

Exercice de synthèse

OBJECTIFS

Cet exercice est adapté d'un sujet de BTS Informatique Industrielle des années 80. Il reprend de nombreux d'éléments abordés dans l'ensemble du cours. Il permet également de faire la transition entre le cours UML, la gestion des structures de données et le langage C++.

Cahier des charges :

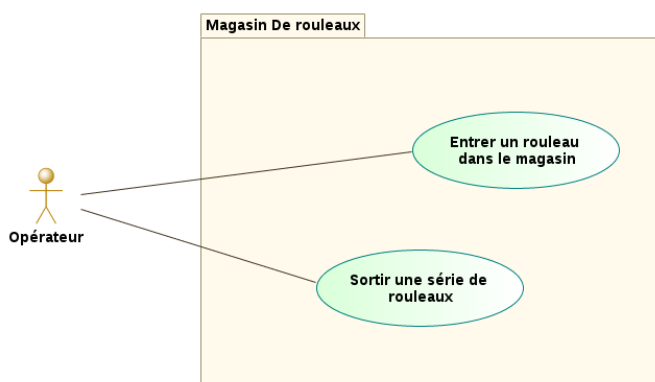
Un atelier assure la finition de rouleaux d'appareils de reprographie. Il traite des rouleaux provenant soit de la fabrication soit du décapage. Les rouleaux sont déposés six par six dans une enceinte hermétique dans laquelle on peut faire le vide. Les rouleaux sont déposés au-dessus d'un creuset contenant du sélénium et sont animés d'un mouvement de rotation permettant de déposer une couche uniforme. Sous l'effet de la chaleur et du vide, le sélénium s'évapore puis se condense sur la surface des rouleaux en couche uniforme de soixante microns.

Avant leur traitement, les rouleaux sont stockés dans un magasin formé d'alvéoles. Il comporte dix rangées de vingt alvéoles permettant de stocker deux cents rouleaux à raison d'un par alvéole. Chacune est repérée par son numéro de rangée suivi par son numéro de colonne, origine en haut, à gauche. Un chariot élévateur se déplaçant sur des rails saisit les rouleaux à l'entrée du magasin pour les ranger dans le magasin de stockage. Il permet aussi de les saisir pour les emmener vers l'enceinte de traitement. Il déplace un seul rouleau à la fois de la zone de stockage vers l'enceinte ou l'inverse.

Un opérateur saisit au fur et à mesure la référence et le diamètre du rouleau sur la console de son poste de travail et donne l'ordre au chariot de le ranger dans le magasin. Pour le retraitement, la quantité de sélénium déposée dans le creuset est proportionnelle au diamètre du plus gros rouleau contenu dans l'enceinte. Pour économiser le sélénium, il faut que les diamètres respectifs des six rouleaux de l'enceinte soient le plus proche possible. L'opérateur par une simple demande permet la sortie d'une série de six rouleaux du magasin en vue du retraitement. Les caractéristiques des rouleaux sorties sont affichées sur l'écran.

Analyse : Étude du magasin de rouleau

Le diagramme de cas d'utilisation ci-dessous définit le périmètre de notre étude.



Acteur du système :

Acteur	Description
Opérateur	<p>Il effectue la saisie des caractéristiques d'un rouleau présent à l'entrée du magasin puis le chariot est sollicité pour le ranger dans une alvéole.</p> <p>Il donne l'ordre de sortir la meilleure série de six rouleaux du magasin vers l'enceinte de retraitement lorsque celle-ci est prête pour un nouveau cycle.</p>

