

DP- Factory Method

2013/10/16

NNNN

今日沒有GirlFriend

Create Object...

```
public class Test {  
  
    public static void main(String[] args) {  
  
        String type="馬尾";  
  
        Girlfriend gf;  
  
        if(type.equals("馬尾")){  
            gf = new 馬尾GirlFriend();  
        }else if(type.equals("長髮")){  
            gf = new 長髮GirlFriend();  
        }else {  
            gf = new 會用GithubGirlFriend();  
        }  
        gf.haveFun();  
  
    }  
  
}
```

Goal

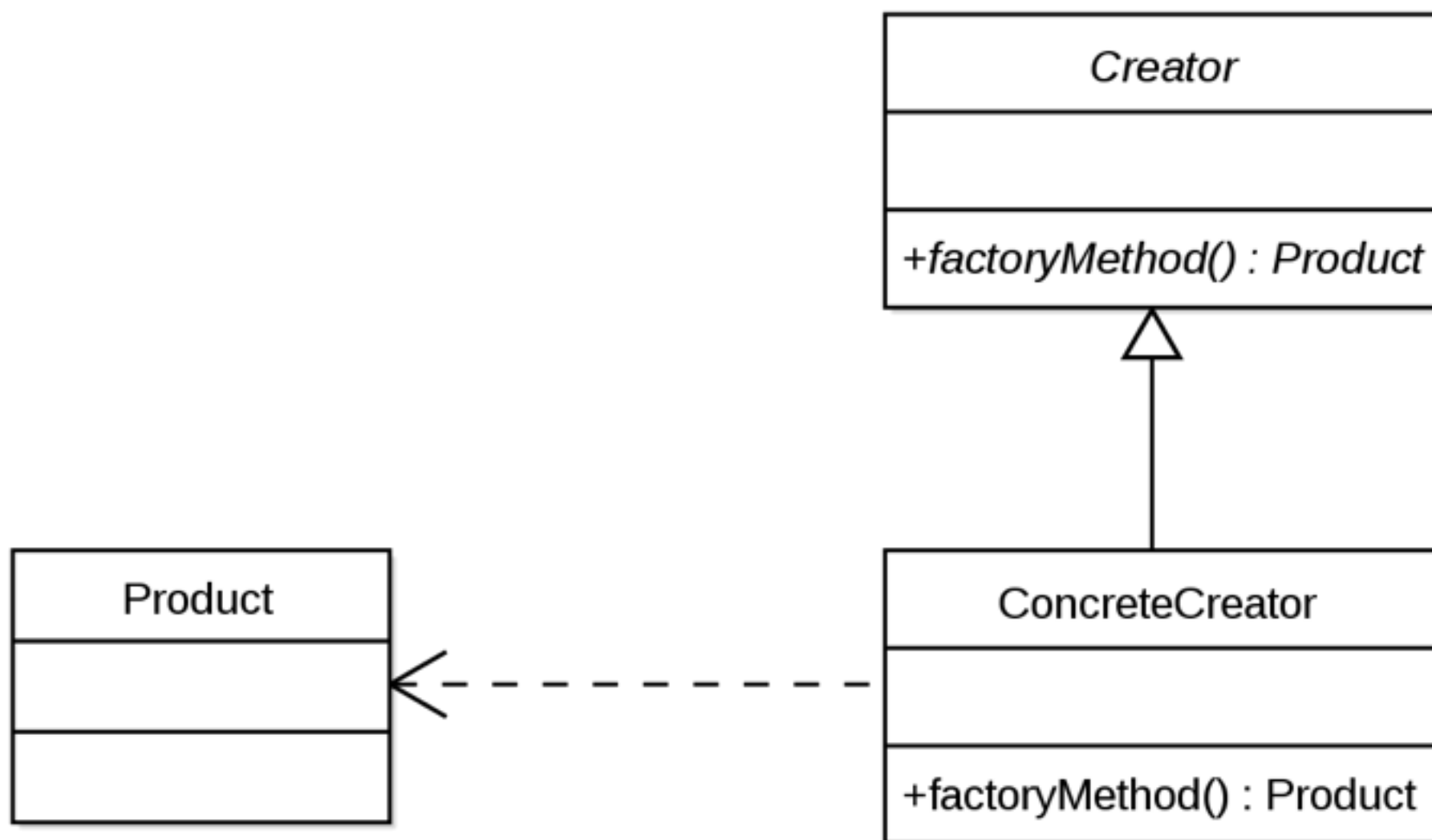
- Define an interface for **creating an object**, but let the classes that implement the interface decide which class to instantiate. The Factory method lets a class defer instantiation to subclasses.

