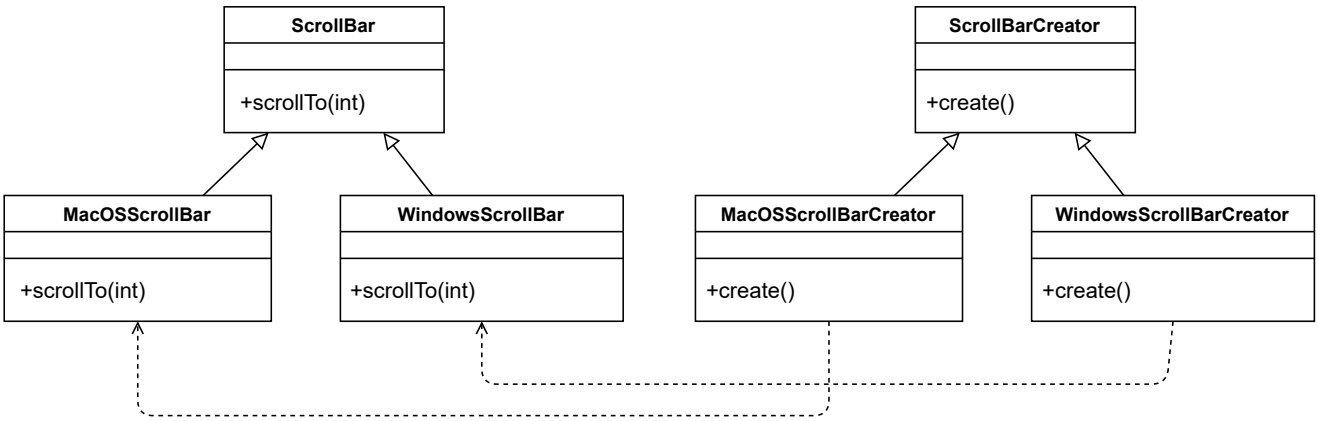
TP1 - Factory

#DesignPattern

Méthode de Fabrique

La méthode de fabrique est un pattern de création qui permet de palier au problème de non-fléxibilité d'objets qui se ressemblent, font la même chose mais dont le design change légèrement. Le pattern propose de créer une classe qu'on appelle une Fabrique (Factory method) qui va abstraire la création d'objet et laisser le détail à ses sous classes. Ici par exemple on souhaite créer une Scrollbar, qui va changer de design selon si c'est une barre MacOS ou une barre Windows. On va donc créer une méthode de fabrique.



La méthode de fabrique ScrollbarCreator permet donc d'universaliser la création d'une scrollbar on peut donc faire :

```
public abstract Class ScrollBarCreator {
    public abstract ScrollBar create();
}

/*************************************************************/

public Class MacOSScrollBarCreator extends ScrollBarCreator {
    @Override
    public MacOSScrollBarCreator create() {
    return new MacOSScrollBar();
    }
}
```

```
public Class WindowsScrollBarCreator extends ScrollBarCreator {
    @Override
    public WindowsScrollBarCreator create() {
    return new WindowsScrollBar();
    }
}

/*************************************************************/

public Class MacOSScrollBar extends ScrollBar{

    public MacOSScrollBar() {

    }

    public void scrollTo(int distance) {
    // ...
    }
}

// pareil pour la windows

/*************************************************************/

// et lorsque l'on créé notre ScrollBar, dans la classe main par
exemple :

public Class Main {

    public static main() {
    ScrollBarCreator macOSsbc = MacOSScrollBarCreator();
        ScrollBar myScrollbar = macOSsbc.create();
    }
}
```

