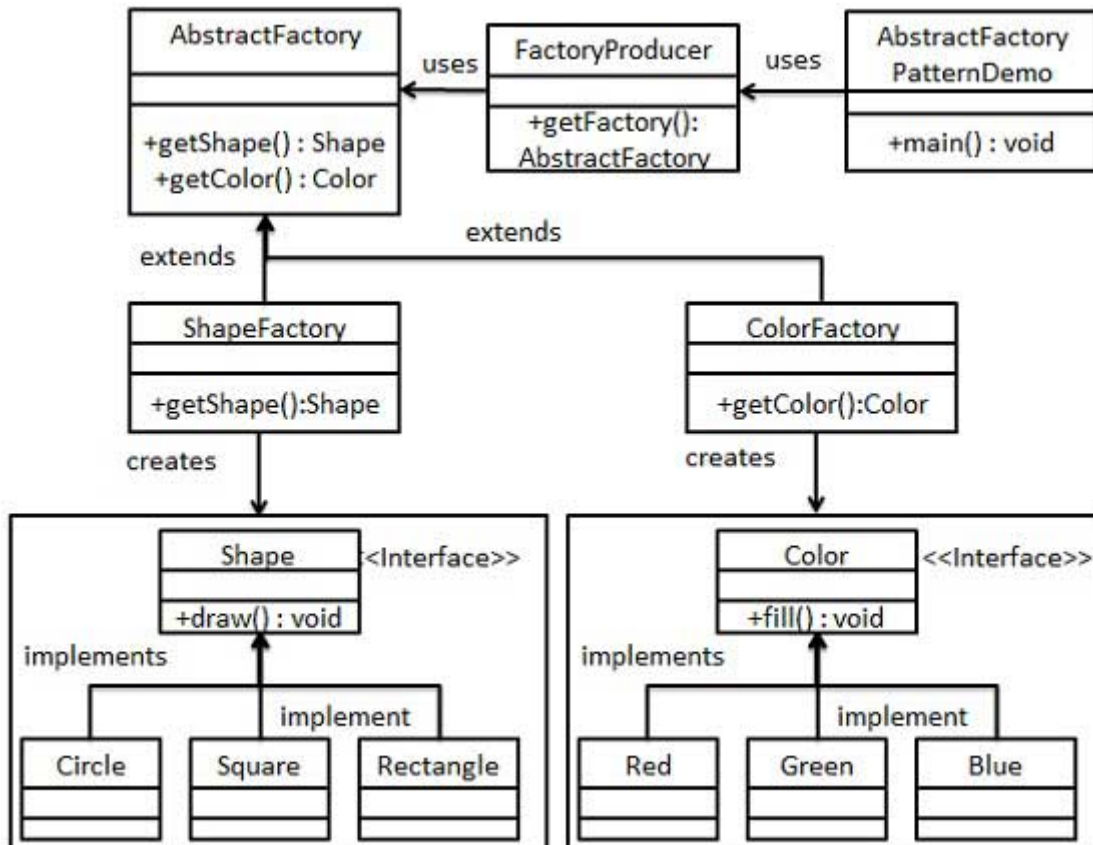


## AbstractFactory

Nous souhaitons mettre en place un design pattern « AbstractFactory » afin de construire des formes et des couleurs depuis une même « AbstractFactory ».

### Exercice 1

Mettre en place le schéma UML suivant :



Quelques informations supplémentaires :

- On utilisera des « String » pour définir si tel ou tel élément est un ShapeFactory ou un ColorFactory. Idem pour Circle, Square et Rectangle ainsi que Red, Green et Blue.
- Si une configuration n'est pas comprise l'objet retournera « null »
- Les implémentations des fonctions « fill() » et « draw() » doivent afficher des textes différents en fonction du type d'objet final.

