Série 1

Cette série met en application les concepts de classe, d'objet, d'abstraction et d'encapsulation vu l'an dernier. Le code Java est à écrire en anglais, en respectant les conventions de nommage. Des commentaires en français sont autorisés.

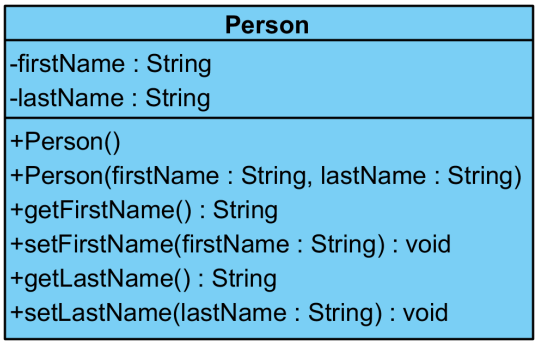
1. Veuillez créer un projet Java nommé « *Serie1* » avec l'EDI IntelliJ

    a. La classe qui contient la méthode « main » peut s'appeler « App »

2. Veuillez créer le package suivant « *ch.hearc.ig.sda.business* ». Ce package comprendra les classes métiers.

    a. Pourquoi séparer les classes dans des packages/paquetages ?

3. Veuillez définir la classe « *Person* » dans le package « *business* » avec les attributs et méthodes suivants :
    a. La classe est « *public* »



4. Définition, déclaration et instanciation

    a. Initialiser et instancier une première personne, par exemple :
        i. *Person person1 = new Person("Alan", "Kay") ;*

    b. Veuillez déclarer « person2 », et lui affecter « person1 »
        i. Par exemple :

            Person person2;

            person2 = person1;

    c. Veuillez schématiser cette situation avec les cases mémoires.

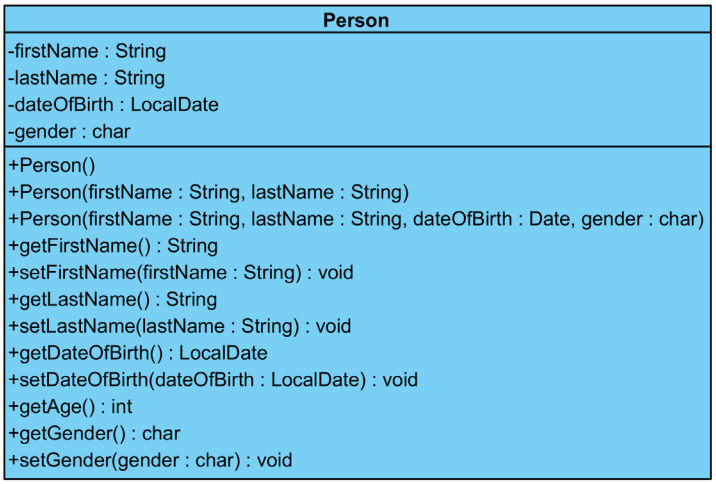
    d. Comparez ces objets avec les « == » et equals()

    e. Que constater-vous ?

5. Créer la méthode *clone()*, qui effectue une *copie superficielle* d'un objet de type « Person ».

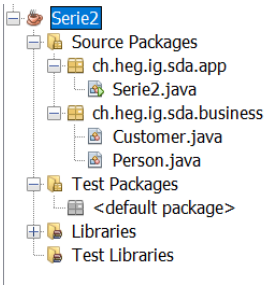
6. Lire le point 2 « Java et la programmation orientée objet » du chapitre « 1. Présentation de Java » du livre de Delannoy (2008)

