**Hướng dẫn Java Design Pattern - Factory**

**Factory Pattern là gì?**

Factory Method là một mẫu khác thuộc nhóm các mẫu thiết kế phục vụ mục đích khởi tạo (kỳ trước chúng ta có mẫu Abstract Factory cũng nằm trong nhóm này). Điểm khác biệt cơ bản của 2 mẫu này đó là Abstract Factory dược dùng để tạo ra nhiều loại đối tượng thuộc cùng một nhóm, còn Factory Method nhằm mục đích thay đổi việc khởi tạo đổi tượng một cách linh hoạt.   
  
Mình có một ví dụ đơn giản cho trường hợp này: Nhà bạn chỉ có một con thú cưng (giống cái nhé), và nó đang mang bầu, tới khi nó đẻ con, bạn chạy sang nhà hàng xóm khoe, nhưng người hàng xóm này chỉ mới chuyển tới thôi, và họ không biết con thú cưng của bạn là thuộc loài nào, nên họ chỉ có thể biết được ngay lúc bạn giới thiệu rằng: “Con thú cưng nhà nó đã đẻ con rồi đó”. Và một bà cụ nhà kế bên, khi nghe thấy bạn khoe sẽ nghĩ ngay rằng: "Thế là con mèo lúc trước mình cho nó đã đẻ ra mấy chú mèo con rồi!".

Qua ví dụ trên chúng ta có thể thấy cách để mẫu Factory Method được áp dụng: Bạn có thể giới thiệu rằng con thú cưng của bạn vừa đẻ con, nhưng chỉ với bấy nhiêu thông tin, người ta chỉ có thể biết rằng vừa có con thú cưng con ra đời, nhưng nếu bạn cho người ta biết con thú cưng đó là con chó, người ta sẽ biết chắc chắn là nó vừa sinh ra con chó con, còn nếu đó là con mèo, thì nó không thể sinh ra con gì khác ngoài con mèo con được (Ví dụ này chúng ta xem tất cả điều kiện là lý tưởng, không xét những trường hợp kinh dị như chó đẻ ra mèo).

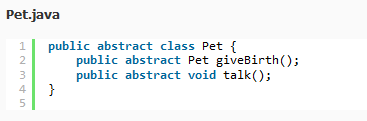
[Click image for larger version. 

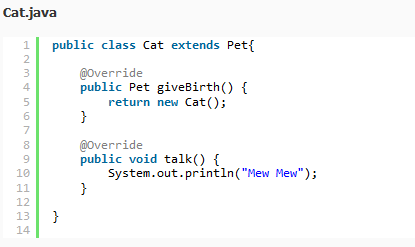
Name: factory-method.jpg 
Views: 212 
Size: 17.4 KB 
ID: 248](http://laptrinh.vn/attachment.php?attachmentid=248&d=1412693707)

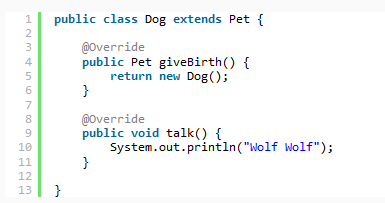
Code ví dụ cho mẫu này sẽ bao gồm các lớp Pet, Dog, Cat đóng vai trò như các Product và ConcreteProduct, bạn (Me) chính là Creator trong mẫu, nếu bạn có một con chó, bạn sẽ là MeAndADog, và sẽ là MeAndACat nếu bạn có một con mèo, 2 lớp này sẽ đại diện cho ConcreteCreator.

Để đơn giản, chúng ta chỉ ban cho thú cưng của mình những gì mà ta muốn: sinh con (giveBirth) và kêu(talk)

