Polytechnique Montréal

Automne 2019

Cours : Structures discrètes - LOG2810

TP2

AUTOMATES ET LANGAGES

Auteurs (Matricule) :

Jérémie Bédard (1952335)

Louis Plessis (1933334)

Yasmine Moumou (1962718)

Date de remise :

3 Décembre 2019

Introduction

Dans le cadre de ce travail pratique, le but principal consistait à implémenter un programme dans le contexte d'un entrepôt d'objets. Ce programme devait contenir un automate afin de faire de la complétion de mots dans un système de recherche d'objets, une interface graphique simple qui affiche un menu et un système de panier et de commande d'objets. Plus spécifiquement, pour faire la recherche d'objets, le programme devait permettre d'appliquer des filtres sur les caractéristiques des objets soit le nom, le code et le type afin d'afficher les objets correspondant à la recherche. De plus, le programme devait comprendre un système d'auto-complétions, c'est-à-dire, permettre à l'utilisateur d'écrire seulement les premières lettres (ou même laisser le champ vide) correspondant aux caractéristiques des objets recherchés et afficher au plus dix objets possibles correspondant aux filtres appliqués. Après une recherche, le programme devait proposer à l'utilisateur de mettre des objets dans un panier, de vider le panier, d'afficher le contenu du panier et de passer à la commande. La présentation des suggestions d'objets dans l'entrepôt devait tenir en compte que certains objets pouvaient ne plus être présent dû à une commande antérieure ou ne plus être disponibles dû à l'ajout des objets en question à un panier.

Solution

Nous avons séparé notre solution en quatre classes : Main, Automate, Objet, et Commande.

La classe Main se charge d'implémenter l'interface à travers la console à l'aide d'un switch case. Plus spécifiquement, elle affiche les options, lit les valeurs en entrée et choisit quelle fonction exécuter en fonction du choix de l'utilisateur en naviguant dans les différents états. Les états possibles comprennent menuPrincipal, faireSuggestion, menuSuggestion, ajoutPanier, viderPanier, passerCommande, afficherPanier et exit. La classe Main prend aussi le soin d'afficher des messages d'erreurs dans ces états si, par exemple, l'utilisateur entre une option inexistante, passe une commande trop lourde, etc. De plus, la classe main instancie un automate de la classe Automate et le crée à partir du fichier texte inventaire.txt et elle instancie une commande de la classe Commande.

La classe Automate comprend principalement une méthode qui crée l'automate à partir du fichier texte inventaire.txt. Essentiellement, cette méthode store les objets dans un ArrayList en lisant le ficher, c'est-à-dire elle store le langage des noms, des types et des codes que l'automate reconnaitra. De plus, la classe comprend une méthode qui retourne une liste de objets suggérés en fonction des filtres entrés par l'utilisateur et du contenu de son panier. Cette méthode est le cœur de l'automate. Elle contient une boucle qui parcourt la liste d'objets (représentant le langage reconnu par l'automate) et passe chacun de ses éléments dans une machine à état. La machine à état traite à l'aide de la liste d'objets, (les langages qu'elle reconnait), les filtres pour déterminer si elle reconnait ceux-ci. Dans le premier état, elle vérifie si le filtre appliqué sur le nom entre dans le lexique des noms, dans le deuxième état, elle vérifie si le filtre appliqué sur le type entre dans le lexique des types et dans le troisième état, elle vérifie si le filtre appliqué sur le code entre dans le lexique des codes. Le tout afin de vérifier que les trois mots entrés sont reconnus par l'automate. Si ce n'est pas le cas, aucune suggestions n'est affichées. Dans le cas contraire, les objets possibles

sont ajoutés aux suggestions seulement s'ils ne sont pas déjà dans le panier. Ensuite, cette méthode retourne la liste des suggestions. Finalement, la classe contient une méthode qui affiche les suggestions en numérotant les objets suggérés afin de permettre à l'utilisateur de choisir un objet à mettre dans son panier en indiquant le numéro de l'objet désiré.

Ensuite, la classe Objet comprend tout ce qui concerne les caractéristiques d'un objet de l'entrepôt soit son nom, son code et son type. De plus cette classe comprend une méthode qui affiche un objet. Le tout afin d'encapsuler la complexité liée aux objets.

En dernier lieu, la classe Commande se charge de contenir le panier de l'utilisateur (un ArrayList d'objets) et de gérer son contenu. Principalement, cette classe contient une méthode pour afficher le panier, le vider, calculer la masse total des objets contenant le panier et accéder au panier.

Difficultés rencontrées

Une des difficultés rencontrées au cours du développement de ce programme était de trouver une façon de retourner une liste d'objets suggérés en fonction des filtres entrés par l'utilisateur. Nous avons trouvé une façon simple de le faire en utilisant la méthode motA.startsWith(motB) des objets String qui renvoie un booléen vrai si le motA commence par le motB mis en paramètre (et qui renvoie toujours vrai si le motB est vide). À l'aide de cette méthode nous avons simplement pu construire une machine à état, parcourir tous les objets de l'inventaire et ajouter à la liste de suggestions l'objet courant lorsque les String des filtres nom, type et code étaient tous les trois reconnus par l'automate (si startsWith retournait vrai) et si l'objet n'était pas déjà dans le panier. Par ailleurs, une autre difficulté que nous avons rencontrée consistait à implémenter le retour au menu principal à partir de l'option d'ajouter un objet au panier. Nous n'étions pas certains de vouloir permettre à l'utilisateur de retourner au menu principale seulement après l'ajout au panier ou avant l'ajout au panier et juste après la sélection de l'option d'ajouter au panier. Nous avons finalement choisi d'ajouter les deux. Plus spécifiquement, il est possible de retourner au menu principal si l'option ajouter au panier a été sélectionnée par erreur (Le programme demande « Voulez-vous entrer dans ce mode (o/n) ? »). De plus, le programme retourne au menu précédant une fois l'ajout d'un objet complété. Le menu précédant offre l'option de retourner au menu principal. Le même principe s'applique pour la fonctionnalité de vidage de panier et de passage de commande.

Conclusion

En conclusion, au cours de ce travail pratique nous avons appris comment un système d'auto-complétions peut être implémenté. Nous avons appris à gérer les différentes erreurs qu'un utilisateur peut générer en interagissant avec un programme. De plus, nous nous sommes familiarisés avec la notion d'automate, de langages et les applications possibles des automates. Finalement, dans des projets futurs, nous aimerions apprendre à concevoir un programme avec une

interface autre que la console afin d’implémenter des applications plus professionnelles, présentables et plus facile à naviguer pour un utilisateur.