Programmation Orientée Objet

TP

v0

IUT d'Annecy, 9 rue de l'Arc en Ciel, 74940 Annecy

POO sur CodeSys

1 Introduction

Dans le prochain TP, nous nous intéresserons à la classe CPont. Cette classe regroupe un chariot (CChariot) et un crochet (CCrochet) qui ont été vus dans les TP précédents. La classe CPont doit définir des missions de plus haut niveau qui encapsulera les missions du chariot et du crochet.

Notion d'agrégation

En UML, on dit que la classe CPont est une agrégation de CChariot et de CCrochet.

Pour rappel, une agrégation est une relation entre deux classes. Elle signifie que la classe agrégat (ici CPont) contient une ou plusieurs instances de la classe agrégée (ici CChariot et CCrochet). Les classes agrégées peuvent, toutefois, exister indépendamment de la classe agrégat.

2 Etude préliminaire

On désire se donner la possibilité de traiter plusieurs pièces en parallèle dans des cuves différentes avec des temps de trempes différents. On désire également pouvoir éventuellement enchaîner des traitements dans plusieurs cuves avant de livrer la pièce.

On désire également pouvoir optimiser les temps de déplacement. Cela signifie, par exemple, qu'il devra être possible de déplacer le pont, chargé ou non, vers une cuve afin d'anticiper un prochain chargement ou déchargement.

2.1 Missions du pont

Exemple de cas d'usage : Traitement d'une pièce dans la cuve 2 Pour traiter une pièce dans la cuve 2, le pont devra :

- Charger une pièce depuis la cuve d'entrée 0E
- Décharger une pièce dans la cuve 2
- *Attente de la fin du traitement (d'autres missions pourront être effectuées pendant cette attente)*
- Charger la pièce traitée depuis la cuve 2
- Décharger la pièce dans la cuve de sortie 0S

Afin d'optimiser les temps de déplacement, le pont pourra aussi, éventuellement :

- se déplacer vers la cuve 1 dans l'attente d'une pièce disponible en cuve 0E.
- se déplacer vers la cuve 2 dans l'attente de la fin du traitement de la pièce en cours.
- se déplacer au-dessus de la cuve 3 dans l'attente de la libération de la cuve de sortie.

Activité 1: Missions du Pont

Question 1 Proposer des missions que devra pouvoir réaliser le pont.

Question 2 En étudiant ces missions, proposer une hiérarchie les reliant. Autrement dit, quelles missions peuvent préempter quelles missions ?

Question 3 À partir des questions précédentes, proposer un type énuméré pour les missions du pont. Ce type sera, au choix, paramétré ou non.

2.2 Etats du pont

Activité 2

Question 4 En justifiant, à partir de l'étude précédente, proposer des états pour le pont.

Question 5 Proposer un type énuméré pour les états du pont. Ce type sera, au choix, paramétré ou non.

2.3 Implémentation des missions

Activité 3

Question 6 Proposer une méthode `MMissionIn` pour prendre en compte la mission reçue par le pont. *À partir de l'état du pont et de la mission reçue, cette méthode devra dicter les actions à réaliser par le chariot et le crochet. Ces ordres seront stockés dans des variables internes à la classe CPont.*

Question 7 Proposer une implémentation de la méthode `MMissionOut` pour affecter les missions au chariot et au crochet. *Cette implémentation devra respecter le paradigme d'une machine de Mealy.*