**Explication du code outil formel. Java**.

But de l’exercice :

On nous demande de créer un programme qui vérifie l’accès d’une personne, dans une entreprise

On se balance sous le système des automates. Dans ce cas, il nous faut identifier les états, l’État initial, les états finaux. De notre cas mes états sont au nombre de six états initial qui correspond à l’État de la carte ensuite pendant la vérification de la carte après la vérification du code , l’accès accepté, l’accès refusé, l’alarme déclencher et l’accès bloqué

Explications des différents états

* État initial : de la construction dans nos automates. Nous avons toujours un état initial. De notre cas. L’État initial correspond à l’état de la carte. En effet, pour pouvoir avoir accès à l’entreprise va falloir que la carte de l’utilisateur soit valide. Pour cela en va initialiser. L’état de la carte à l’état initial.
* Vérification carte :

La vérification carte se présente comme le passage de l’État initial. À la prochaine étape lorsque la carte a un état initial. La première étape ce sera de vérifier la carte si notre carte est valide. Alors on passe à la prochaine étape qui est la vérification du code. Dans le cas contraire, on aura un accès refusé

* Vérification du code :

On la spécifié auparavant que pour pouvoir passer à l’étape de vérification du code, il nous faut une carte valide . Au niveau de la vérification du code, nous avons deux possibilités soit le code est erroné soit le code est correct. Si notre code est erroné nous aurons un accès refusé. Dans ce cas, on parle de cartes, valide et code erroné entraîne accès refuser

De même Sina, code est correct, on parle de carte, valide et code correct entraîne accès accepter

* Accès accepter :

L’individu aura accès à l’entreprise dans le cas de système de vérification pour avoir accès à une entreprise on a un État accès accepté lorsque la carte est valide est le code est correct

* Accès refusé

Il y a deux possibilités pour avoir un accès refusé

Soit la carte est non valide, soit la carte est valide. Le code n’est pas valide soit la carte est en valide et le code est invalide.

* Accès bloqué

Lorsque le code est incorrect l’utilisateur à trois tentatives pour trouver le code correct à la première tentative, si le code entré est juste, on a un accès accepter tout en sachant que la carte est valide

Si la deuxième tentative, le qui est juste, il y aura accès

De même si la troisième tentative est juste, il aura accès

C’est-à-dire que pendant les trois tentatives, la première peut être fausse, la deuxième vraie, la première, et la deuxième peuvent être fausse et la troisième sera juste mais dans le cas où les trois tentatives sont fausses alors on a un accès bloqué

Si durant la première tentative, le code est juste, les deux tentatives vont s’afficher de telle sorte, A, ce qu’il puisse remettre le code pour confirmation

* Alarme, déclenchée

L’alarme est déclenchée dans notre cas, lorsque la carte est refusée et que toutes les tentatives du code sont fausses. Dans ce cas, l’alarme est déclenchée.

**Explication des différentes classes du programme**

J’ai créé quatre classes. La classe automate. La classe carte la classe Main automate et la classe spécification

Pour s’accorder avec les différents niveaux énoncés dans l’exercice

* L’étape qui est la spécification du système avec la logique formelle correspond à la classe spécification

Dans cette classe, j’ai créé une méthode :

* Contrôle accès qui prend un paramètre. La carte valide de type boolean et code correct de même type le but de cette méthode aide de pouvoir contrôler les différents états entre les transitions

Exemple, prenons le cas de notre carte valide pour pouvoir passer au prochain état, on va contrôler l’accès de ce fait, la méthode contrôle accès aura carte valide , et code correct à faux ce qui va activer l’alarme

Donc le but de cette méthode c’est de vérifier d’abord l’état de la carte. Ensuite si l’état de la carte est égal à l’État obtenu lors de la vérification de la carte. Notre programme va passer à la prochaine étape qu’elle a vérification du code Dans le cas contraire, il va activer l’alarme.

Ensuite, j’ai une méthode :

* getEtat

Cette méthode comprend les différents états que peut avoir notre automate, soit accès accepté, qui retourne, accès accepté, soit accès refusé qui retourne assez refusé, soit déclencher qui retourne à l’alarme déclencher, soit accepté ,bloqué qui retourne, accepte, bloqué. Dans le cas où l’un des états n’appartient pas aux états énumérés on affiche inconnu, veuillez déclarer l’identité

* Classe carte:

Cette classe n’a pas été demandé dans l’exercice mais je l’ai créé pour avoir une idée de l’exemple de cartes que je veux avoir pour les différentes personnes elle comprend le nom, le prénom, le matricule qui va être utilisé notre main plus tard

* Classe automate :

Cette classe correspond à l’étape deux qui est modélisation du système avec des automates et langage formel

Les attributs de cette classe sont les différents états de l’automate, c’est-à-dire état initial. Vérification carte, vérification, code accès, accepté, accès, refusé, alarme, déclencher accès bloqué qui sont des types état.

On a également tentative qui est de type entier état de la carte qui est de type état et Check de type chaîne de caractère

Elle comprend la méthode. Vérification carte, Vérification code, activer alarme.

* Au niveau de la méthode, vérification carte

Cette méthode, prends en paramètre carte valide de type Boolean

Si l’état de la carte est à l’état initiale, on va vérifier la carte si la carte est valide. Alors l’état de la carte va passer à la prochaine étape et va prendre l’état. Vérification code et va afficher carte validée

La variable check va garder carte validée pour plus tard

* Méthode vérification code : il prend en paramètre le code correct

Ici, l’état de la carte est à l’État Vérification code. Si le code est correct, on va afficher accès accepté. Dans le cas contraire, l’utilisateur va devoir entrer un nouveau code, en fonction des tentatives possibles si les tentatives ont été utilisés et qu’aucune des tentatives n’est juste alors l’accès est bloqué .

* Méthode activer alarme

Si l’état de la carte est un accès refusé et que les tentatives ont toute été utilisé, l’état de la carte prend alarme déclencher

* La dernière méthode getEtat, retourne l’état de la carte
* Classe Main automate

C’est la classe principale qui va permettre d’avoir des échanges avec l’utilisateur.