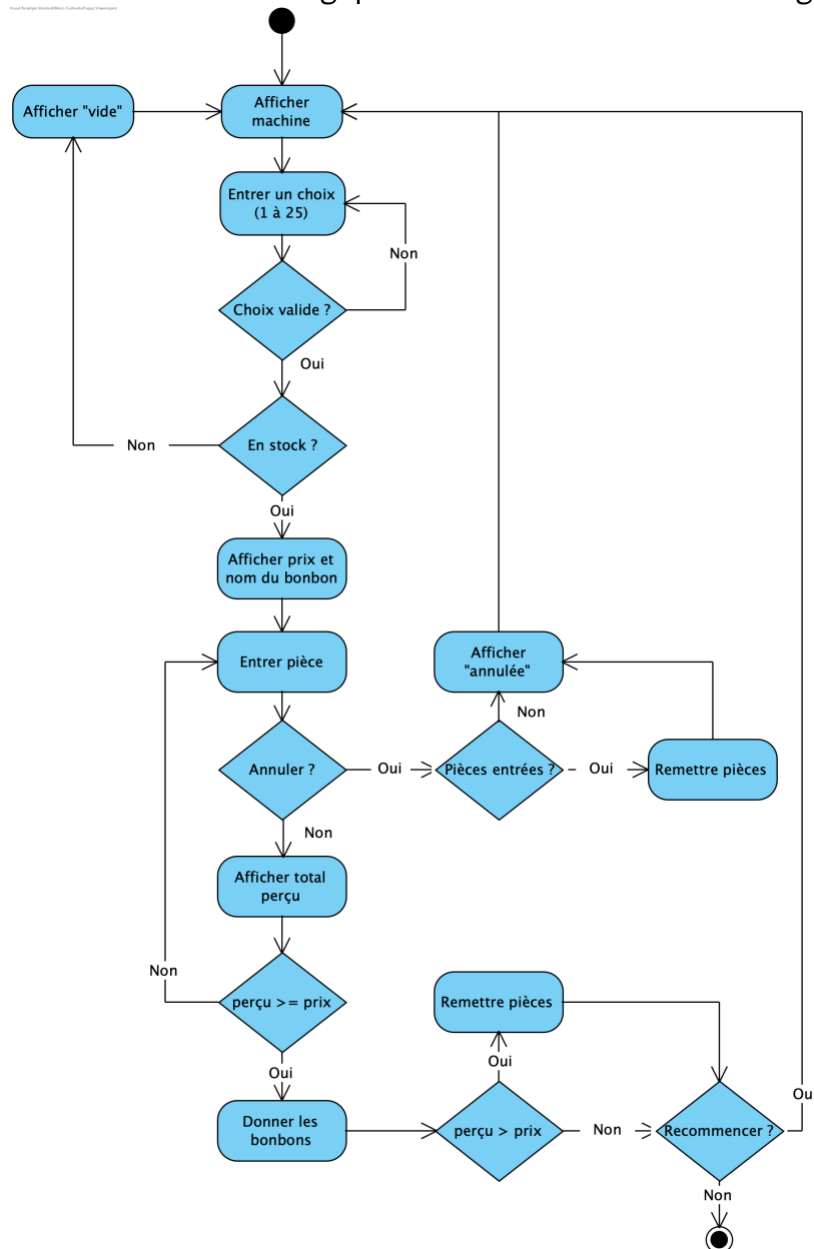


Énoncé

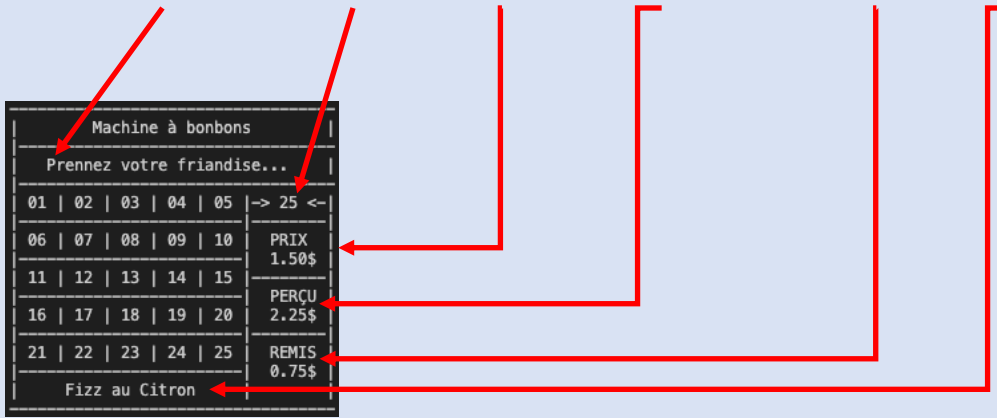
Vous êtes embauché par une entreprise en démarrage spécialisée dans la vente de machines distributrice de bonbons. Vous avez été sélectionné pour vos talents en programmation C# afin de programmer le logiciel se trouvant à l'intérieur de la machine.