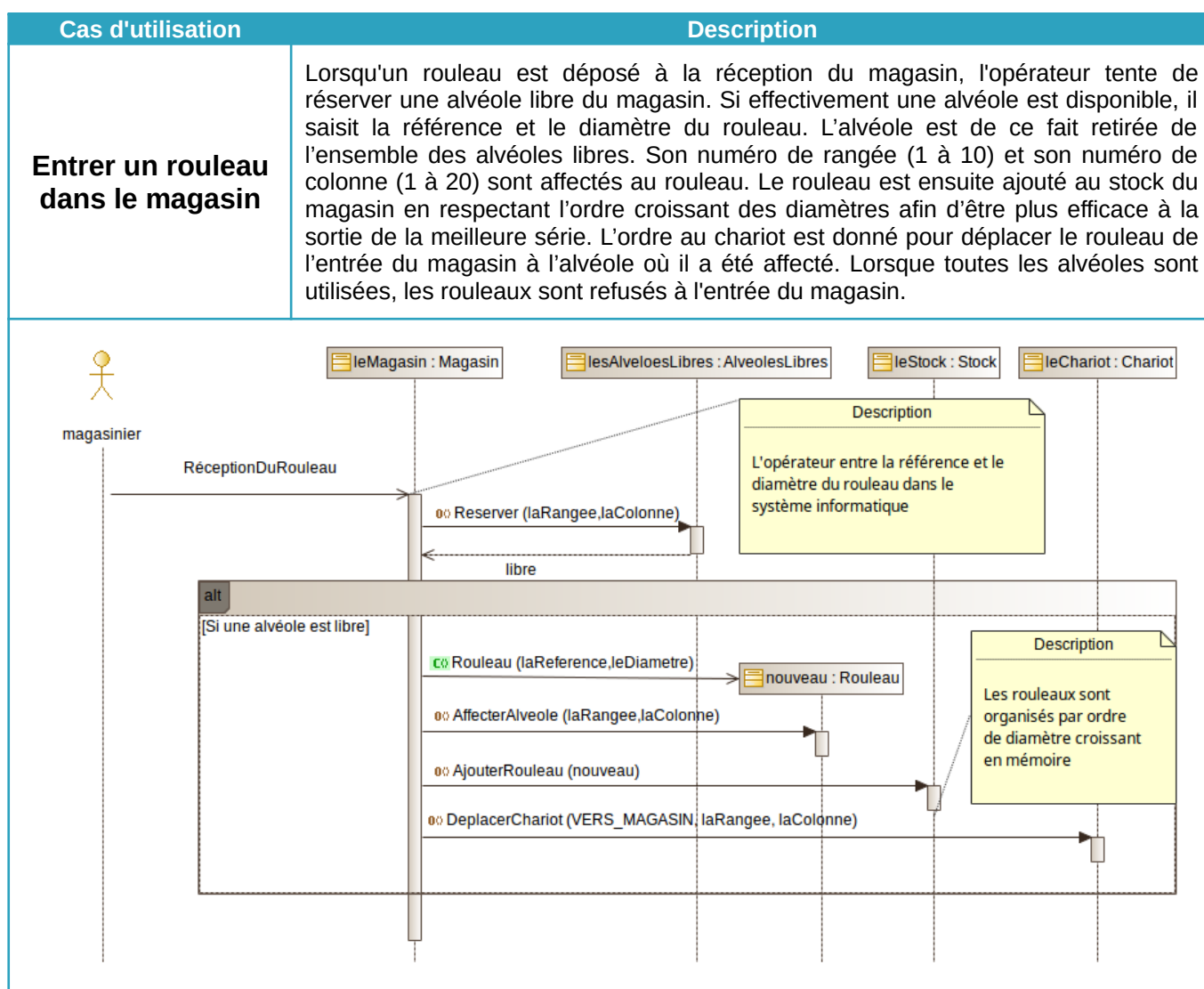
1. Complétez les numéros d'alvéoles en fonction des rangées et des colonnes du magasin.

