



1. Descriptif

Partie commune I21-I22

Le projet a pour objet de modéliser des distributeurs de denrées alimentaires : boissons en canettes, sucreries (barres chocolatées, sachets de madeines,...),... Les produits d'un distributeur donné sont tous de même type.

On souhaite construire la hiérarchie de classes qui permettent de représenter et de traiter ces différentes entités, sachant que :

1. Tous les distributeurs contiennent un clavier qui permet de sélectionner l'emplacement d'un produit. Cependant, le mode de sélection est différent suivant le type de distributeur :
 - Dans les distributeurs de sucreries, les compartiments sont placés en rangées. La sélection d'un produit nécessite de rentrer le code de la rangée (une lettre) et le rang du compartiment dans la rangée (de 1 à 5 de gauche à droite). Il y a 8 rangées de 5 compartiments. Les produits d'une même rangée ont le même prix¹.
 - Un distributeur de canettes comprend un nombre fixé de compartiments de même capacité en nombres de canettes. À chaque compartiment est associé un bouton du sélecteur. Les canettes d'un même compartiment sont de même type (même boisson, même contenance, même prix).
2. Les distributeurs contiennent un monnayeur, c'est-à-dire un jeu de compartiments contenant des pièces de monnaies de différents types (par exemple 3×1 centimes, 50×5 centimes, 32×10 centimes,..., 5×2€). Le monnayeur a une capacité maximale pour chaque type de pièces. Il est doté d'une fonction de rendu de monnaie qui, à partir d'une somme donnée par un utilisateur et d'un prix, renvoie le nombre des pièces de chaque type qui peut constituer la monnaie rendue — s'il est possible de rendre la monnaie.
3. Les sucreries ont une date de péremption, une marque, un nom de produit et un poids.
4. Tous les produits ont un code barre et sont dotés d'un prix.

Partie spécifique à I22

Le projet devra avoir une interface graphique permettant de mettre en œuvre les fonctionnalités de :

- Visualisation d'un distributeur — qui pourra être un distributeur de type canettes ou de type sucreries.
- Ajout d'un nouveau distributeur (option facultative dans le projet).

¹ tous les compartiments d'une même rangée ont des produits de même prix.

La visualisation d'un distributeur est faite dans une fenêtre qui offre un certain nombre de fonctionnalités communes pour les deux types de distributeurs ou bien spécifiques à un distributeur.

Tous les distributeurs doivent permettre la consommation d'un produit (soit directement en sélectionnant une canette, soit en entrant un code pour une sucrerie). La somme versée doit être saisie et le distributeur indique alors les informations sur le produit (sucrerie ou canette) servi ainsi que les pièces de monnaies rendues si nécessaire.

On peut avoir les informations sur un distributeur sous forme textuelle, et les informations sur son monnayeur (pièces qu'il contient).

On peut également « remplir » le distributeur (ajout de canettes ou de sucreries) ou ajouter des pièces dans le monnayeur.

2. Réalisation

Vous devez réaliser un programme java qui permet de modéliser au moins les éléments de base décrits par l'énoncé.

Modélisation (I21)

Le programme doit au moins comporter une classe pour représenter les types de canettes, les produits, les distributeurs ainsi qu'une sous-classe par catégorie de distributeur. Les classes doivent être hiérarchisées s'il y a lieu. Vous pouvez être amenés à créer des classes supplémentaires.

Les classes doivent comporter les constructeurs nécessaires. Les distributeurs, et compartiments sont construits vides.

Vous devez doter chaque contenant (compartiment, monnayeur,...) de méthodes permettant d'ajouter ou de retirer une partie de leur contenu (ajouter/retirer un produit à un compartiment, des pièces de certaines catégories au monnayeur,...). Vous devez contrôler les ajouts ou retraits excessifs par rapport aux capacités et au contenu actuel d'un contenant.

Vous devez spécifier les autres accesseurs utiles et placer les modificateurs d'accès (`public`, `private`) adéquats sur les attributs et les méthodes.

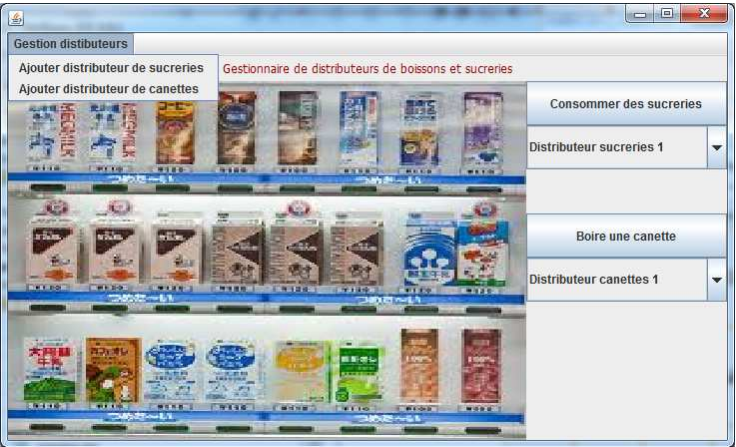
Une méthode `toString` doit être associée à chaque classe non abstraite. Elle doit synthétiser la structure de l'instance (par exemple les divers compartiments qu'un distributeur contient) et son état (les canettes contenues dans les compartiments).

Vous devez ajouter les méthodes permettant de faire fonctionner le distributeur, c'est-à-dire de choisir un produit, de rentrer le montant nécessaire — ou plus — en spécifiant les pièces de chaque type, et d'obtenir des pièces en retour ainsi que les caractéristiques du produit sélectionné ou une indication des causes d'échec (montant insuffisant, compartiment vide,...).