Wiki

Factory Pattern

- define an interface for creating an object, lets subclasses decide which class to instantiate.

- Roles in Factory Method
 - Product
 - Concrete Product
 - Creator
 - ConcreteCreator



```
;□
```

```
3 {
```

```
id main(String[] args) {  
-generated method stub
```

```
f = GirlfriendFactory.;
```

```
;
```

- getDefault() : Girlfriend - GirlfriendFactory
- getGirlFriend(String type) : Girlfriend - GirlfriendFactory
- get會用GithubGirlFriend() : Girlfriend - GirlfriendFactory
- get長髮GirlFriend() : Girlfriend - GirlfriendFactory
- get馬尾GirlFriend() : Girlfriend - GirlfriendFactory
- class : Class<gf.GirlfriendFactory>

Press '⌘/' to show Template Proposals

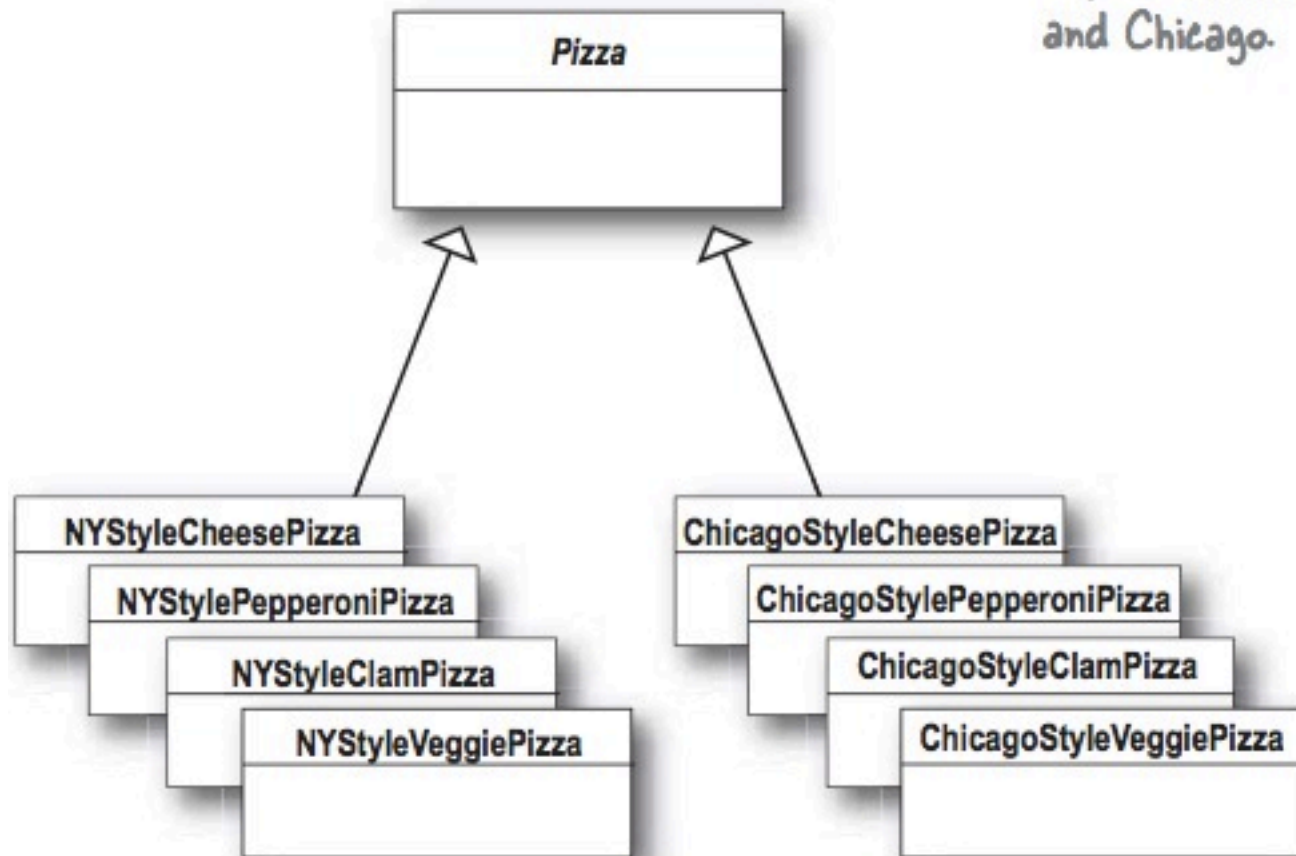
You can do lots of
things in **factory**

