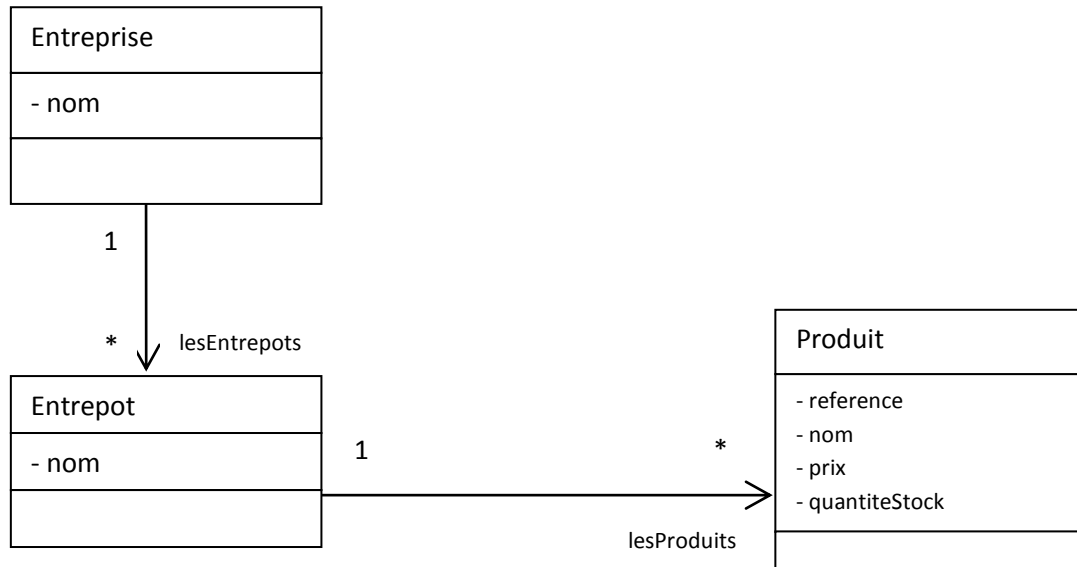


BLOC 2 – Ch 10 – Suite de l'exercice Burotic93 (collections, associations)

Objectif : coder sur poste l'application utilisée par Burotic93 pour gérer ses stocks de produits.

Une classe **Entreprise** représente une entreprise (qui pourrait être Burotic93 ou une filiale, ou un partenaire commercial). Une entreprise possède plusieurs entrepôts, où sont stockés les produits.

On vous donne le diagramme de classes partiel suivant :



Travail à faire :

- 1) Créer dans Netbeans le projet **Burotic93** avec le package **burotic93**.
- 2) Ajoutez dans ce package la classe **Produit**. Vous pouvez récupérer le code de la correction sur Trello.
- 3) Ajoutez dans ce package la classe **Entrepot**. Vous pouvez récupérer le code de la correction sur Trello.
- 4) Ajoutez dans ce package la classe **Entreprise**. Codez la classe, elle possède :
 - un attribut **nom**
 - et un attribut **lesEntrepots**

- 5) Ajoutez dans la classe **Entreprise** les méthodes suivantes :

- le constructeur :

```
public Entreprise(String n)
```
- l'accesseur :

```
public ArrayList<Entrepot> getLesEntrepots()
```
- une méthode qui ajoute un entrepôt à la liste **lesEntrepots** :

```
public void ajouteEntrepot(Entrepot unEntrepot)
```

- 6) Ajoutez une classe principale **Program** et codez dans la méthode **main** :

```
public static void main(String[] args)
{
    // Entrepôt de Marseille
    Entrepot unEntrepot = new Entrepot("Marseille");

    Produit p1 = new Produit("PH01", "photocopieuse", 4500, 1);
    Produit p2 = new Produit("PO01", "portable", 1500, 20);
    Produit p3 = new Produit("EC10", "écran", 300, 25);

    unEntrepot.ajouteProduit(p1);
    unEntrepot.ajouteProduit(p2);
    unEntrepot.ajouteProduit(p3);
}
```

```

// 2ème entrepôt de Lyon
Entrepot entrepot2 = new Entrepot("Lyon");

Produit p4 = new Produit("PHO1", "photocopieuse", 4500, 3);
Produit p5 = new Produit("ENC1", "enceinte", 60, 0);

entrepot2.ajouteProduit(p4);
entrepot2.ajouteProduit(p5);

// Bobigny
Entrepot entrepot3 = new Entrepot("Bobigny");

Produit p6 = new Produit("PHO1", "photocopieuse", 4500, 1);
Produit p7 = new Produit("SCA1", "scanner", 60, 0);

entrepot3.ajouteProduit(p6);
entrepot3.ajouteProduit(p7);

// instantiation de l'Entreprise
Entreprise entreprise = new Entreprise("Burotic93");

// on ajoute les entrepôt à l'entreprise :
entreprise.ajouteEntrepot(unEntrepot);
entreprise.ajouteEntrepot(entrepot2);
entreprise.ajouteEntrepot(entrepot3);

// affichage des entrepôts de l'entreprise:
System.out.println( entreprise.getLesEntrepots() );
}

```

Résultat : cela affiche tous les entrepôts et tous leurs produits, conformément aux méthodes **toString()** qui ont été codées dans **Entrepot** et **Produit**.

- 7) Codez dans la classe **Entreprise** la méthode **getQuantiteStockE()** qui retourne la quantité en stock de tous les produits de tous les entrepôts. Sa signature est :

```
public int getQuantiteStockE()
```

Indication : vous devez réutiliser la méthode **getQuantiteStockTotale** de la classe **Entrepot**.

Testez. Résultat attendu : 50.

- 8) Codez dans la classe **Entreprise** la méthode **getStockProduit** qui retourne la quantité totale en stock (dans tous les entrepôts) d'un produit dont la référence est passée en paramètre. Sa signature est :

```
public int getStockProduit(String ref)
```

Indication : vous devez réutiliser la méthode **rechercherProduit** de la classe **Entrepot**.

Testez plusieurs cas : un produit qui existe (référence « PHO1 » => résultat 5) et un produit inexistant (référence « ZZZ » => résultat 0).

- 9) Codez dans la classe **Entreprise** la méthode **getProduitsEnRupture** qui retourne la liste de tous les produits en rupture de stock, c'est-à-dire les produits qui ont un stock égal à 0, dans au moins un entrepôt. Sa signature est :

```
public ArrayList<Produit> getProduitsEnRupture()
```

Indication : vous devez réutiliser la méthode **getProduitsEnRupture** de la classe **Entrepot**. Pour ajouter une liste à une autre liste, vous pouvez utiliser la méthode **addAll** de **ArrayList**.

Testez. Résultat attendu :

```
[produit : ref ENC1, enceinte, 60.0 eur., qte stock : 0, produit : ref SCA1, scanner,
60.0 eur., qte stock : 0]
```