7. Veuillez modifier la classe « *Person* » du package « *business* » avec les attributs et méthodes suivants :



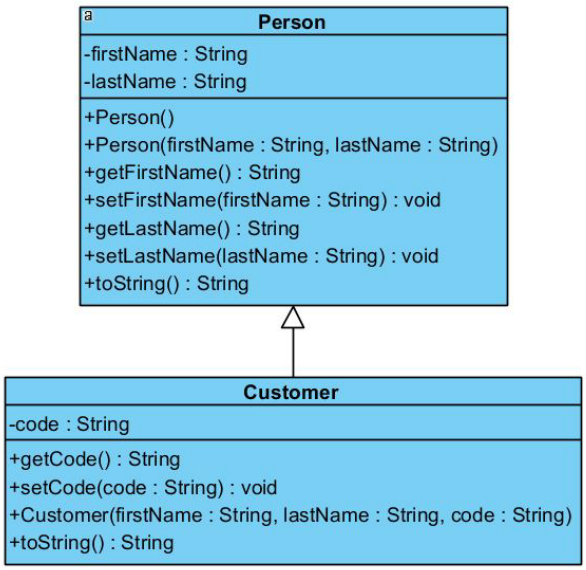
    a. Le type LocalDate provient de « *java.time.LocalDate* »

    b. A quoi sert un « import » dans Java ?



8. Créer une méthode statique dans la classe « App » qui déclare, instancie et affiche une personne dans la console

9. Créer la classe « Customer », qui est une sous-classe de la classe « Person »
    a. Réutiliser le constructeur de la classe Person dans celui du Customer



10. Redéfinir la méthode « toString() » des classes « Person » et « Customer », en s'appuyant sur la classe « StringBuilder »
    a. Afficher une personne avec le message « Je suis une personne… » et un client avec le message « Je suis un client… », avec les attributs respectifs

    b. Observer les résultats dans la console

11. Mise en pratique des tableaux

Si nécessaire, revoir la théorie sur les tableaux (array)

a. Créer une méthode statique dans la classe « Serie2 » qui déclare et initialise un tableau de 3 personnes
   i. Créer 3 personnes et ajouter-les dans le tableau

b. Créer une méthode qui calcule l'âge moyen des personnes en utilisant une boucle « for »
   i. Afficher le résultat dans la console

c. Créer une méthode qui retourne la personne la plus jeune en utilisant une boucle « while »
   i. Afficher le résultat dans la console

d. Créer une méthode qui affiche la personne la plus âgée en utilisant une boucle « do while »
   i. Afficher le résultat dans la console

e. Modifier la méthode qui initialise le tableau de 3 personnes, pour initialiser un tableau de 10 personnes
   i. Exécuter le code et commentez les éléments de la console
   ii. Est-ce qu'une erreur est survenue ? laquelle ?

```
ch.heg.ig.sda.app.Serie2 >
Usages | Output - Serie2 (run)
run:
Average age: 25.000000

Personne la plus jeune :
I'm a person
First name : Benoît
Last name : Charron
Exception in thread "main" java.lang.NullPointerException

Personne la plus âgée :
I'm a person
First name : Steve
Last name : Dawson
        at ch.heg.ig.sda.app.Serie2.getAverageAge(Serie2.java:179)
        at ch.heg.ig.sda.app.Serie2.Point14Error(Serie2.java:163)
        at ch.heg.ig.sda.app.Serie2.main(Serie2.java:40)
C:\Users\maximili.jeannere\AppData\Local\NetBeans\Cache\8.2\executor-snippets\run.xml:53: Java returned: 1
BUILD FAILED (total time: 0 seconds)
```

```
public class NullPointerException
extends RuntimeException
```

Thrown when an application attempts to use `null` in a case where an object is required. These include:

- Calling the instance method of a `null` object.
- Accessing or modifying the field of a `null` object.
- Taking the length of `null` as if it were an array.
- Accessing or modifying the slots of `null` as if it were an array.
- Throwing `null` as if it were a `Throwable` value.

Applications should throw instances of this class to indicate other illegal uses of the `null` object. `NullPointerException` objects may be constructed by the virtual machine as if suppression were disabled and/or the stack trace was not writable.

Plus d'infos sur : https://docs.oracle.com/javase/8/docs/api/java/lang/NullPointerException.html