Encapsulation...

Abstract Factory

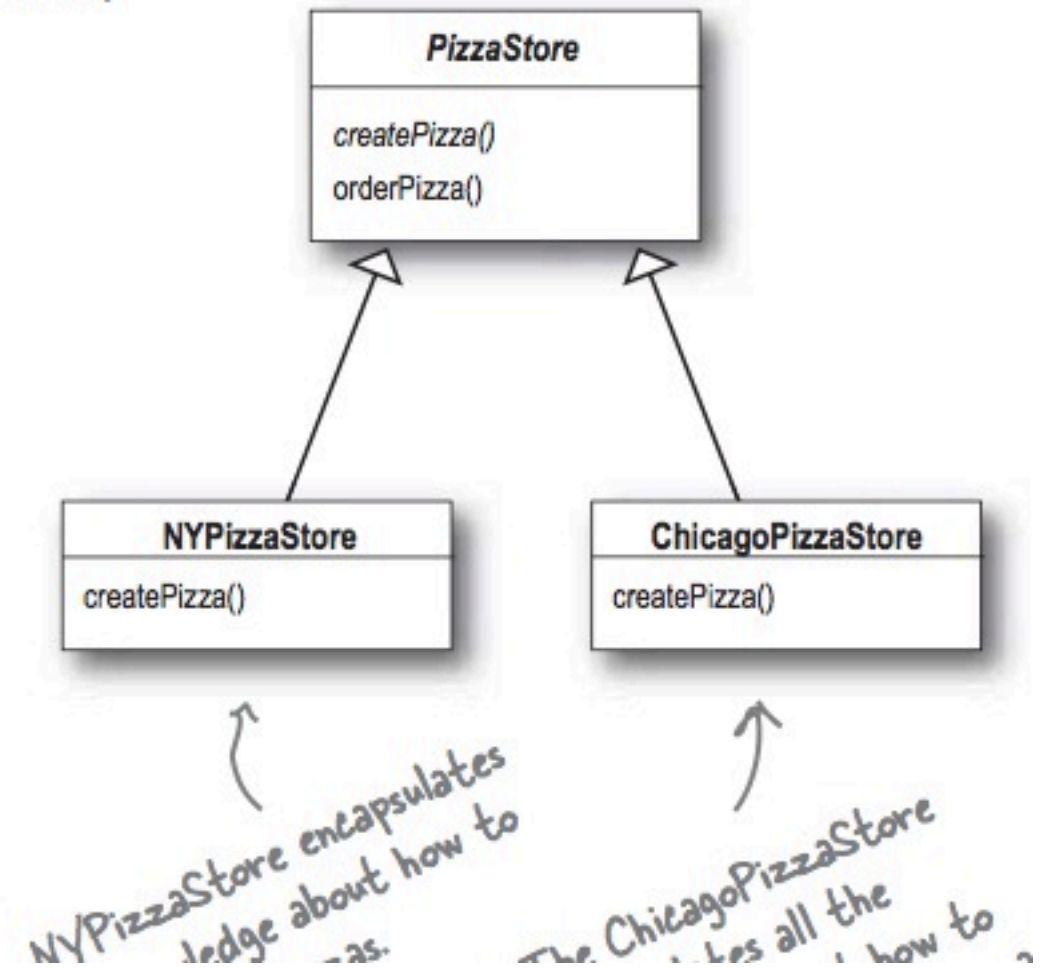
- It provides a way to encapsulate a group of individual factories that have a common theme without specifying their concrete classes.

