



# Mini-projet 2

11



## Présentation de mini-projet 2

<b>Nom de l'atelier</b>	<b>Mini-projet2: Gestion des publications</b>
<b>Objectifs de l'atelier</b>	A la fin de ce mini-projet, vous devez être capable de : <ul style="list-style-type: none"><li>comprendre les concepts orientés objets : classe, objet, composition, agrégation, héritage, classes abstraites, interfaces, gestion de exceptions.</li><li>appliquer ces concepts dans la réalisation d'une petite application</li></ul>
<b>Mots clés</b>	classe, objet, membres statiques, composition, agrégation, héritage, classe abstraite, interface, gestion des exceptions
<b>Durée</b>	libre mais ne dépasse pas un mois
<b>Outils et environnements</b>	Eclipse ou équivalent

## Mini-projet2 : Gestion des publications



### Attention

On vous donne le **choix** entre programmer une **version simplifiée** ou plus **complète** de ce mini-projet.

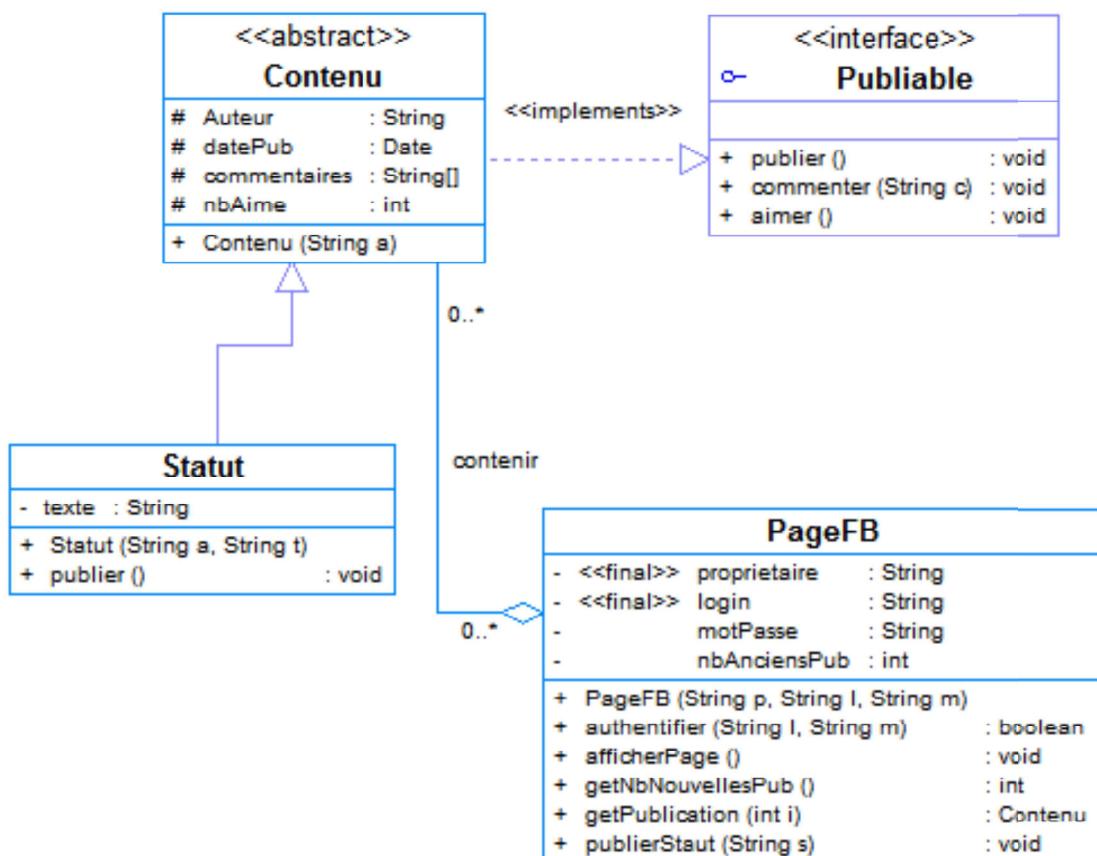
## Version simplifiée du mini-projet

### Énoncé

On se propose d'implémenter un système de gestion des publications fonctionnant selon les principes suivants:

- Le système permet d'afficher les **pages facebook** après authentification, de publier des contenus, de commenter et d'apprécier (aimer) des publications.
- Une page est composée de plusieurs **contenus**. On se limite à traiter les contenus textuels représentant des statuts.

