

Rappel : signatures de méthodes

Lors de la définition de classes, nous sommes amenés à définir un ensemble d'attributs typés, et des méthodes permettant d'effectuer un traitement sur des données et / ou de modifier la valeur des attributs.

Lors de la définition d'une méthode, il faut prêter attention à divers aspects :

- le nom de la méthode,
- le type et le nom des informations passées en entrée de la méthode (*i.e.* les nom et type de ses arguments),
- le type de l'information retournée par la méthode (type de retour),
- (éventuellement) la visibilité de la méthode (publique, privée, etc),
- le genre de méthode, c'est-à-dire méthode de classe (statique) / d'instance (dynamique).
Une méthode de classe peut-être appelée indépendamment de la construction d'un objet de la classe.

Ces informations sont regroupées dans ce que l'on appelle la signature de la méthode, dont le format est le suivant :

```
[Visibilité] [Static] Type-de-retour Nom (Type-arg1 Nom-arg1 , Type-arg2  
Nom-arg2 , ...)
```

Prenons l'exemple de la méthode principale (point de départ du programme), dont la signature s'écrit (en Java) :

```
Public static void main(String[] args)
```

Cette signature nous informe que :

- la méthode s'appelle `main`,
- elle ne retourne aucune valeur (type de retour `void`),
- elle prend en entrée un tableau de chaînes de caractères (`String[] args`) appelé `args`,
- cette méthode est une méthode de classe, et ne nécessite pas la construction d'un objet pour être utilisée (mot-clé `static`).

Exercice 1 : la classe `AutomateBancaire` (suite)

Dans cet exercice, vous allez reprendre la classe `AutomateBancaire` du TD précédent. Nous rappelons qu'un automate bancaire est une machine permettant de créditer / débiter d'une certaine somme d'argent certains comptes bancaires. Dans le TD précédent, vous avez vu comment définir la classe `AutomateBancaire` (qui utilise la classe `CompteBancaire`). Il reste à définir la classe principale permettant de construire un automate et de l'utiliser. Nous souhaitons que l'automate soit construit et qu'il se mette en mode *en service*. Dans ce mode, l'automate propose un menu textuel à l'utilisateur, par exemple :

Bonjour, veuillez saisir votre numéro de compte:

Une fois le numéro de compte saisi, l'automate propose un service, par exemple via le menu suivant :

Choisissez une opération (tapez 0 pour quitter):

1. Créditer votre compte
2. Débiter votre compte
3. Afficher les informations de votre compte

Exercice 2 : librairie d'opérations sur les chaînes de caractères

Dans cet exercice, nous vous demandons de définir une librairie d'opérations agissant sur les chaînes de caractères. Pour cela, vous allez définir la classe `Chaines` comportant des méthodes implantant chacune une opération sur une chaîne de caractères. Les opérations sur les chaînes qui sont attendues sont les suivantes :

- vérifier si la chaîne est un palindrome (un palindrome est une chaîne qui se lit dans les deux sens, exemple : "tapat"),
- crypter la chaîne d'entrée au moyen du procédé suivant :
 - pour crypter une chaîne nous prenons chaque caractère, lui associons un nombre entier compris entre 1 et 26 (a valant 1, etc), puis nous décalons les chiffre de 10 unités, et retranscrivons en caractères (2 devient b, etc).
- décrypter une chaîne ayant été cryptée au moyen du procédé ci-dessus.

Implantez et tester chacune de ces opérations.

NB Vous prendrez soin de définir les signatures de vos méthodes de manière cohérente (par exemple, il doit être possible de composer les méthodes de cryptage et décryptage).