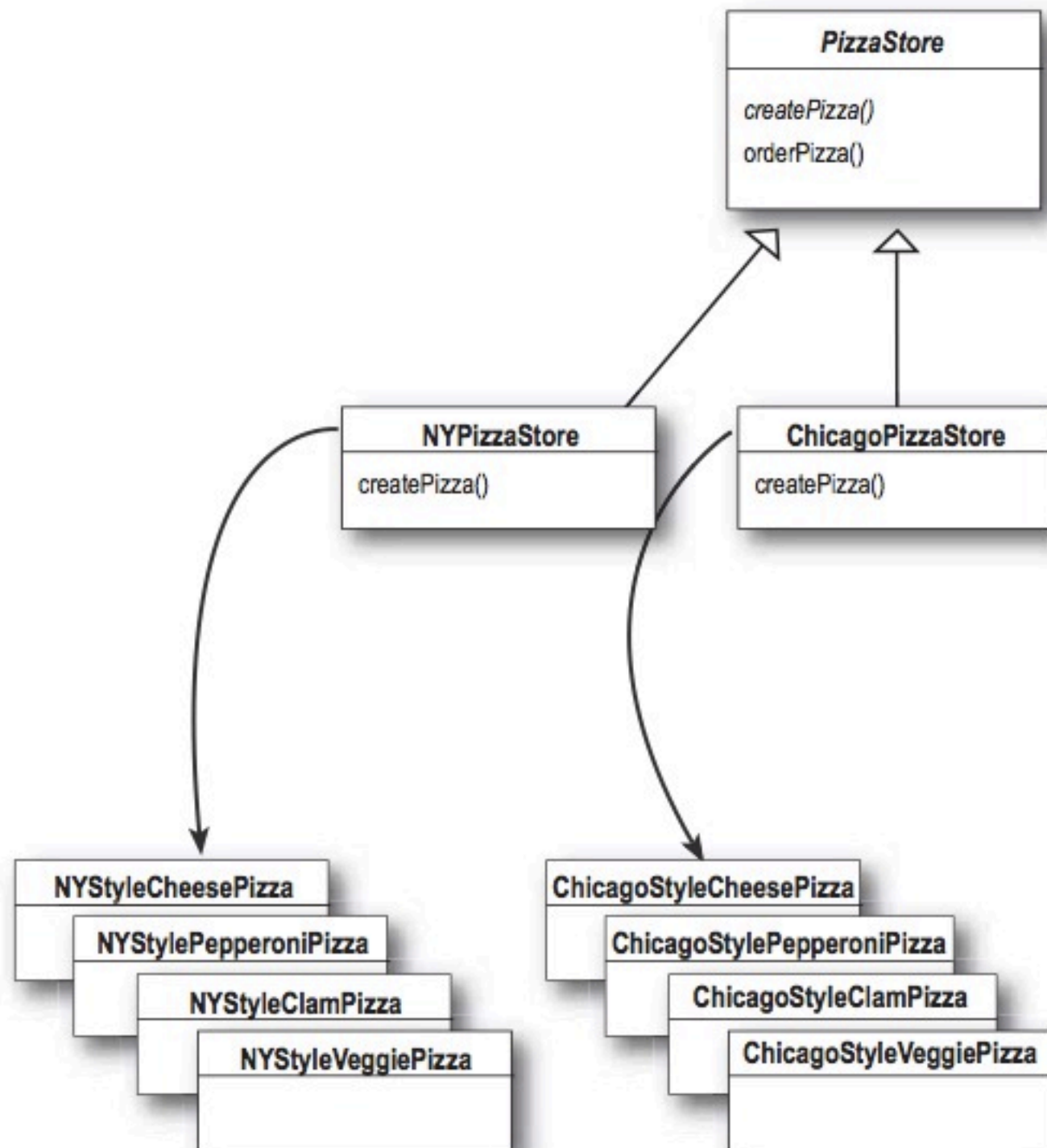
The Product classes



Notice how these class hierarchies are parallel: both have abstract classes that are extended by concrete classes, which know about specific implementations for NY and Chicago.

The Creator classes





DEMO

- Reference

- <http://blog.monkeypotion.net/gameprog/pattern/abstract-factory>
- <http://blog.monkeypotion.net/gameprog/pattern/factory-method>