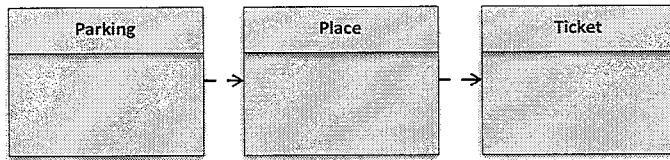


Première étape

Un parking urbain comporte un certain nombre de places numérotées de 1 à la capacité du parking (place 54 par exemple) qui peuvent être occupées ou pas par des véhicules. Un véhicule est représenté par un ticket, qui contient son numéro d'immatriculation (par exemple, "CF 782 TM") et son horaire d'arrivée (exemple : 14h45). Une place occupée est associée au ticket du véhicule concerné ou à rien dans le cas contraire.

Pour gérer cette situation en java, on utilise les classes Parking, Place et Ticket (voir l'extrait de diagramme ci-dessous ou en annexe).



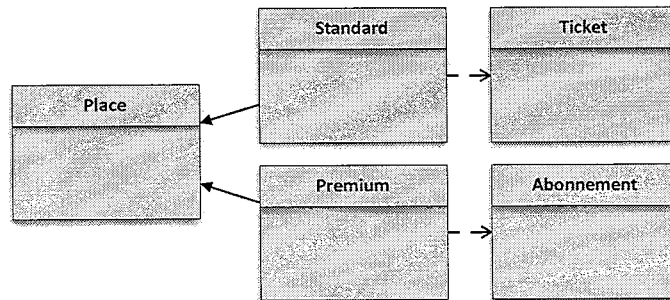
Questions

1. Complétez le diagramme de l'annexe en écrivant pour chaque classe les déclarations des attributs nécessaires. Les groupes/collections peuvent être symbolisés par des tableaux et les horaires d'arrivée par deux entiers.

Pour répondre aux questions suivantes, vous considérerez que les attributs sont dotés des accesseurs nécessaires.

2. Écrivez la méthode `isLibre(...)` de la classe `Place` qui indique le statut d'une place, c'est-à-dire si elle est libre (sans véhicule associé) ou occupée.
3. En supposant que tous les attributs soient dotés des accesseurs nécessaires, écrivez la méthode `getLibres(...)` de `Parking` qui calcule le nombre de places libres du parking en fonction du statut des places.

Deuxième étape



Les gestionnaires du parking veulent réserver certaines places à des clients abonnés. Pour ce faire, on distingue désormais deux types de places : les places *standard*, identiques aux places de la première étape, qui sont associées chacune à un ticket si elles sont occupées, et les places *premium*, qui sont associées chacune à un *abonnement* si elles sont louées et à rien dans le cas contraire. Les informations d'un abonnement sont les *nom*, *prénom* de l'abonné, le *numéro d'immatriculation* de son véhicule et la *date* de fin d'abonnement (on supposera qu'il existe une classe `Date` directement utilisable).

Questions

1. Ajoutez au diagramme de l'annexe les déclarations des attributs des différentes classes après cette réorganisation, sachant que les places ont un *type*, de valeur "standard" ou "premium".
Pour la suite, comme pour la première étape, vous considérerez que les attributs sont dotés des accesseurs nécessaires.
2. À l'entrée du parking, un afficheur doit indiquer la première place disponible — par ordre croissant des numéros. Sachant que les places premium ne peuvent pas être utilisées pour un stationnement temporaire, écrivez la méthode `getPremiere()` qui restitue la première place standard disponible.
3. Toutes les places standard occupées ou premium louées sont associées indirectement à un numéro d'immatriculation. Il figure, soit sur le ticket associé à une place standard, soit, pour une place premium, dans les informations de l'abonnement correspondant.
Écrivez la méthode `getImmatriculation(...)` des classes `Place`, `Standard` et `Premium` qui permet de restituer, soit ce numéro, soit la valeur `null` si la place n'est pas occupée ou louée.
4. Écrivez la méthode `toString()` pour les instances suivantes :
 - a. instances de `Ticket` : elle doit restituer le numéro minéralogique de la voiture correspondante et l'heure d'arrivée. Par exemple : "CF 782 TM (arrivée : 14h35)".
 - b. instances de `Abonnement` : elle restitue les prénom, nom de l'abonné, le numéro minéralogique de sa voiture, et la date de fin de son abonnement. Par exemple : "Joël Savelli, CF 782 TM (Fin : 3/12/2021)".
 - c. instances de `Place`, elle doit restituer leur numéro (place 24), leur type et leur statut :
 - places standard : *libre* ou les caractéristiques du ticket. Par exemple : "place 24 (standard) : libre" ou "place 67 (standard) : CF 782 TM (arrivée : 14h35)"
 - places premium : *À louer* ou les caractéristiques de l'abonnement.
"place 4 (premium) : À louer" ou "place 2 (premium) : Richard Genestier, AF 212 PT (Fin : 12/07/2021)"
 - d. instances de `Parking` qui doit restituer les caractéristiques des différentes places :
 - place 1 (premium) : à louer
 - place 2 (premium) : Joël Savelli, CF 782 TM (Fin : 3/12/2021)
 - place 3 (premium) : à louer
 - place 4 (premium) : Richard Genestier, AF 212 PT (Fin : 12/07/2021)
 - place 5 (standard) : AM 620 CT (arrivée : 14h13)
 - place 6 (standard) : libre
 - place 7 (standard) : GL 483 AZ (arrivée : 12h23)
 - ...

Annexe à rendre

Diagramme Première étape

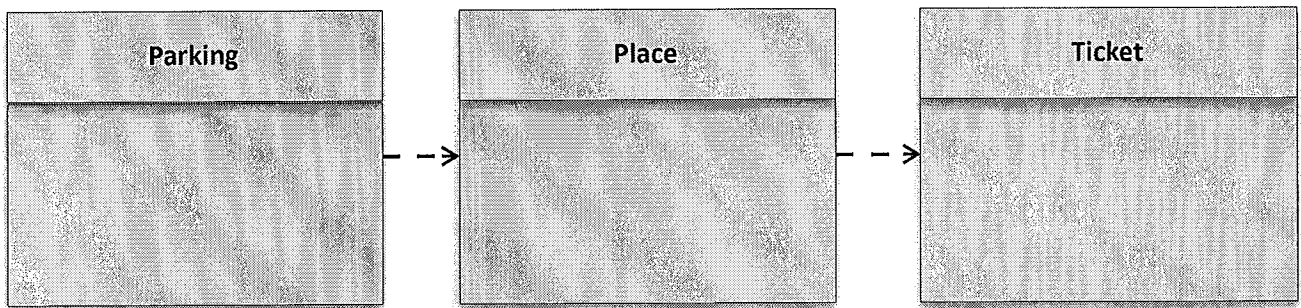


Diagramme deuxième étape