Cédrik, un employé de l'entreprise a déjà programmé les fonctions d'affichages de la machine vous ne serez donc responsable que de la logique de la machine.

Cédrik a pris soin de bien documenter la logique de la machine dans un ordinogramme :



Clarifications de la structure du projet

Fichier / Dossier	Utilité
Projet.csproj	Gestionnaire des configurations et fichiers du projet
Program.cs	Emplacement principal où l'on devrait retrouver la logique de votre machine à bonbons. L'ensemble de votre code quoi !
Board.cs	<p>Classe programmée par Cédrik permettant l'affichage des données et messages sur la machine à bonbon. Vous n'avez <u>pas le droit</u> de modifier cette classe.</p> <p><code>Print(message, selection, price, received, returned, result)</code></p> 
Candy.cs	Classe simple permettant de sauvegarder un nom, un prix et une quantité pour chaque bonbon disponibles (25).
Data.cs	Classe incluant tout le code nécessaire à la lecture des Candy dans le fichier de données
candies.data	Fichier de données incluant la liste des bonbons disponibles.

Liste de fonctions obligatoires

Nom	Utilité
GetSelection()	Fonction permettant à l'utilisateur d'effectuer un choix entre 1 et le nombre de bonbons disponibles
GetCandy()	Fonction qui reçoit un numéro de sélection et qui retourne le Candy associé
GetCoin()	Fonction permettant à l'utilisateur de sélectionner une pièce de monnaie à insérer dans la machine à bonbons.
LoadCandies()	Fonction permettant d'initialiser la machine à bonbons avec les données provenant d'un fichier de données. Cette fonction sera dans la classe Data.cs

Le visuel attendu

1	<p>Lors du démarrage de la machine à bonbons, un écran sera visible à l'aide de la fonction <code>Machine.Print()</code>.</p> <p>Entrée sans paramètres cette fonction affichera la machine à bonbon telle que présentée à droite.</p>	<p>Message général</p>
2	<p>Une fois une sélection valide faite, le prix de la friandise s'affiche et il est maintenant possible d'insérer de la monnaie dans la machine.</p>	<p>Sélection effectuée</p> <p>Prix de la sélection</p>
3	<p>Une fois de la monnaie entrée dans la machine, cette dernière doit afficher le total du montant perçu et ce jusqu'à ce qu'il y ait suffisamment d'argent de perçu.</p>	<p>Montant perçu</p>
4	<p>Quand le montant reçu atteint le prix de vente, on voit apparaître la friandise dans la porte de sortie en bas. Si le montant reçu est plus grand que le prix alors on retourne le trop perçu.</p>	<p>Montant remis</p> <p>Friandise accessible</p>

Nom : _____

Critères de correction

#	Critère de correction	Points
1	Le projet s'exécute sans modifications	/3
2	Il est possible de faire un choix de bonbon (1 à 25 validé) à partir d'une fonction	/3
3	Un message d'erreur apparaît si la sélection est vide	/3
4	Le prix et le nom du bonbon s'affiche sur la machine	/3
5	Il est possible d'annuler l'opération en recevant l'argent donnée à la machine	/3
6	On voit augmenter l'argent perçu sur la machine (utilisation judicieuse d'un switch)	/3
7	Les bonbons sont automatiquement donnés quand l'argent perçu est suffisant	/3
8	Au besoin, l'argent perçu en trop est retournée à l'utilisateur	/3
9	Il est possible d'acheter d'autres bonbons (à l'infini)	/3
10	La structure du projet a été respectée	/3
11	Les fonctions obligatoires ont été correctement codées	/20
12	Le code est conforme aux spécifications, propre et structuré	/10
13	Attitude générale lors de la création du projet	/10
	TOTAL :	/60