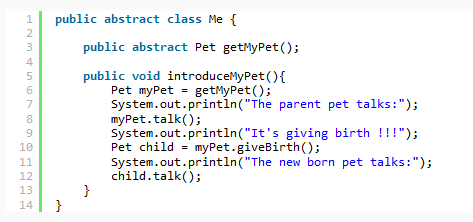
**Pet.java**

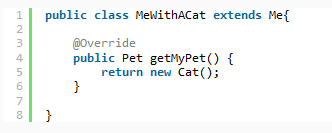
  
  
**Cat.java**

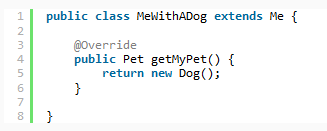
  
  
**Dog.java**



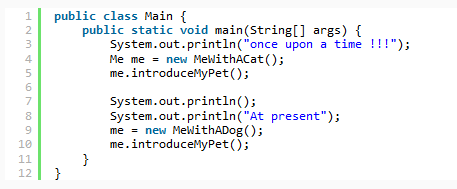
Bạn (Me và các lớp con) sẽ có thể chọn loại thú cưng của mình(getMyPet) và giới thiệu chúng (introduceMyPet)  
  
**Me.java**

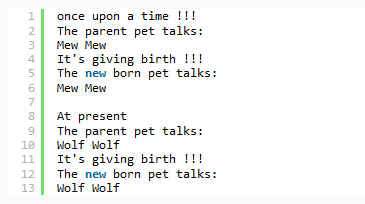
  
  
Ở đây bạn có thể thấy lớp cha này không xác định được con thú cưng của bạn thuộc loại nào, nó chỉ có thể định nghĩa cách bạn giới thiệu thú cưng của mình. Để biết được loại thú cưng, bạn cần định nghĩa với các lớp con của Me.  
  
**MeWithACat.java**

  
  
**MeWithADog.java**



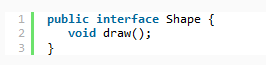
Và phần cuối cùng là xem chúng sẽ hoạt động với nhau như thế nào  
  
**Main.java**

  
  
**Output:**

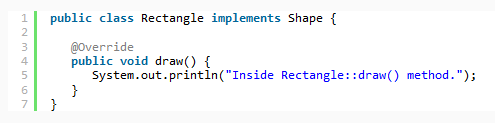
****

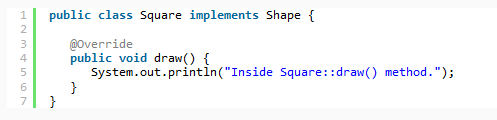
[Click image for larger version. 

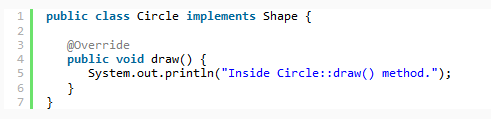
Name: factory_pattern_uml_diagram (1).jpg 
Views: 384 
Size: 18.7 KB 
ID: 249](http://laptrinh.vn/attachment.php?attachmentid=249&d=1412694084)  
  
**Step 1**  
  
Tạo lớp interface.  
  
**Shape.java**



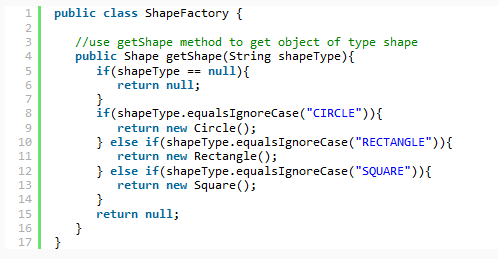
**Step 2**  
  
Tạo lớp implementing lớp interface vừa tạo.  
  
**Rectangle.java**

  
  
**Square.java**

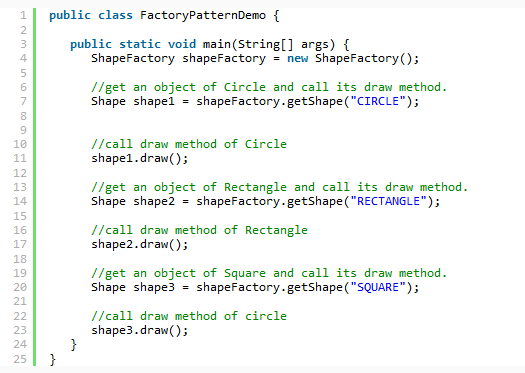
  
  
**Circle.java**



**Step 3**  
  
Tạo Factory class.  
  
**ShapeFactory.java**



**Step 4**  
  
**FactoryPatternDemo.java**

  
  
**Step 5**