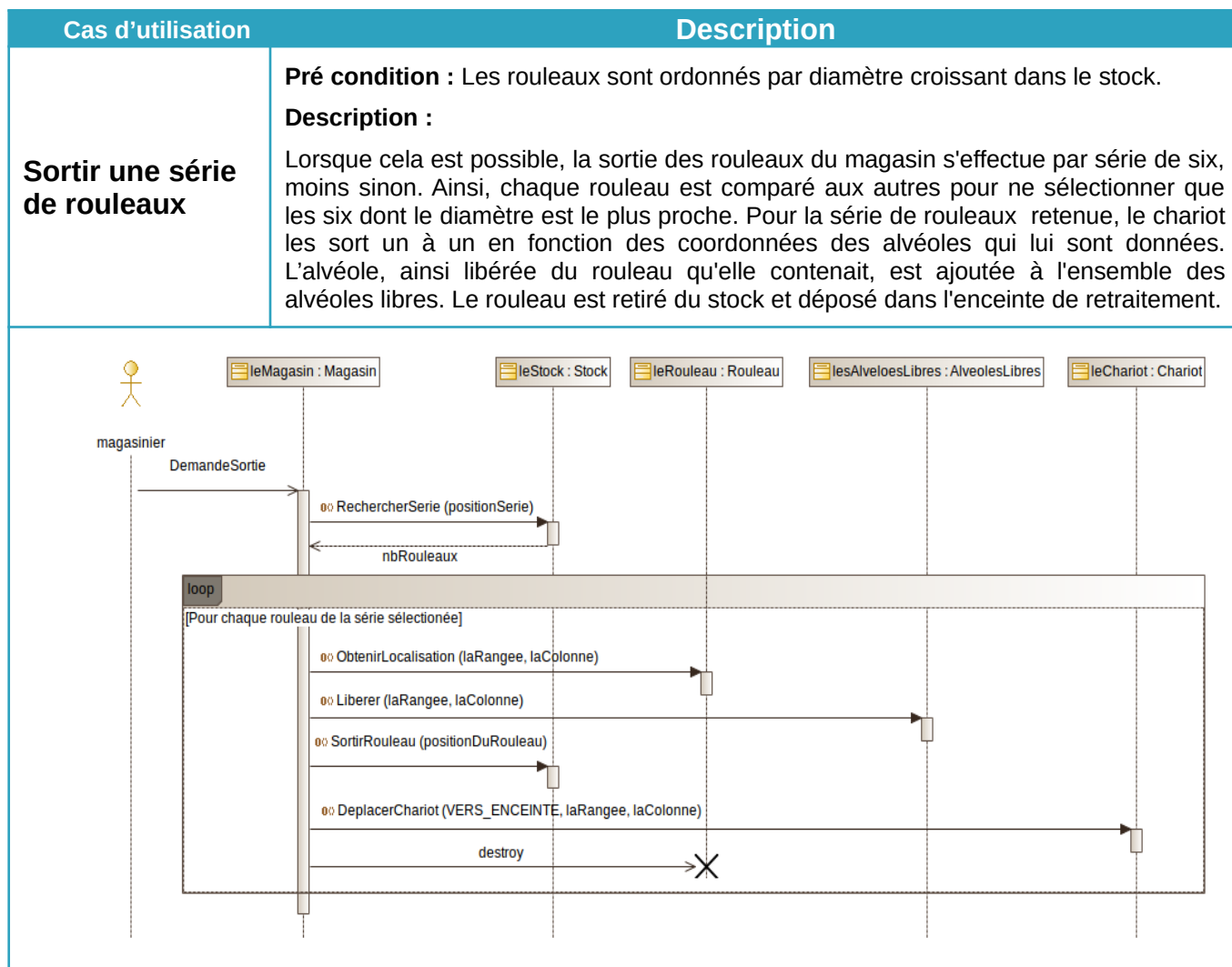
Le magasin de rouleaux		Les colonnes								
		1	2	3	4	...	17	18	19	20
Les rangées	1	1								
	2									
	...									
	9									
	10									200

2. Proposer une formule de calcul permettant de donner le numéro d'alvéole en fonction du numéro de rangée et du numéro de colonne et les formules inverses donnant la rangée ou la colonne en fonction du numéro d'alvéole et du nombre total de colonnes.