Interface (I22)

L'application pourra avoir une fenêtre principale permettant de choisir une fonctionnalité de consommation (Consommer des sucreries, Boire une canette) ou de gestion (Ajout d'un distributeur). Les options d'ajout sont facultatives (les instances de distributeurs peuvent être créées par programme).

L'application principale pourra avoir la forme suivante (ou toute autre organisation offrant les mêmes possibilités). On note que le programme devra contenir un certain nombre de distributeurs déjà ajoutés à l'initialisation du programme.



Lors de la sélection de l'option, consommer des sucreries, après avoir sélectionné un distributeur, une boîte de dialogue est ouverte qui représente le distributeur de sucreries avec les fonctionnalités décrites précédemment. Un exemple de visualisation d'un distributeur pourrait être la suivante (ou toute autre représentation ayant les mêmes fonctionnalités).



Le nom des produits en « tête » de compartiment sont affichés, ainsi que le prix de la rangée de produits. Il est possible de sélectionner un produit (Lettre + Chiffre) et de saisir la somme à verser. Ensuite la boisson servie est décrite ainsi que

le rendu de monnaie. Il est possible d'ajouter de l'argent dans le monnaie et des sucreries dans un compartiment (des boîtes de dialogues seront ouvertes pour saisir les informations nécessaires).

Lors de la sélection de l'option, boire une canette, après avoir sélectionné un distributeur, une boîte de dialogue est ouverte qui représente le distributeur de canettes avec les fonctionnalités décrites précédemment. Un exemple de visualisation d'un distributeur pourrait être la suivante (ou toute autre représentation ayant les mêmes fonctionnalités). Des images des canettes disponibles, avec leurs prix et leurs quantités en stock sont affichées. Pour sélectionner une boisson, il est possible de cliquer sur les boutons (qui ont les noms des boissons). Ensuite la monnaie rendue et la description de la boisson servie sont affichées. Il est possible d'ajouter des l'argent dans le monnaie et des canettes (des boîtes de dialogues seront ouvertes pour saisir les informations nécessaires). Il est également possible d'avoir des informations sur le distributeur et le monnaie (sous forme textuelle)



Les images écran sont données à titre indicatif et toute autre organisation de l'application principale et des boîtes de dialogue est possible.

3. Rapport

Vous devez rendre un rapport (rédigé sur ordinateur !) qui doit rendre compte de votre travail. Celui-ci contiendra les informations suivantes :

Introduction

Cahier des charges : décrit la manière dont vous interprétez le sujet, ainsi que les fonctionnalités que doit avoir votre réalisation.

Modélisation Objet (I21)

- Description de la structure des classes (schéma) ainsi que du contenu des classes principales. Attention : il s'agit ici de montrer les principales fonctionnali-

tés mais pas de donner des listings exhaustifs. Vous pouvez mettre de tels listings en annexe. Précisez les relations d'héritage et d'utilisation entre classes.

- Description des algorithmes principaux. Par exemple ici celui qui permet de rendre la monnaie.

Interfaces visuelles (I22)

Le rapport devra contenir

- une description du contenu des différentes fenêtres et boîte de dialogue (ne pas donner les arborescences des composants). On expliquera clairement et en français les fonctionnalités disponibles dans ces fenêtres.
- une description des classes liées à l'interface (si vous ne les avez pas décrites dans la partie modélisation)
- une description détaillée des fonctionnalités importantes (mode de fonctionnement).

Le rapport devra comporter également un manuel utilisateur avec des images écran.

Bancs d'essais

Vous devez montrer comment fonctionne votre réalisation sur des exemples.

Conclusion

Décrivez l'état de votre réalisation par rapport au cahier des charges de l'introduction, ses limites et ce qui pourrait être amélioré.

Annexe

Vous placerez en annexe la javadoc de votre application ainsi que les éléments que vous souhaitez montrer et qui sont trop détaillés pour être exposés dans le corps du rapport (les listings de certaines classes par exemple).

4. Démonstration

La démonstration est destinée à présenter les principales fonctionnalités de votre réalisation. Il est important de faire attention aux points suivants :

1. Respectez le temps imparti.
2. Si vous avez réalisé votre programme sur votre ordinateur personnel, testez-le sur les ordinateurs de la faculté préalablement à la démonstration.
3. Ne vous perdez pas dans les détails techniques en partant de votre code. Structurez votre démonstration autour de jeux d'essai (une partie de ceux du rapport). L'enseignant vous posera des questions techniques s'il le juge utile.
4. Tous les participants au projet doivent prendre la parole de manière à peu près égale. Vous devez être le plus naturels possible lors des passages de parole.

En résumé, une démonstration doit être soigneusement préparée.

5. Échéances et conseils

La démonstration du projet aura lieu la semaine du 16 au 20 mai dans les créneaux de TPs. Vous devez apporter le rapport sous forme papier et sous forme de fichier (format pdf souhaité). Vous apporterez aussi un fichier compressé (extension zip) contenant le code du projet.

Vous pouvez ajouter d'autres distributeurs et/ou d'autres produits. Toutefois, ce n'est conseillé que si toutes les fonctionnalités de base sont bien réalisées et que vous êtes à jour dans les autres matières.

Il est (vivement) conseillé d'obtenir avant tout une version qui fonctionne, même avec des fonctionnalités plus restreintes que celles qui sont spécifiées par l'énoncé.