

Devoir de spécification – (10%) : *Gestion de stock*

À remettre *avant* 23h55, le mardi **19 avril 2017**.  
Pénalité de retard : 0.2 point / 4 par *heure* de retard.  
Travail en équipe de 2 à 4 personnes.

Il s'agit de modéliser une version simplifiée d'un système de *gestion de stock* avec le langage Perfect.

## 1 Description du problème

### 1.1 Spécification fournie

Une entreprise vend des produits en détail. Pour ce faire, elle gère un stock de produits. Une étude de marché a révélé un ensemble de **produits** constituant le *marché cible* de cette entreprise. L'entreprise gère un **stock** qui consiste en une certaine **quantité** de produits associée à chacun de ses produits. Deux produits différents peuvent avoir des quantités en stock différentes. La **capacité** de stockage de chacun des produits est la même pour tous et sera représentée par une constante entière positive de votre choix. Ainsi, la quantité de chacun des produits en stock ne peut dépasser cette capacité. En outre, **à chaque produit en stock** est associé un **seuil** minimal. Quand la quantité d'un produit, suite à une vente, baisse en deçà de ce seuil, il faut renouveler le produit, c'est-à-dire en acheter une autre quantité. Le système de gestion devrait tenir un registre de **renouvellement** dans lequel il recense les quantités achetées de chacun des produits dans le but de mieux cibler le marché. Les produits qui sont renouvelés en les plus grandes quantités sont ceux qui seraient jugés les plus populaires. On prend une quantité  $q$  donnée comme paramètre par le gérant à l'opération déterminant la popularité des produits, si le renouvellement d'un produit est supérieur ou égal à cette quantité on dit qu'il est populaire. **L'ensemble des produits gardés en stock doivent faire partie des produits désignés par l'ensemble identifié comme marché cible.** À chacun sera associé un seuil minimal qui sera fourni dès l'initialisation, en paramètre d'entrée. Le renouvellement d'un produit commence à zéro et évolue à chaque fois qu'on achète une quantité d'un produit en rajoutant la quantité achetée. La quantité gardée en stock d'un produit doit être maintenue supérieure ou égale au seuil minimal et inférieure ou égale à la capacité de stockage. Initialement le gérant commence avec un ensemble de produits considéré comme le marché cible initial, un ensemble de produits en stock avec leurs quantités associées, une association produit seuil minimal pour chacun des produits en stock. Ces informations doivent être fournis en paramètres dès l'initialisation. Évidemment, il faut prendre garde que ces paramètres respectent bien l'invariant.

La spécification inclut :

1. les types utilisés,
2. la classe **Stock** qui spécifie :
  - (a) les variables de l'état général,
  - (b) les invariants portant sur les variables de l'état général,
  - (c) l'initialisation de la classe (le schéma "build"),
  - (d) un schéma pour acheter une quantité  $q$  d'un produit  $p$  déjà en stock. Évidemment, ça précise les préconditions pour réaliser l'achat,
  - (e) un schéma pour acheter une quantité  $q$  d'un produit  $p$  faisant partie du marché cible mais qui n'est pas un stock. Évidemment, on précise le seuil minimal  $s$  à ce nouveau produit,

- (f) un schéma pour vendre une quantité  $q$  d'un produit  $p$  en stock et les préconditions qui permettent de réaliser la vente,
- (g) un schéma pour vendre une quantité  $q$  d'un produit  $p$  si cette vente fait baisser la quantité en stock en deçà du seuil. Ici, il faut acheter puis vendre, séquençement d'appels de schémas,
- (h) un schéma qui donne la l'ensemble des produits les plus en demande, en comparant la quantité renouvelée à une quantité  $q$  donnée en paramètre. Les Produits ayant un renouvellement supérieur ou égal à  $q$  sont considérés populaires,

## 1.2 Besoins spécifiques du travail à faire

Le gérant veut améliorer la gestion de stock et ce en utilisant deux stratégies. La première consiste à enregistrer les quantités renouvelées de ce produit et détecter ainsi les produits dont la demande excède une certaine quantité. La deuxième consiste à faire un sondage de satisfaction pour savoir quels sont les produits susceptibles de satisfaire les clients. La décision de retirer un produit du stock est basée sur la combinaison de ces deux stratégies.

