

Langages de programmation

TP 12

Objectifs

- Les classes abstraites
- Méthodes abstraites

Théorie

Les classes abstraites

- ▶ Les classes abstraites ne peuvent être instanciées (avec «new»).
- ▶ Ils ont des sous-classes avec des exemples concrets
- ▶ Une classe est considérée comme abstraite si au moins une méthode dans la classe est abstraite (n'a pas d'implémentation).
- ▶ Une méthode abstraite n'a pas de mise en œuvre.
- ▶ Une classe abstraite peut contenir des variables d'instance et des méthodes qui sont mises en œuvre.
- ▶ Une sous-classe doit implémenter toutes les méthodes abstraites de sa superclasse (pour que l'objet de la sous-classe puisse être instancié).

Théorie

Importance

- ▶ Réutilisation du code
- ▶ Contraintes pour l'implémentation des méthodes abstraites (dans les différentes sous-classes)
- ▶ Flexibilité et maintenabilité

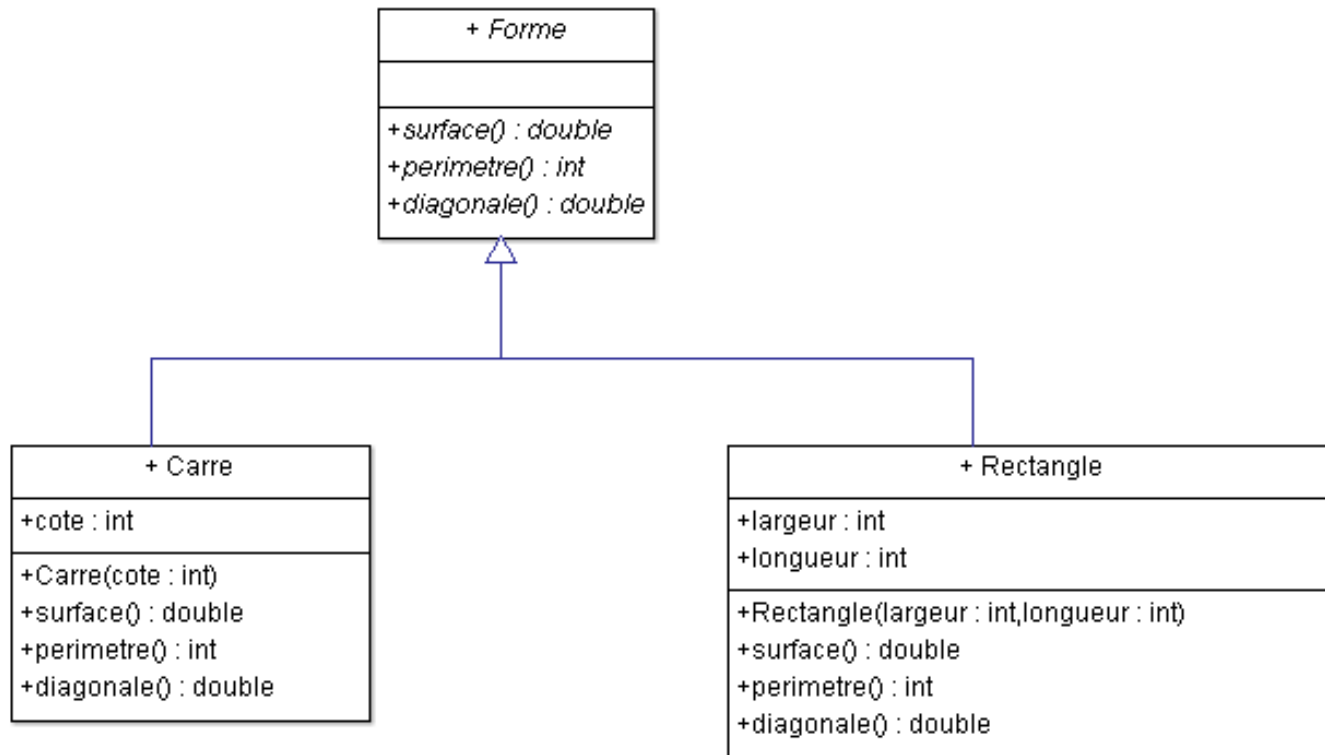
Example:

```
public abstract class AAA{  
  
    .....  
  
    public abstract void aaa (); // pas de corps  
  
}
```

Vous ne pouvez pas appeler le constructeur d'une classe abstraite!

