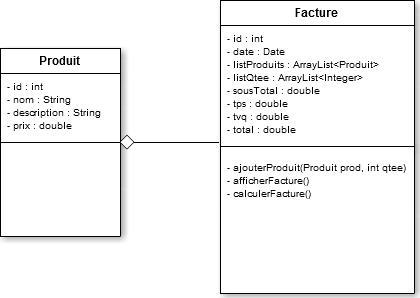
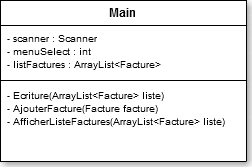
# Mise en contexte

Nous créons une application de facturation pour une boutique. Voici les deux principales classes, qui représentent des objets :



Nous avons également ajouté une classe Main, afin de contrôler le flux de l’application :

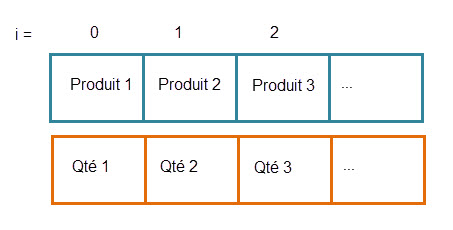


# Objets, structures de données, tableaux

## Produit

* 1. Créez l’objet Produit dans le même package que l’objet Facture.
     1. Ajoutez les champs de l’objet, tel que mentionné dans le diagramme de classes.
     2. Les champs doivent être « private »
     3. Générez les getters et setters, afin de pouvoir avoir accès aux champs.
     4. Ajoutez un constructeur, qui prend en paramètre le id, le nom, la description et le prix.
        1. Affectez les champs de l’objet avec les paramètres du constructeur.

## Facture

* 1. Dans l’objet Facture;
     1. Ajoutez un constructeur qui prend en paramètre l’id et la date de la facture.
        1. Affectez les champs de l’objet avec les paramètres du constructeur.
     2. Ajoutez une fonction calculerFacture.
        1. La fonction ne prend aucun paramètre d’entrée, et ne retourne rien.
        2. La fonction doit calculer le sous-total, la tps, la tvq et le total, et affecter les champs de la classe.
           1. Pour calculer le sous-total, une boucle doit parcourir la liste de produits, multiplier la quantité par le prix unitaire. La quantité se trouve dans une deuxième liste, mais exactement au même indice;
           2. Accédez aux valeurs de la liste de produits en utilisant l’accesseur de l’ArrayList, puis celui de l’objet Produit :

listProduits.get(i).getPrix()

* + - * 1. Accédez aux valeurs de la liste de quantités en utilisant l’accesseur de l’ArrayList :

listQtee.get(i)

* + - * 1. Calcul de la tps : sous-total \* 0.05
        2. Calcul de la tvq : sous-total \* 0.095

## Main

* + 1. Complétez la fonction Ecriture
       1. La fonction prend en entrée une ArrayList de Facture
       2. Le fichier dans lequel sont enregistrées les factures doit se nommer « factures.txt », et être enregistré directement dans le répertoire du projet
       3. Dans le code de la fonction, vous devez vérifier si le fichier existe, et le créer s’il n’existe pas, avant d’écrire la liste de factures dans ce fichier.
       4. Afin de pouvoir écrire un objet dans un fichier et éventuellement le récupérer, il ne faut pas oublier d’implémenter l’interface Serializable pour les classes Facture et Produit
       5. Utilisez l’objet ObjectOutputStream afin d’écrire l’ArrayList dans le fichier.
    2. Complétez la fonction main;
       1. Ajoutez l’appel à la fonction Ecriture (case 2)
       2. Ajoutez le code dans le case 1 pour effectuer une nouvelle vente (ajouter une facture).
          1. Dans le case 1 (du sous-menu ajouter produit dans facture), suite aux informations concernant le produit à ajouter dans la facture, instanciez un nouveau Produit, en utilisant les informations obtenues de l’utilisateur

Notez que l’utilisateur entre manuellement toutes les informations d’un produit à ajouter à la facture

* + - * 1. Ensuite, ajoutez le produit dans la facture, en utilisant l’appel de la fonction approprié