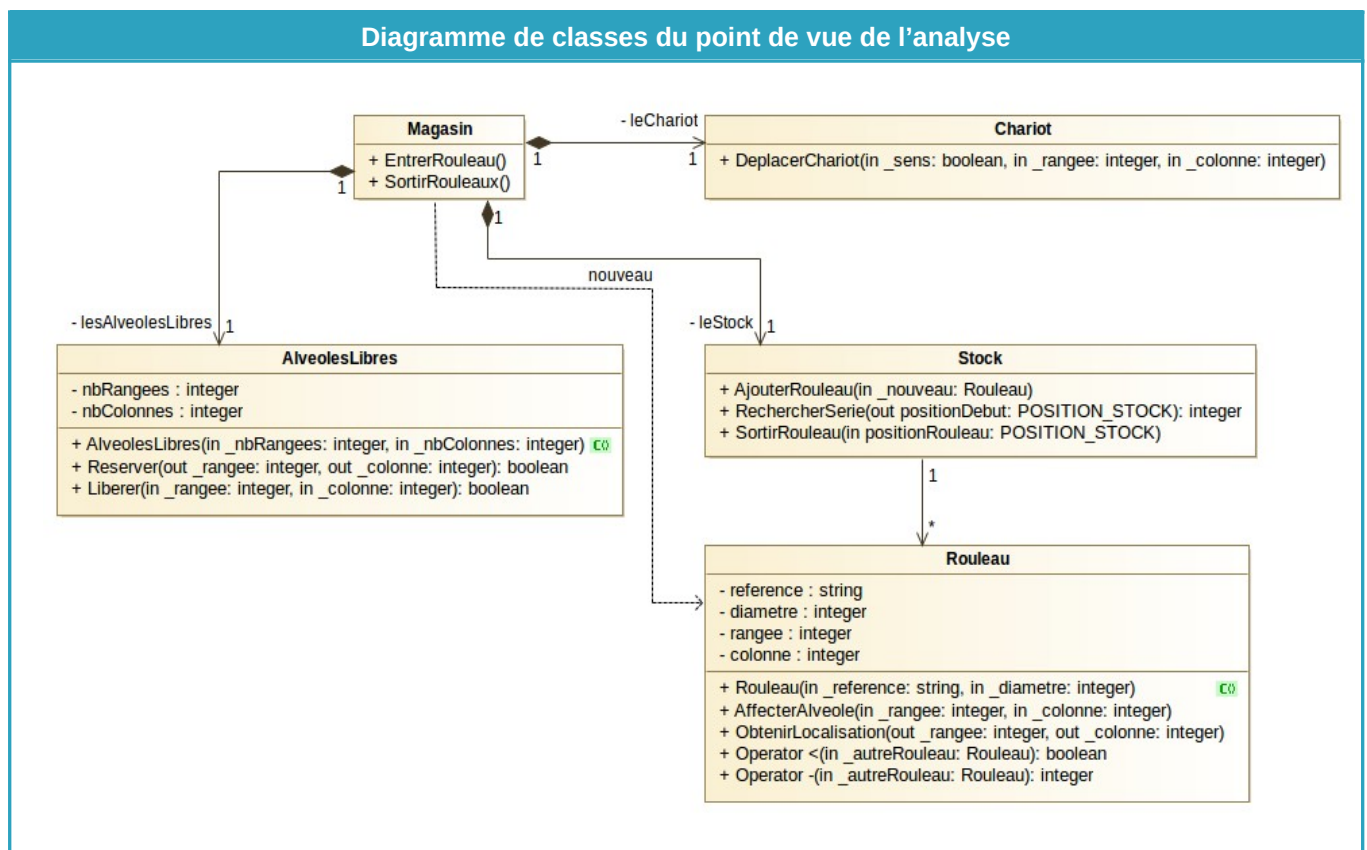
numAlveole =
rangée =
colonne =

Description des cas d'utilisation :





À partir des deux diagrammes de séquence précédent, on obtient le diagramme de classes suivant :