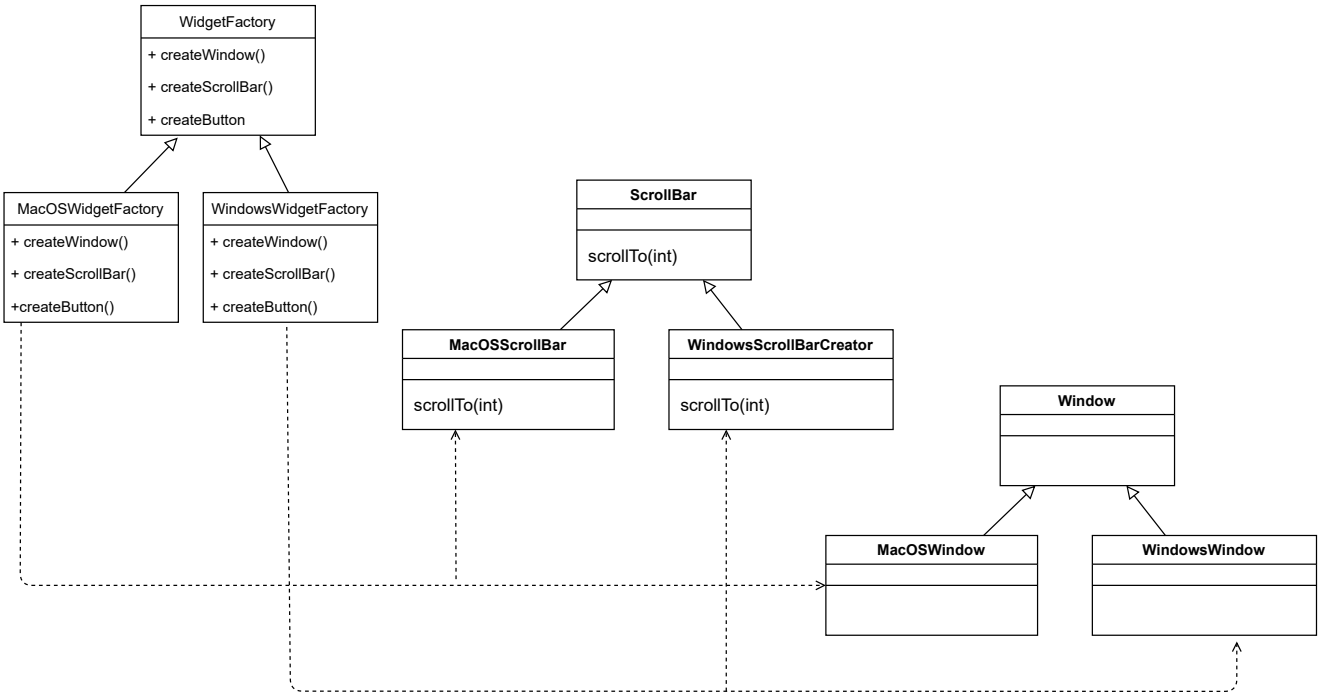
- ScrollBar défini la classe abstraite des objets créés par la méthode de fabrique (ici create()).
- MacOSScrollBar et WindowsScrollBar héritent (ou implémentent en soi, soit on fait une abstract class ScrollBar soit on fait une interface) de l'abstract

class ScrollBar avec quelque chose de concretScrollBarCreator déclare la méthode fabrique qui retourne un object de type ScrollBar
- MacOSScrollBarCreator redéfini la méthode de fabrique pour retourner l'instance d'un MacOSScrollBar

Méthode de fabrique abstraite

On peut utiliser une fabrique abstraite afin d'implémenter différentes factory ou groupes d'objets sans que connaître la classe concrète à utiliser.

Par exemple, on peut créer une fabrique abstraite pour construire une interface pour MacOS ou Windows qui va être implementé (ou qui sera hérité) par deux fabriques concrètes (comme vues au dessus).




```
public abstract Class MyAbstractWidgetFactory {

    public Window createWindow();
    public ScrollBar createScrollBar();
    public Button createButton();
}

/*************************************************************************/


public Class MacOSWidgetFactory extends MyAbstractWidgetFactory {
    public Window createWindow() {
        return new MacOSWindow();
```

```
    }
    public ScrollBar createScrollBar() {
        return new MacOSScrollBar();
    }
 // ...
}

// pareil pour windows en soit
// on réutilise les fonction de la partie précédente...

/*********************************************************************/


// et mtn dans le client :

public Class Main {

    public static main() {
    WidgetFactory macOSWf = MacOSWidgetFactory();
        Window myWindow = macOSWf.createWindow();
        ScrollBar myScrollbar = macOSWf.createScrollBar();
    }
}
```

- WidgetFactory défini la classe pour les opérations qui créent des objets de produit abstraits (Window, Scrollbar etc..)
- MacOSWidgetFactory et WindowsWidgetFactory implémentes les fonctions pour créer des objets de produits qui sont en rapport avec la fabrique, des objets concrets (MacOSWindow, MacOSScrollBar pour la MacOSWidgetFactory, etc...)
- Le reste du schéma reprend le format la factory standard. Produit standard → produit concret
- Le client (classe Main) ne déclare que des éléments abstrait et les instancie à l'aide de fabriques.