Le système est modélisé via les éléments suivants :

1. Types : **Produit**, **Quantite** et **Niveau : Produit** représente le type abstrait des produits, **Quantite** représente les quantités des produits en stock et **Niveau** représente le niveau de satisfaction estimé, le niveau associé à un produit est un entier entre 1 et 10.
2. Constantes : **capacite**, **satisfactionSeuil** et **renouvellementSeuil**. **capacite** représente la capacité maximale de stockage pour les produits. Elle est la même pour tous les produits. **satisfactionSeuil** est le niveau de satisfaction minimal. **renouvellementSeuil** est la quantité de renouvellement de produits. En deçà de cette quantité le renouvellement est considéré faible.
3. Variables d'état général : **stock**, **marcheCible**, **produitsPopulaires**, **produitsNonPopulaires**, **seuil**, **renouvellement**, **satisfaction**, **fermeture**. **stock** est une fonction qui associe la quantité en stock d'un produit à ce produit. **marcheCible** est un ensemble de produits représentant la ligne de produits sous la responsabilité du gérant. Les produits en stock sont répartis en deux ensembles disjoints **produitsPopulaires** et **produitsNonPopulaires** dont l'union représente l'ensemble des produits en stock. Un produit est considéré en stock tant qu'on n'a pas décidé de le retirer des ventes et ce même si sa quantité est à zéro. Un produit est mis dans l'ensemble **produitsPopulaires** si son renouvellement excède ou est égal à **renouvellementSeuil**, sinon il est placé dans **produitsNonPopulaires**. **seuil** est une fonction qui associe à chaque produit en stock un seuil minimal de quantité devant se trouver en stock. **renouvellement** est une fonction qui associe à chaque produit en stock le cumul des quantités achetées. **satisfaction** est une fonction qui associe à chaque produit en stock un niveau de satisfaction. **fermeture** est une variable booléenne si elle est vraie cela veut dire que la ligne de produits va être fermée par le gestionnaire.
4. Conditions de fonctionnement :
  - On ne fonctionne pas avec une ligne de produits vide (**marcheCible**),
  - Les produits en stock doivent faire partie de la ligne de produits
  - Tous les produits sont retirés du stock et **fermeture** est vraie ou bien au moins un produit n'est pas retiré du stock et **fermeture** est fausse.
  - La quantité d'un produit en stock ne doit pas excéder **capacite**
  - Le seuil d'un produit en stock ne doit pas excéder **capacite**
  - La quantité d'un produit en stock doit être au moins égale au seuil associé à ce produit.
  - La popularité des produits est mise à jour par le gérant périodiquement pour répartir les produits en stock en deux ensembles disjoints les produits populaires et les non populaires et ce en comparant leur renouvellement au **renouvellementSeuil**. Attention la condition pour qu'un produit soit populaire ou non ne doit pas figurer comme invariant car il faut supposer

initialement qu'un produit est populaire quand on l'introduit en vente pour lui donner sa chance.

- Si le produit n'est pas populaire **et** que son niveau de satisfaction n'est pas acceptable, moins que **satisfactionSeuil**, alors on décide de mettre son seuil de stockage à zéro et ainsi on le liquide en permettant des ventes sans renouvellement puisque le seuil est à zéro. Quand la quantité en stock de ce produit est à zéro, il est retiré du stock.
- Si l'ensemble des produits en stock devient vide il y a décision de fermeture.
- Un sondage de satisfaction peut être effectué à la demande du gérant.
- Le gérant peut décider de retirer un produit du marché cible. Il ne faut pas que ce produit soit en stock et il faut laisser au moins un produit dans le marché cible.

## 2 Travail à faire pour ce TP

L'état général, l'initialisation et les schémas de la spécification fournie sont à adapter pour respecter la nouvelle description des besoins.