L'architecture proposé devra permettre la création du point d'entré suivant :

```

// get shape factory
AbstractFactory shapeFactory =
FactoryProducer.getFactory(ShapeFactory.SHAPE);

// get an object of Shape Circle
Shape shape1 = shapeFactory.getShape(Circle.CIRCLE);
    
```

```
// call draw method of Shape Circle
shape1.draw();

// get an object of Shape Rectangle
Shape shape2 = shapeFactory.getShape(Rectangle.RECTANGLE);

// call draw method of Shape Rectangle
shape2.draw();

// get an object of Shape Square
Shape shape3 = shapeFactory.getShape(Square.SQUARE);

// call draw method of Shape Square
shape3.draw();

// get color factory
AbstractFactory colorFactory =
FactoryProducer.getFactory(ColorFactory.COLOR);

// get an object of Color Red
Color color1 = colorFactory.getColor(Red.RED);

// call fill method of Red
color1.fill();

// get an object of Color Green
Color color2 = colorFactory.getColor(Green.GREEN);

// call fill method of Green
color2.fill();

// get an object of Color Blue
Color color3 = colorFactory.getColor(Blue.BLUE);

// call fill method of Color Blue
color3.fill();

// get an object of Shape Circle
Shape myShape =
FactoryProducer.getFactory(ShapeFactory.SHAPE).getShape(Circle.CIRCLE);

// call draw method of Shape Circle
myShape.draw();
```

## Exercice 2

On rajoutera une interface « Item » implémenté par Circle, Square, Rectangle, Red, Green et Blue.  
L'interface « Item » sera défini comme suis :

```
public interface Item {
    String getName();
}
```

L' « AbstractFactory » devra permettre de créer directement un item en choisissant la bonne « Factory ». Ce qui nous permettra d'ajouter au point d'entrée de l'application :

```
//get directly an item with factory auto selecting
Item item = FactoryProducer.getItem(AllObjects.BLUE);
System.out.println("this is a " + item.getName());

item = FactoryProducer.getItem(AllObjects.GREEN);
System.out.println("this is a " + item.getName());

item = FactoryProducer.getItem(AllObjects.RED);
System.out.println("this is a " + item.getName());

item = FactoryProducer.getItem(AllObjects.SQUARE);
System.out.println("this is a " + item.getName());

item = FactoryProducer.getItem(AllObjects.RECTANGLE);
System.out.println("this is a " + item.getName());

item = FactoryProducer.getItem(AllObjects.CIRCLE);
System.out.println("this is a " + item.getName());
```

## Exercice 3

On souhaite maintenant créer 2 nouvelles classes :

- « CombinedItem » qui possède les attributs :
  - Shape shape
  - Color color
- « DrawItem » qui possède :
  - les attributs :
    - String name
    - List<CombinedItem> items
  - la fonction :
    - print() : qui pour chaque « CombinedItem » affichera la « Shape » et la « Color »

Le tout nous permettant l'ajout dans notre point d'entrée de :

```
////////////////////////////////////
CombinedItem cItem1 = new CombinedItem();
cItem1.setColor(colorFactory.getColor(Blue.BLUE));
cItem1.setShape(shapeFactory.getShape(Square.SQUARE));
```

```
CombinedItem cItem2 = new CombinedItem();
cItem2.setColor(colorFactory.getColor(Red.RED));
cItem2.setShape(shapeFactory.getShape(Square.SQUARE));

CombinedItem cItem3 = new CombinedItem();
cItem3.setColor(colorFactory.getColor(Blue.BLUE));
cItem3.setShape(shapeFactory.getShape(Circle.CIRCLE));

CombinedItem cItem4 = new CombinedItem();
cItem4.setColor(colorFactory.getColor(Green.GREEN));
cItem4.setShape(shapeFactory.getShape(Rectangle.RECTANGLE));

DrawItem dItem = new DrawItem();
dItem.setName("mon dessin");
dItem.getItems().add(cItem1);
dItem.getItems().add(cItem2);
dItem.getItems().add(cItem3);
dItem.getItems().add(cItem4);

dItem.print();
```