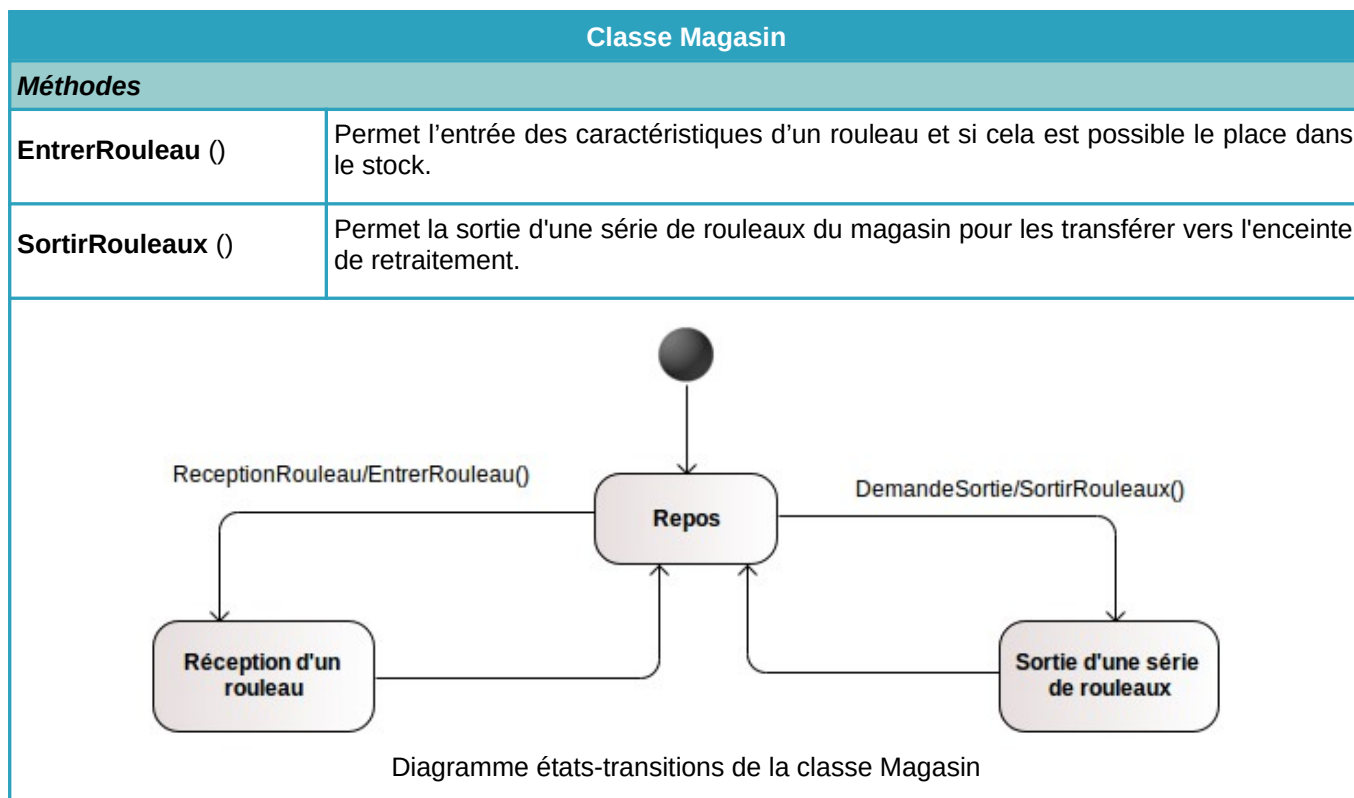
Descriptions des classes :

Classe Rouleau		
Méthodes		
Rouleau (IN _reference string, IN _diametre integer)		Constructeur de la classe, il initialise la référence et le diamètre du rouleau.
AffecterAlveole (IN _rangee integer, IN _colonne integer)		Attribue une alvéole (rangée et colonne) à un rouleau qui vient d'entrer dans le magasin.
ObtenirLocalisation (OUT _rangee integer, OUT _colonne integer)		Restitue le numéro de rangée et de colonne où est stocké le rouleau dans le magasin.
Boolean ← Operator < (IN _autreRouleau Rouleau)		Vrai si rouleau1 < rouleau2, faux sinon
Integer ← Operator - (IN _autreRouleau Rouleau)		Retourne la différence entre les diamètres du rouleau2 et du rouleau1
Attributs		
reference	: string	Code fournisseur
diametre	: integer	Diamètre du rouleau exprimé en millimètre [100 à 300]
rangee	: integer	Numéro de la rangée où est stocké le rouleau [1 à 10]
colonne	: integer	Numéro de la colonne où est stocké le rouleau [1 à 20]

Classe Chariot
Méthodes
DeplacerChariot (IN _sens boolean, IN _rangee integer, IN _colonne integer)
Déplace le chariot vers ou depuis la rangée et la colonne avec _sens : VERS_MAGASIN → FAUX ; VERS_ENCEINTE → VRAI

