

TP01

1. École

L'objectif est de créer une classe « école » afin de mettre en pratique vos connaissances fraîchement acquises sur les classes.

L'exercice est séparé en 3 parties :

- Les variables d'instances
- Le constructeur
- Les méthodes de classe
 - Accesseurs et Mutateurs

Voici le détail de la classe représentant une école. Une école possède les champs suivants :

- Nom
- Adresse
- Type (Publique / Privée)
- Nombre de bâtiments
- Degré (Primaire, C.O. ou Universitaire)
- Domaine (Général, Informatique, Gestion, ...)
- Nombre d'étudiants

Cette classe doit posséder les méthodes suivantes :

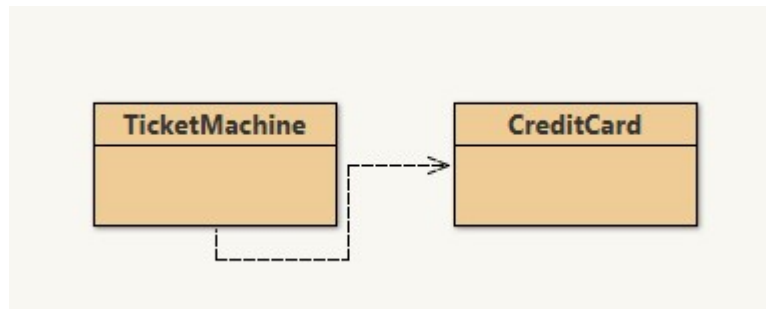
- Changer le nombre de bâtiments par une valeur donnée
- Changer le nombre d'étudiants par une valeur donnée
- Une méthode affichant :
 - Un affichage complet :
 - Nom de l'école, adresse
 - Ecole "type" de degré xxxxx dans le domaine yyyyyy
 - Il y a X bâtiments pour Y étudiants
 - Un affichage partiel :
 - Nom de l'école et adresse
- Changer le degré par une valeur donnée :
 - Si l'école est publique
 - Si le degré est Primaire ou C.O., le domaine est obligatoirement Général

Exemples de sorties :

```
Haute École de Gestion, Rue de la Tambourine 17
Ecole Publique de degré Universitaire dans le domaine Informatique, Gestion, Economie, Management
Il y a 5 bâtiments pour 800 étudiants
```

2. Better Ticket Machine

Il vous est demandé d'implémenter le paiement par carte de crédit à la TicketMachine vu pendant le cours d'algorithmique. Pour faire cela, il vous faudra créer une nouvelle classe « CreditCard ».



Dans le cadre de cet exercice, une carte est définie par un code d'identification de la carte, un code pin et un montant se trouvant sur cette carte. Le code d'identification est un grand nombre séparé par des " – " (par exemple « 4950-1234-5678-9012 »). Ce qui identifie une carte Visa ou Mastercard est les 4 premiers chiffres avec respectivement 4950 pour Visa et 5468 pour Mastercard. Vous comparerez donc ces 4 chiffres (sous forme d'un int) avec les constantes à votre disposition dans TicketMachine.

La classe CreditCard doit pouvoir vérifier le code pin entré par l'utilisateur sur la TicketMachine. Cette machine ne prend que les cartes « Visa » et « MasterCard » en charge. Un message est affiché pour n'importe quelle autre carte.

Le reste du programme doit rester fonctionnel ! De plus, le paiement doit être fonctionnel via la carte de crédit (le ticket doit « s'imprimer »).

Deux fonctions seront donc à implémenter : Le paiement par carte et l'entrée du code pin par l'utilisateur.

Exemples de sorties :

Créer une instance de TicketMachine

VotreFonctionEntrerCodePin() (Changez ce nom par celui de votre choix)

VotreFonctionPaiementParCarte() (Changez ce nom par celui de votre choix)

Si la carte n'est pas une Visa ou une MasterCard

`We only accept "Visa" and "MasterCard".`

Si le code pin n'est pas correct

`The pin you entered is incorrect`

S'il n'y a pas assez d'argent sur la carte :

`Not enough balance on this card.`

PrintTicket()