Pour programmer ce système, on se base sur le diagramme de classes suivant:





## Méthode

### Partie 1

1. Écrivez le code Java de l'interface **Publiable**.
2. Écrivez le code Java de la classe abstraite **Contenu** sachant que :

Un **contenu** est caractérisé par les attributs suivants :

- **Auteur** et **datePub** représentent respectivement l'auteur du contenu et sa date de publication.
- Utilisez la classe **Date** se trouvant dans le package java.util pour définir l'attribut datePub.
- **commentaires** qui est une liste de commentaires. Un commentaire est de type chaîne. Utilisez la classe ArrayList se trouvant dans le package java.util pour définir cet attribut.
- **nbAime** qui est le nombre de personnes qui ont aimé le contenu

Un **contenu** englobe les méthodes suivantes :

- Un **constructeur** qui initialise l'auteur à l'aide d'un paramètre et la date de publication avec la date système. Pour avoir la date système, il suffit d'utiliser le constructeur sans paramètre de la classe **Date**.
- La méthode **Commenter** (String c) ajoute le commentaire c à la liste des commentaires.
- La méthode **aimer** () incrémente le nombre de ceux qui ont aimé le contenu.

Écrivez le code Java de la classe **Statut** qui est une classe dérivée de la classe Contenu sachant que cette classe représente un contenu textuel et englobe:

- Un attribut nommé texte représentant le texte du statut.
- Un constructeur initialisant les attributs Auteur et texte à l'aide des paramètres.

La méthode **publier** () permettant d'afficher un statut comme suit :

```
*****
Texte du statut
Publié par Auteur le date de publication
Nombre de commentaires commentaires nombre de j'aime j'aime
    Commentaire 1
    Commentaire 2
    etc ..
*****
```

```
*****
Vive la Tunisie
Publié par Mohamed Khelifa le 10/12/2017
2 commentaires 30 j'aime
    Bravo
    Merci à l'équipe
*****
```

P

### Partie 2

Écrivez le code Java de la classe **PageFB** sachant que :

- Une **page** est caractérisée par les attributs **propriétaire**, **login** et **motPasse**.
- Une page englobe un **ensemble de contenus** à afficher nommé publications.

- Utilisez ArrayList pour définir l'attribut **publications**.
- Pour calculer le **nombre des nouvelles publications**, la page utilise un attribut nommé **nbAncienPub** qui est mis à jour après chaque affichage de la page.
- Le **constructeur** d'une page initialise le propriétaire, le login et le mot de passe..
- La méthode **authentifier** (String l, String m), permet de vérifier si le login et le mot de passe donnés en paramètres sont corrects.
- La méthode **getNbNouvellesPub()** permet de retourner le nombre des nouvelles publications. Ce nombre est calculé en faisant la différence entre le nombre actuel des publications et le nombre de publications anciennes (**nbAncienPub**).
- La méthode **afficherPage()** permet d'afficher une entête de la page suivant ce modèle :

```
*****
Page de Mohammed Khelifa
Le nombre de nouvelles publications : 3
*****
```

Par la suite, la méthode publie la liste des publications dans l'ordre décroissant de leur ajout (les derniers contenus ajoutés sont affichés en premier).

Cet affichage se fait après une **authentification réussie** où le login et le mot de passe sont à saisir à partir du clavier. Autrement la méthode lève et propage une exception nommée **LoginPwdIncorrect** à définir.

- La méthode **publierStatut** (String s) ajoute un statut aux anciennes publications.
- La méthode **getPublication** (int i) retourne le contenu numéro i de la page.

### **Partie 3**

Écrivez le code Java d'une classe de test nommée **SystèmeFB** qui contient une seule méthode main permettant de :

- Déclarer 2 pages à partir de ce tableau.
- La page P1 publie le statut suivant : « Bonjour les amis »
- La page P2 publie le statut suivant : « On va gagner !!!! »
- Afficher la page P1 sachant qu'on doit traiter l'exception LoginPwdIncorrect par un simple affichage du message de l'exception.
- Sélectionner la première publication de la page P1, l'aimer et la commenter avec le texte « Merci ».
- Afficher la page P2 puis réafficher la page P1.