Classe AlveolesLibres	
Méthodes	
AlveolesLibres (IN _nbRangees integer, IN _nbColonnes integer)	Constructeur de la classe, il initialise les alvéoles libres. Au démarrage de l'application, toutes les alvéoles sont libres. Les valeurs par défauts sont les valeurs imposées par le magasin soit respectivement 10 et 20.
boolean Reserver (OUT _rangee integer, OUT _colonne integer)	Retourne vrai, si une alvéole est libre, dans ce cas, la méthode fournit la localisation (sa rangée et sa colonne) et la supprime des alvéoles libres. Si aucune alvéole n'est disponible, la méthode retourne faux
Liberer (IN _rangee integer, IN _colonne integer)	Ajoute l'alvéole dont les coordonnées sont passées en paramètre à l'ensemble des alvéoles libres.
Attributs	
nbRangees : integer	Nombre de rangées dans le magasin
nbColonnes : integer	Nombre de colonnes dans le magasin

Classe Stock	
Méthodes	
AjouterRouleau (IN _nouveauRouleau Rouleau)	Ajoute un rouleau dans le stock en tenant compte de la taille de son diamètre. Les rouleaux sont classés par ordre croissant de diamètre.
integer RechercherSerie (OUT _positionDebut POSITION_STOCK)	Recherche dans le stock la série de rouleaux dont le diamètre est le plus proche. Retourne le nombre de rouleaux à sortir (au maximum 6). En paramètre de sortie si le paramètre de retour est supérieur à 0, fournit la position du premier rouleau de la série à sortir.
SortirRouleau (IN _positionRouleau POSITION_STOCK)	Enlève le rouleau dont la position est passée en paramètre du stock (en mémoire de l'ordinateur).



Conception du système :

Les deux diagrammes de séquence de l'analyse ainsi que le diagramme de classe font apparaître la classe **AlveolesLibres**.

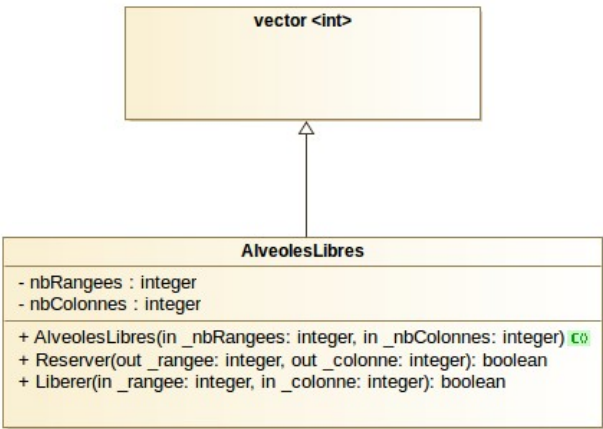
3. Indiquez son rôle dans l'application.
4. Proposez une structure de données capable de tenir ce rôle, justifiez votre choix.
5. Quelle taille doit-elle posséder ?
6. Comment cette structure doit-elle être initialisée ?

De même, les deux diagrammes de séquence de l'analyse utilisent la classe **Stock**.

7. Indiquez son rôle dans l'application.
8. Proposer une structure de données capable de tenir ce rôle, justifiez votre choix.
9. Comment cette structure doit-elle être initialisée ?
10. Quel objet parmi ceux représentés dans l'analyse mémorise la position d'un rouleau dans le magasin ?

Réalisation du système

Pour la suite de l'étude, on propose d'implémenter les alvéoles libres sous la forme d'une pile. L'utilisation de la classe **stack** de la librairie STL est tout à fait appropriée. Si l'on souhaite pouvoir visualiser son contenu, il devient nécessaire d'utiliser à la place la classe **vector**.



L'implémentation est réalisée sous la forme d'un héritage. La classe **AlveolesLibres** devient donc une sorte de **vector** contenant des entiers. Les alvéoles libres seront stockées dans cette pile.

- 11. Quel fichier doit être inclus pour utiliser le conteneur ?
- 12. Quel espace de nommage est utilisé par ce conteneur ?
- 13. Complétez le tableau ci-après afin de déterminer quelles méthodes de la classe de base vont être appelées dans chaque méthode de la classe **AlveolesLibres**. Vous préciserez également pour quel usage.