- écrire une spécification en **Perfect** du système qui respecte les demandes suivantes :
  - Modélisez l'état général et l'état initial.
  - Modélisez les opérations suivantes en faisant attention aux invariants (opérations normales, pas nécessairement totales)
    1. Achat d'une certaine quantité d'un produit existant en stock. Évidemment, il ne faut pas qu'il y ait décision de fermeture de la ligne de produits.
    2. Achat d'une certaine quantité d'un produit faisant partie de la ligne des produits mais qui n'était pas en stock. En plus de la quantité du produit, il faut fournir un seuil de stockage et un niveau de satisfaction pour ce produit. Il ne faut pas qu'il y ait décision de fermeture pour pouvoir acheter.
    3. Vente d'une certaine quantité d'un produit en stock. Là aussi, il ne faut pas qu'il y ait décision de fermeture. Il faut que la quantité restant en stock respecte la contrainte du seuil minimal.
    4. Achat puis vente d'une quantité d'un produit populaire. Il ne faut pas qu'il y ait décision de fermeture. La quantité restant en stock va être plus basse que le seuil si on vend directement. Il faut donc acheter pour atteindre la capacité maximale puis vendre.
    5. Mise à jour de la popularité des produits selon le renouvellement.
    6. Mise à jour des seuils, on fournit une association produit quantité minimale de stockage. On associe à un produit le seuil fourni ou bien zéro s'il n'est pas populaire et si son niveau de satisfaction est plus bas que le seuil de satisfaction.
    7. Mise à jour des niveaux de satisfaction suite à un sondage. L'association produit niveau de satisfaction est fournie en paramètre.
    8. Retrait d'un produit du marché cible.
- écrire un document de spécification (seulement les sections 2, 4 et 5 que vous retrouvez dans les notes de cours, chapitre 7 du module 2) qui reflète votre spécification ; vous avez la possibilité de spécifier ces sections dans votre fichier .pd sous forme de commentaires.

La notation est approximativement celle-ci : 20 points pour l'état général et les invariants, 10 points pour l'initialisation et 70 points les opérations. Il y a des opérations qui seraient notées plus que d'autres selon la difficulté de l'opération. Vu qu'il y a inter-dépendance entre les spécifications des éléments de ce système, cette notation n'est pas rigide, vous pouvez avoir tout spécifié et la vérification ne passe pas alors il y aura pénalité. C'est donné à titre indicatif seulement. Même si la vérification et le check passent et que la spécification ne correspond pas à ce qui est demandé et qu'elle est incohérente, il y aura pénalisation.

**Modalités de remise :** Vous devez remettre *par intranet* les documents suivants dans un .zip. Four-nissez dans l'entête du fichier .pd en tant que commentaires le nom, le matricule et le programmes d'étude de chaque membre de l'équipe (perte de 5% des points sinon).

1. Votre projet : fichier .pd . CE FICHIER DOIT COMPILER (PASSER LA VÉRIFICATION SYN-TAXIQUE DE AINSI QUE LA VÉRIFICATION FORMELLE DE PERFECT DEVELOPER) EN UTILISANT LES COMMANDES **Check** et **Verify**. Si le **check** ne passe pas pénalité de 4 points sur 10. Si le **verify** ne passe pas, pénalité allant jusqu'à 7 points sur 10.

2. Un fichier (.pdf) contenant les sections 2, 4 et 5 du document de spécification. Des explications claires, bien structurées et bien écrites sont essentielles pour l'atteinte des objectifs de ce devoir. Encore une fois, vous pouvez décrire ces sections, sous forme de commentaires, à même votre fichier .pd. Dans ce cas, vous n'avez pas besoin de remettre un fichier .pdf.

**Autres contraintes :** Vous devez respecter les contraintes suivantes.

- Ce devoir se fait en équipe de 2 à 4 personnes.
- Vous devez prendre toutes les mesures nécessaires pour éviter le plagiat. Si deux équipes proposent des solutions trop semblables, des sanctions pourraient être prises conformément au règlement départemental.

Bon travail !