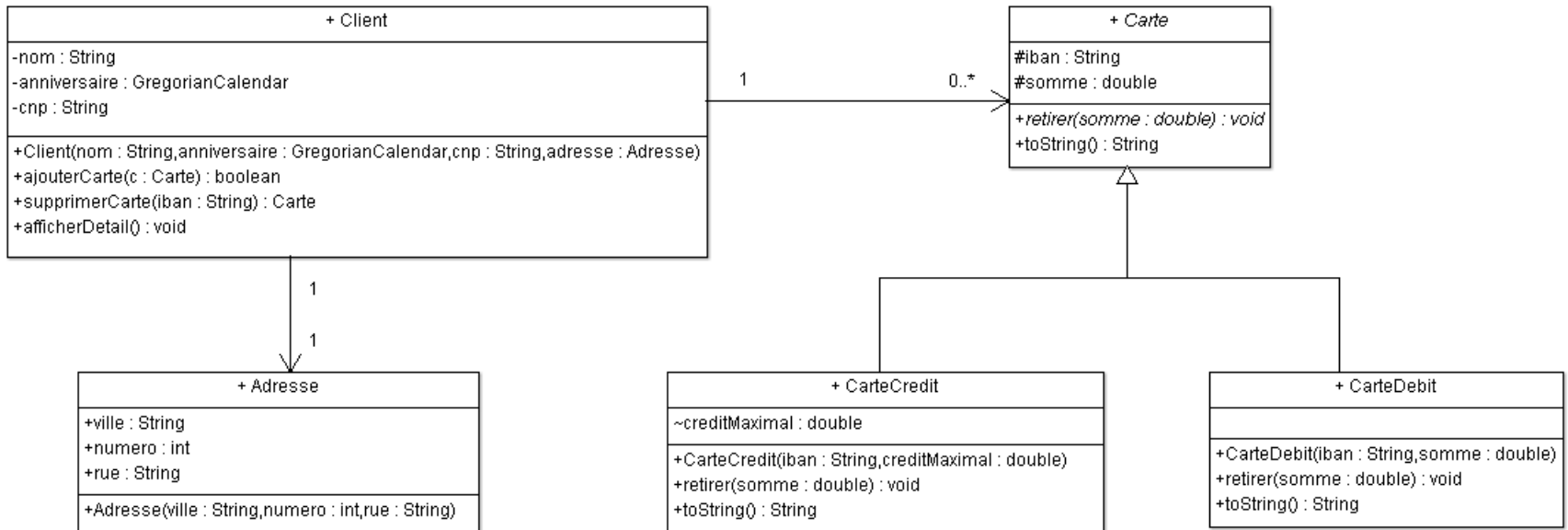
Problèmes en classe

1. Mettez en œuvre les classes du modèle.



Problèmes en classe

2. Mettez en œuvre les classes du modèle.



Devoir

1. Le propriétaire d'un magasin doit faire une statistique concernant le rapport entre le prix (en lei) de deux sortes de ses produits (cartes graphiques et moniteurs) et la performance de ces produit. Dans le cas de la production, le critère de performance est donné par le nombre de points obtenus dans 3DMark. Les moniteurs sont évalués en fonction de la mise à jour obtenue à une résolution de 1152 x 864. Selon cette note, la performance (sauvegardée comme un nombre réel) est calculée après la formule suivant:

$$\text{performance} = \text{Score obtenu} / \text{maximumScore} * 100,$$

HardwareProduit est une classe abstraite, ayant calculerPerformance() et calculerPrixLei() comme méthodes abstraites. En outre, cette classe contient les variables privées monnaiePrix, score, leiPrix et performance. La valeur de la variable leiPrix sera calculée, en fonction de la devise le prix et le type qui est utilisé. Nous supposons que le prix de la carte vidéo est donné en € et pour les moniteurs en \$.

Devoir

Taches:

- A.** Mettre en oeuvre la classe à partir du diagramme.
 - B.** Afin de tester les classes, implémentez la classe TestShop avec une méthode principale qui effectue les tâches suivantes:
 - a.** Créer un *array* (ou ArrayList) de produits ProduitHardware[], en utilisant les données provenant des fichiers:
CartesGraphiques.txt Moniteurs.txt
130 76 249 78
108 77.79 224 95
152 72.55 249 96
53 54.69 279 107
 - b.** Calculer la performance de chaque produit de l'*array*.
 - c.** Imprimer les informations pertinentes pour chaque objet de l'*array*.
- Utilisez le diagramme de classes UML suivant associé à l'application:

Devoir