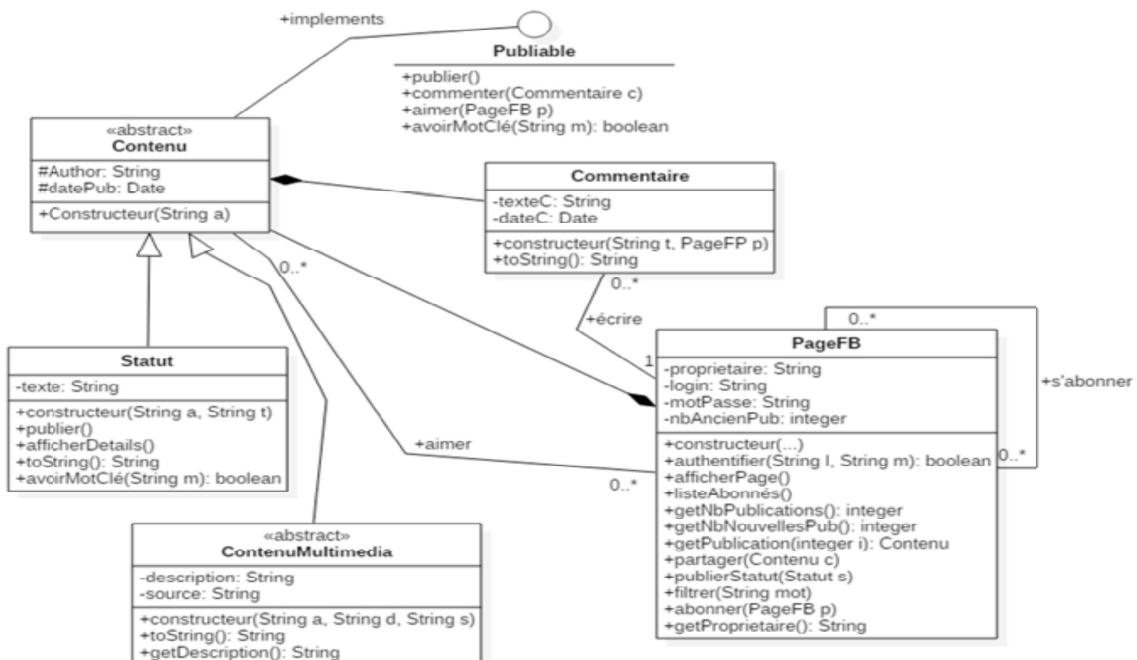
# Version complète du mini-projet

## Énoncé

On se propose d'implémenter un système de gestion des publications fonctionnant selon les principes suivants:

- Le système permet d'**afficher** les pages après authentification, de **s'abonner** à des pages, de **partager** et **publier** des contenus, de **commenter** et d'**apprécier** les publications et **d'effectuer** des recherches thématiques en utilisant des mots clés.
- Une **page** est composée de plusieurs **contenus** qui peuvent être des statuts (contenu textuel) ou multimédia.

Pour programmer ce système, on se base sur le diagramme de classes suivant:



## Méthode

### Partie 1

1. Écrivez le code Java de l'interface **Publifiable**.
2. Écrivez le code Java de la classe abstraite **Contenu** sachant que :

Un **contenu** est caractérisé par les attributs suivants :

- Une liste de commentaires (utilisez le type ArrayList) appelée **comments**.
  - Une liste des pages qui ont apprécié le contenu (utilisez le type ArrayList) appelée **likers**.
  - **Author** et **datePub** représentent respectivement l'auteur du contenu, sa date de publication.
- Utilisez la classe **Date** se trouvant dans le package java.util pour définir l'attribut **datePub**.

Un **contenu** englobe les méthodes suivantes :

- Un **constructeur** qui initialise l'auteur à l'aide d'un paramètre et la date de publication avec la date système. Pour avoir la date système, il suffit d'utiliser le constructeur sans paramètre de la classe Date.
- La méthode **publier()** et **avoirMotClé( String m)** sont abstraites au niveau de la classe Contenu.
- La méthode **Commenter** (Commentaire c) permet d'ajouter le commentaire c à la liste des commentaires comments.

La méthode **aimer** (PageFB p) permet d'ajouter cette page à la liste de ceux qui ont aimé le contenu (l'attribut likers)

**3.** Écrivez le code Java de la classe **Commentaire** qui englobe les membres suivants :

- **texteC** et **dateC** sont deux attributs représentant respectivement le texte et la date d'un commentaire.
- **auteurC** est un attribut de type PageFB représentant la page qui a écrit le commentaire.
- Un constructeur à 2 paramètres permettant de créer un commentaire à l'aide de son texte et la page qui a commenté. La date est par défaut la date système.
- Une méthode **toString()** qui retourne les données d'un commentaire (texte, date et auteur) sous forme d'une chaîne.

**4.** Écrivez le code Java de la classe abstraite **ContenuMultimedia** qui est une classe dérivée de la classe Contenu sachant que cette classe englobe:

- Deux attributs description et source de type chaînes de caractères.
- Un **constructeur** initialisant les attributs auteur, description et source à l'aide des paramètres.
- Une méthode **getDescription ()** qui retourne la description du contenu.
- La méthode **avoirMotClé(String m)** permet de rechercher le mot clé m dans la description de l'objet multimédia.
- Vous pouvez utiliser la méthode Contains de la classe String pour effectuer la recherche.