Vous pouvez utiliser la documentation de référence des conteneurs : <https://fr.cppreference.com/w/cpp/container> pour répondre.

Méthodes de AlveolesLibres	Méthode de vector	Usage de la méthode
Constructeur		
Reserver		
Liberer		

- 14. Réalisez le codage de la classe **Alveoleslibres** en C++. Ajoutez une méthode **Visualisez** à cette classe afin de visualiser le contenu de la pile.
- 15. Réalisez un programme principal utilisant la classe **Menu** afin d'effectuer un test fonctionnel de votre classe. Pour les besoins du test, le constructeur peut recevoir comme paramètres 2 et 5 par exemple. Vérifiez par exemple qu'il n'est pas possible de réserver une alvéole si la pile est vide (un message s'affiche dans le programme principal). De même, vérifiez qu'il n'est pas possible de libérer une alvéole si la pile est saturée. Après chaque appel d'une méthode de la classe **AlveolesLibres**, on peut visualiser son contenu.



À présent, intéressons-nous à la classe **Rouleau**, cette classe ne réalisant pas d'allocation dynamique de mémoire, il n'y a pas besoin de surcharger le constructeur, de créer un constructeur de copie et un destructeur pour que la classe soit dite **Coplien**.

16. Réalisez le codage de la classe **Rouleau** en C++. Ajoutez une méthode **Visualisez** à cette classe afin de visualiser son contenu.
17. Réalisez un programme principal permettant la vérification du fonctionnement de cette classe.

```

Terminal
Rouleau référence : A-320 Diamètre : 200
Dans magasin en 4 - 5
Rouleau référence : B-747 Diamètre : 150
Dans magasin en 2 - 3
r1 est en 4 - 5
r2 est en 2 - 3
r1 est supérieur ou égal à r2
la différence entre les diamètres est de : 50
Appuyez sur <ENTRÉE> pour fermer cette fenêtre...

```

Voici un exemple de résultat recherché pour cette classe avec deux rouleaux instanciés de la manière suivante :

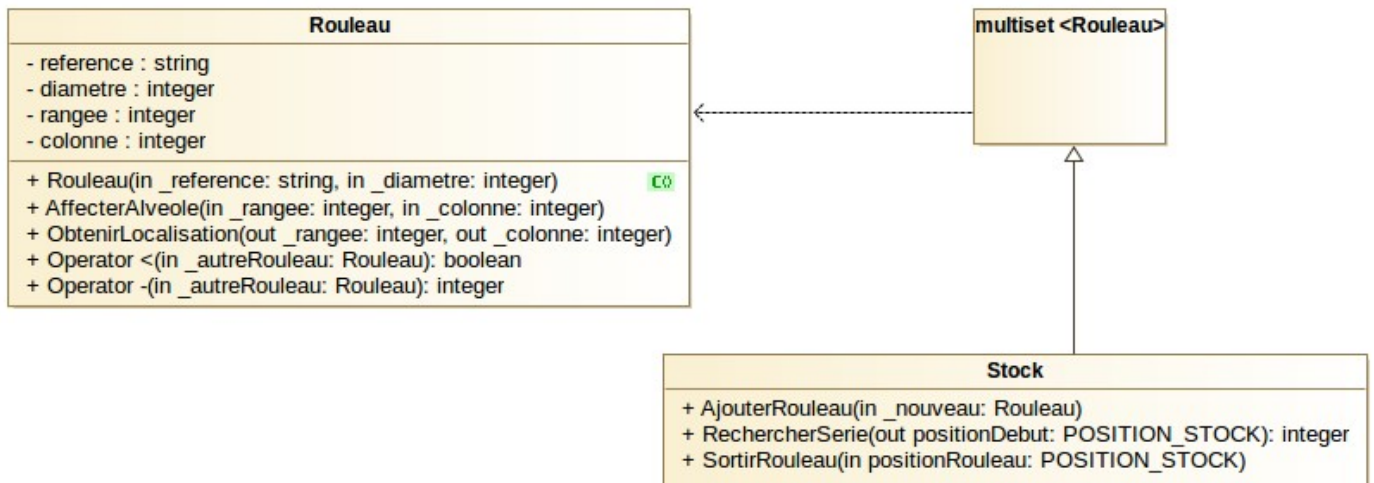
```

Rouleau r1("A-320", 200);
Rouleau r2("B-747", 150);

```

Le premier est dans l'alvéole (4,5), le second dans celle en (2,3).

