Factory Method

Linda Marshall and Vreda Pieterse

Department of Computer Science
University of Pretoria

2014

Overview

1. Identification

2. Structure

3. Participants

4. Related Patterns

5. Example
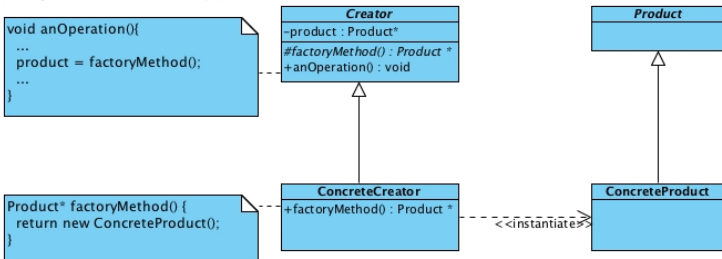
Name and Classification:

Factory Method (Class Creational)

Intent:

"Define an interface for creating an object, but let subclasses decide which class to instantiate. FM lets a class defer instantiation to subclasses" GoF(107)

Visual Paradigm for UML Standard Edition(University of Pretoria)

```
void anOperation(){
  ...
  product = factoryMethod();
  ...
}
```

```
Product* factoryMethod() {
  return new ConcreteProduct();
}
```

Creator

−product : Product*

#factoryMethod() : Product *
+anOperation() : void

Product

ConcreteCreator

+factoryMethod() : Product *

ConcreteProduct

<<instantiate>>

- a Creator creates product which the client uses
- Product is always created by Creator
- ConcreteCreators create specific concrete product
- Makes use of the Template Method design pattern
- Forces the creation of an object to occur in a common factory rather than

Product

- defines the product interface for the factory method to create

ConcreteProduct

- implements the interface for the product

Creator

- declares the factory method which returns a product object
- default factory method implementations may return a default concrete product

ConcreteCreator

- overrides the factory method to return an instance of the product

Related Patterns

- **Abstract Factory** (87): May be used to implement.

- **Template Method** (325): May be called by.

- **Prototype** (117): Used to initialise objects.