

Polytech Paris-Saclay – ET4 Informatique – Compléments Objets

On considère un laboratoire dans lequel des chercheurs évaluent des stratégies de déplacement de robots. Les expériences sont faites dans une pièce (définie par une longueur et une largeur) contenant des objets, caractérisés par un identifiant et leur position dans la pièce. Certains de ces objets sont inertes, comme les caisses de bois, d'autres sont mobiles, comme les robots. Un robot peut soulever et transporter des objets de tout type, y compris d'autres robots portant eux-mêmes des objets, et les transporter à travers la pièce sous réserve que la masse totale des objets portés, directement ou non, soit inférieure à une charge maximale définie pour ce robot. Un robot peut par exemple porter directement deux caisses et un autre robot, ce dernier portant lui-même d'autres objets. Il n'y a pas de limitation sur le portage autre que la contrainte sur la masse totale des objets portés et le fait que seuls les robots situés directement au sol peuvent charger ou décharger un objet. Un objet chargé doit l'être à partir du sol : on ne transfère pas directement entre robots des objets qu'ils portent.

1. Décrire une hiérarchie de classes permettant de modéliser le problème.
2. Écrire une méthode pour calculer la masse totale d'un robot et des objets qu'il transporte.
3. Écrire une méthode pour lister tous les objets de la pièce. Pour chacun, on donnera son identifiant, sa position et sa masse (y compris celle des objets transportés pour les robots). Pour un robot on donne en plus le nombre d'objets qu'il transporte directement ou non.
4. Écrire les méthodes de chargement et déchargement d'un objet.
5. Écrire une méthode `deplacer(dx, dy)` pour déplacer un robot posé au sol. Si le robot n'est pas au sol, une exception est levée.

On considère maintenant un nouveau type d'objets : les sacs de sable. Un sac de sable porté par un robot en mouvement (qu'il soit au sol ou non) perd une portion du sable qu'il contient, proportionnellement à la distance sur laquelle il est transporté.

6. Modifier la hiérarchie proposée pour intégrer cette nouvelle contrainte.
7. Modifier si besoin la méthode `deplacer`. Le code que vous aviez écrit pour calculer la masse d'un robot est-il toujours correct ?

On veut maintenant ajouter un superviseur, chargé de détecter certains événements. On demande que le superviseur affiche automatiquement un message à chaque fois qu'un objet est posé sur le sol ou soulevé du sol. Tout objet est supposé posé au sol à sa création et on prévient le superviseur de l'existence de ce nouvel objet

8. Modifier la hiérarchie proposée pour intégrer cette possibilité. Indiquez le protocole de communication entre les objets et le superviseur. Précisez les méthodes et constructeurs impliqués.
9. Généraliser votre solution pour autoriser l'existence de plusieurs superviseurs s'intéressant à des objets ou des événements différents (un objet peut être sujet à plusieurs supervisions, par exemple certaines liées à sa position verticale, d'autres à ses déplacements, ou à des modifications de masse, etc). On doit pouvoir ajouter/retirer des objets de la collection des objets supervisés par un superviseur donné. On doit pouvoir facilement ajouter un nouveau type de supervision.