**5.** Écrivez le code Java de la classe **Statut** qui est une classe dérivée de la classe Contenu sachant que cette classe représente un contenu textuel et englobe:

- Un attribut nommé **texte** représentant le texte du statut.
- Un **constructeur** initialisant les attributs **Auteur** et **texte** à l'aide des
- La méthode **publier ()** permettant d'afficher un statut comme suit :

\*\*\*\*\*  
**Texte du statut**  
*Publié par Auteur le date de publication*  
*Nombre de commentaires commentaires nombre de j'aime j'aime*  
*Commentaire 1*  
*Commentaire 2*  
*etc ..*  
\*\*\*\*\*

\*\*\*\*\*  
**Vive la Tunisie**  
*Publié par Mohamed Khlifa le 10/12/2017*  
*2 commentaires 30 j'aime*  
*Bravo*  
*Merci à l'équipe*  
\*\*\*\*\*

- La méthode **avoirMotClé**(String m) permet de rechercher le mot clé m dans le texte du statut. Vous pouvez utiliser la méthode Contains de la classe String pour effectuer la recherche.
- La méthode **afficherDetails()** permettant de publier le statut comme décrit plus haut et affiche en plus la liste des commentaires et la liste de tous ceux qui ont aimé la page.

## **Partie 2**

Écrivez le code Java de la classe **PageFB** sachant que :

- Une page est caractérisée par les attributs propriétaire, **login** et **motPasse** qui permettent d'identifier le propriétaire de la page. Une page englobe un ensemble de contenu à afficher nommé **publications**. Utilisez le type ArrayList.
- Une page contient une liste des pages abonnées (ArrayList) nommée **IstAbonnés**.
- Pour calculer le nombre des nouvelles publications, la page utilise un attribut nommé **nbAncienPub** qui est mis à jour après chaque affichage de la page.
- Le **constructeur** d'une page permet d'initialiser le propriétaire, le login et le mot de passe.
- La méthode **authentifier** (String l, String m), permettant de vérifier si le login et le mot de passe donnés en paramètres sont corrects.
- La méthode **afficherPage()** permettant de publier les publications dans l'ordre décroissant de l'ajout (les derniers contenus ajoutés sont affichés au début) et ceci après authentification. Le login et le mot de passe sont saisis à partir du clavier. Cette méthode lève et propage une exception nommée **LoginPwdIncorrect** à définir. L'entête de la page suit ce modèle, elle est ensuite suivie de la liste des publications :

\*\*\*\*\*  
*Page de Mohamed Khlifa*  
*Nombre d'abonnés : 2*  
*3 nouvelles publications*  
\*\*\*\*\*

- La méthode **abonner** (PageFB p) permet de s'abonner à la page p. s'abonner à la page consiste à ajouter la page courante à la liste des abonnés de la page p.

- La méthode **partager** (Contenu c) permettant d'ajouter le contenu c à la liste des publications de la page ainsi qu'aux publications de tous les abonnés.
- La méthode **publierStatut** (Statut s) permettant de partager le statut s à tous les abonnés de la page. Si la page ne contient aucun abonné, la méthode lève, propage et traite une exception nommée **ZeroAbonnéException** à définir. Le traitement consiste à afficher le message correspondant à cette exception.
- La méthode **listeAmis** () permettant d'afficher les abonnés de la page.
- La méthode **getPublication** (integer i) permettant de retourner le contenu numéro i de la page.
- La méthode **getNbPublications()** permettant de retourner le nombre des publications.
- La méthode **getNbNouvellesPub()** permettant de retourner le nombre des nouvelles publications. Ce nombre est calculé en faisant la différence entre le nombre actuel des publications et le nombre de publications anciennes (**nbAncienPub**).
- La méthode **filtrer** (String m) permettant d'afficher les publications contenant le mot clé m donné en paramètre

### **Partie 3**

- Écrivez le code Java d'une classe de test nommée **SystèmeFB** qui contient une seule méthode main permettant de :
- Déclarer 3 pages à partir de ce tableau.
- La page P1 s'abonne à la page P2 et P3.
- La page P2 publie le statut suivant : « On va gagner !!!! »
- La page P3 publie le statut suivant : « j-1, RDV demain à 9h, vous pouvez gagner plein des cadeaux »
- La page P1 publie le statut suivant : « Bonjour les amis »
- Afficher la page P1
- Rechercher les publications de la page P1 qui contiennent le terme « gagner ».
- Sélectionner la deuxième publication, l'aimer et la commenter avec le texte « ok » par la page P1.
- Afficher la page P2.