Maintenant que nous disposons de la classe **Rouleau**, nous pouvons réaliser la classe **Stock** qui doit contenir ces rouleaux. Afin de faciliter la recherche de la meilleure série de rouleaux, ceux qui ont un diamètre le plus proche, les rouleaux doivent être organisés dans le stock par ordre croissant de diamètre. Le conteneur de type **set** de la librairie STL permet de réaliser un ensemble ordonné d'éléments, cependant comme il peut y avoir des rouleaux possédant le même diamètre, notre ensemble doit supporter les doublons d'où le choix du **multiset**. Comme pour les alvéoles libres, nous allons faire hériter **Stock** de **multiset** pour simplifier les traitements ainsi la classe **Stock** devient une sorte de **multiset**.



La relation de dépendance montre juste que le **multiset** utilise la classe **Rouleau**. La relation entre **Stock** et **Rouleau** représentée lors de la phase d'analyse a été remplacée par la relation d'héritage pour répondre à la nouvelle implémentation choisie.

18. Quel fichier doit être inclus pour utiliser le conteneur ?
19. Quel espace de nommage est utilisé par ce conteneur ?

20. Réalisez le codage de la classe **Stock** en C++. Ajoutez une méthode **Visualisez** à cette classe afin de visualiser le contenu du stock.
21. Réalisez un programme principal utilisant la classe **Menu** afin d'effectuer un test fonctionnel de votre classe.

```

Terminal
+-----+
| 1 | Ajouter un Rouleau |
| 2 | Rechercher une serie de rouleaux |
| 3 | Sortir un rouleau |
| 4 | Visualiser |
| 5 | Quitter |
+-----+
votre choix svp : 

```

Voici un exemple de menu, pour l'ajout du rouleau, on peut se contenter de saisir une référence et un diamètre de rouleau.

Lors de l'ajout, vérifiez bien que les rouleaux soient bien classés par ordre croissant. Dans votre jeu de test, prévoir les cas : insertion avant, entre-deux, au début et à la fin de l'ensemble ordonné.

Pour la suppression, on peut demander un numéro correspondant à un numéro d'ordre dans l'ensemble, puis incrémenter un itérateur jusqu'à la position souhaitée et enfin appeler la méthode **SortirRouleau**. Chaque fois, la méthode **Visualiser** permet d'afficher le stock.

Intégration

Vous pouvez poursuivre dans un nouveau projet nommé **MagasinDeRouleaux** afin d'intégrer l'ensemble des classes. En respectant les diagrammes de séquence, codez la classe **Magasin**. Pour coder la classe **Chariot**, on peut simplement se contenter d'afficher un message pour indiquer le sens du transfert vers l'enceinte ou vers le magasin et les coordonnées de l'alvéole. Le programme principal appelle les deux méthodes de la classe **Magasin**, vous pouvez à nouveau utiliser un menu pour faire cela.

```

Terminal
+-----+
| 1 | Entrer un rouleau |
| 2 | Sortir une serie de rouleaux |
| 3 | Visualiser le magasin |
| 4 | Quitter |
+-----+
votre choix svp : 

```

La saisie des données se fait dans la classe **Magasin**.

Voici un exemple de visualisation :

```

Terminal
Rouleaux dans le magasin :
Rouleau référence : B-747 Diamètre : 150
Dans magasin en 2 - 4
Rouleau référence : A-320 Diamètre : 200
Dans magasin en 2 - 5
Numéros d'alvéoles disponibles :
1 2 3 4 5 6 7 8
appuyer sur la touche Entrée pour continuer...

```

22. Avec Modelio, réalisez le diagramme de classes complet de l'application. Pour les classes conteneurs, ne reproduisez que les méthodes utilisées.