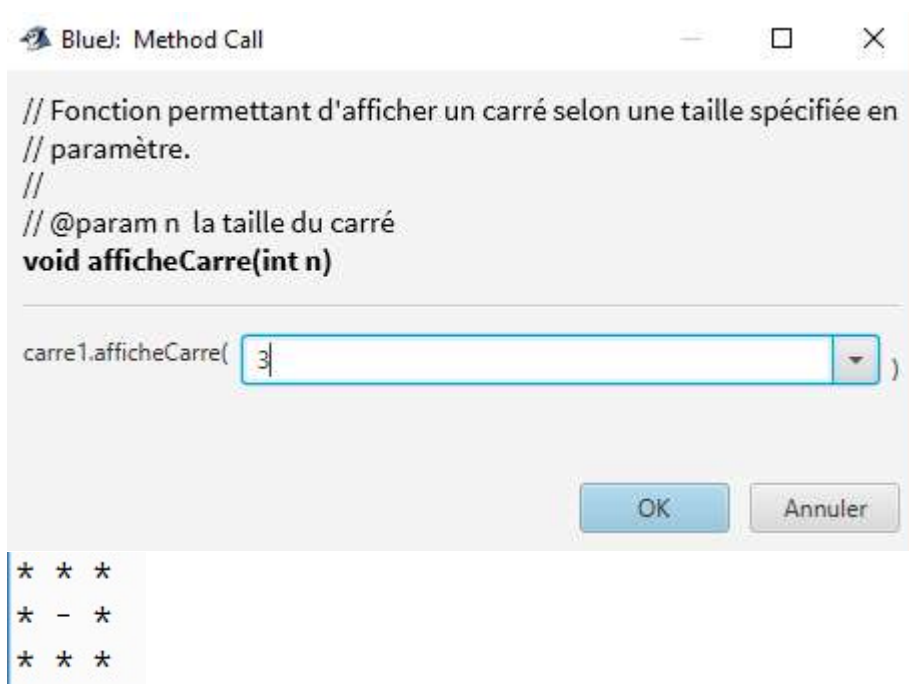
Si le code pin est correct, que la carte a le bon début d'identification (Visa ou Mastercard) et qu'il y a assez d'argent sur la carte :

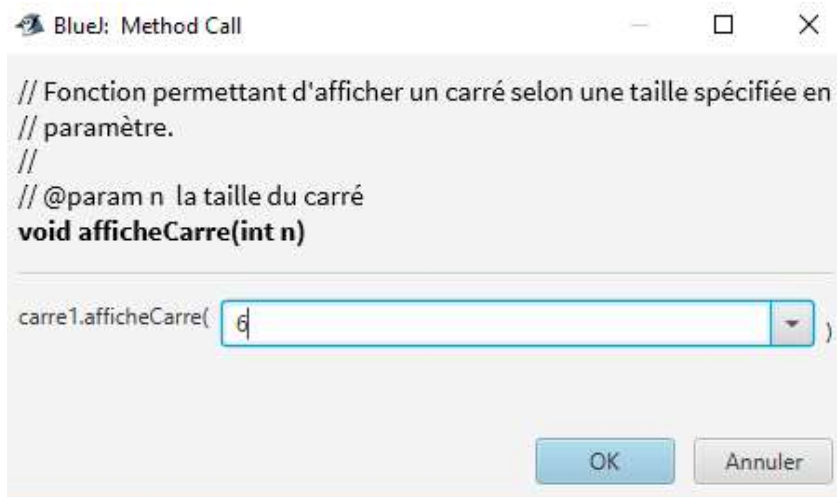
```
#####  
# The BlueJ Line  
# Ticket  
# 100 cents.  
#####
```

3. Carré

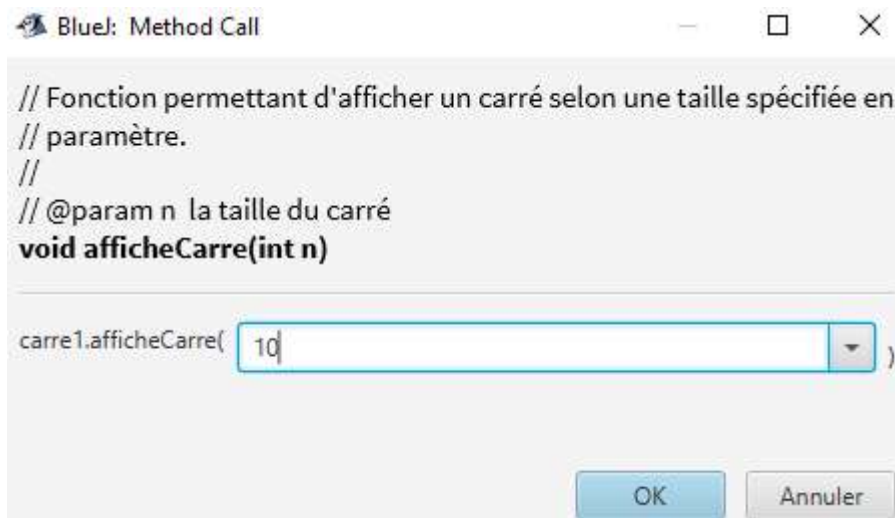
Ecrire un programme permettant l'affichage d'un carré dans la console (voir exemples de sorties). La taille du carré est paramétrable.

Exemples de sorties :





```
* * * * *
* - - - - *
*           *
*           *
* - - - - *
* * * * *
```



```
* * * * *
* - - - - *
*           *
*           *
*           *
*           *
*           *
*           *
* - - - - *
* * * * *
```