Client delegates object instantiation to this object
 - Handles families of product hierarchies





WINCUBATE

Jesper Gulmann Henriksen

PhD, MCT, MCSD, MCPD

Phone : +45 22 12 36 31

Email : jgh@wincubate.net

WWW : <http://www.wincubate.net>

Hasselvangel 243

8355 Solbjerg

Denmark