TD 2 Des classes, des constructeurs, des attributs

Pendant ce TD nous allons essayer de comprendre comment manipuler des objets, des classes et des constructeurs, et comment comprendre leur fonctionnement. Vous allez travailler en binôme.

1. Nous allons partir sur une classe Etudiant, pour laquelle on a le code suivant :

```
public class Etudiant {
        private String nom ;
        private String prenom ;
        private String etablissement ;
        private String formation ;
        private int annee ;
        private int age ;
}
```

Dans une classe principale MonCode, dans la méthode main, nous allons avoir le code suivant :

```
…
public static void main(String[] args){
        final Etudiant alain = new Etudiant() ;
        System.out.println(alain) ;
}
```

- Est-ce que ce code compilera ? Pourquoi (pourquoi pas) ?
- Quel sera l'affichage ?

Nous allons rajouter le constructeur suivant dans la classe Etudiant :

```
public Etudiant(String monNom, String monPrenom, int monAge){
        this.nom = monNom ;
        this.prenom = monPrenom ;
        this.age = monAge ;
}
```

- Quel est l'effet d'ajouter cette méthode sur la compilation/exécution du code dans la méthode principale ?
- Faites un diagramme de classe pour la classe Etudiant.

- Le constructeur ci-dessus (signature Etudiant(String, String, int)) n'initialise que trois des attributs de la classe Etudiant. Imaginez-vous qu'on remplace le code dans la méthode main par les lignes suivantes :

  ```
  …
  public static void main(String[] args){
        final Etudiant alain = new Etudiant("Dupont", "Alain",18) ;
        System.out.println(alain) ;
  }
  ```

  Est-ce que ce code compilera ? Pourquoi (pourquoi pas) ?
- Quelle(s) est/sont la/les raison(s) pour laquelle/lesquelles on pourrait vouloir mettre juste trois valeurs en entrée d'un constructeur pour la classe Etudiant ?
- Comment peut-on initialiser les autres attributs de notre étudiant créé, Alain ?

2. Imaginez-vous une classe Etudiant avec les attributs mentionnés dans l'exercice numéro 1, mais avec les méthodes suivantes :

   - **public Etudiant(String, String, String, String)** : qui initialise les valeurs des attributs du type String de la classe Etudiant (dans l'ordre donnée en haut) aux valeurs données en entrée respectivement , et qui met **this.annee = 1** et **this.age = 18**;
   - **public Etudiant(String, String, String, String, int, int)** : qui initialise les valeurs des attributs de la classe Etudiant (dans l'ordre donnée en haut) aux valeurs en entrée, respectivement ;
   - **public void initAnnee(int)** : qui change la valeur de l'attribut annee à la valeur donnée en entrée.
   - **String toString()** : qui affiche les attributs d'un étudiant dans l'ordre décrite ci-dessus, séparés par des virgules. (par exemple : **Dupont, Alain, IUT, informatique, 1, 18**)

   Dans la méthode principale (main) nous avons le code suivant :

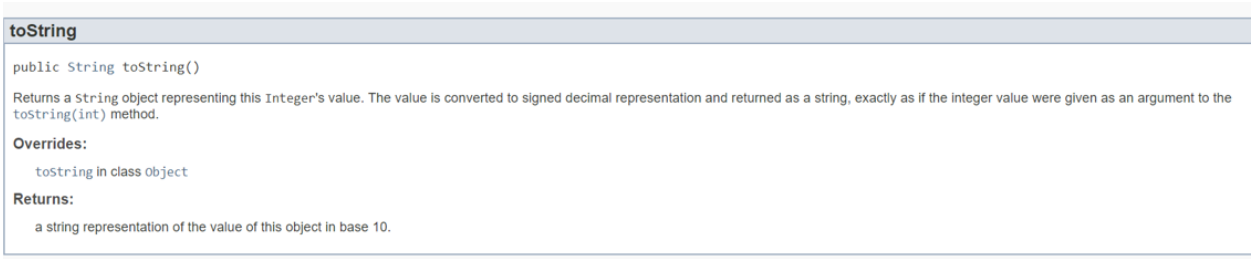
   ```
   …
   public static void main(String[] args){
         final Etudiant alain = new Etudiant("Dupont", "Alain", "IUT", "informatique", 1, 18) ;
         final Etudiant copieDAlain = Alain ;
         System.out.println(alain) ;
         System.out.println(copieDAlain) ;
         alain.initAnnee(2) ;
         System.out.println(alain) ;
         System.out.println(copieDAlain) ;

   }
   ```

- Est-ce que ce code compile ? Quel est le résultat de l'exécution ? Pourquoi ?
- Et si on remplace la ligne **alain.initAnnee(2)** par **copieDAlain.initAnnee(2)** ?
- Quelle est le diagramme de classe pour la classe Etudiant avec ces nouvelles méthodes ?

3. Dans la classe principale nous allons créer une méthode qui prendra en entrée une chaine de caractères (représentant la valeur d'un attribut), et qui retournera tous les étudiants avec ces caractéristiques. Nous allons également interpréter l'**age**/l'**annee** comme une chaine de caractères.

   La méthode aura la signature **Etudiant[] trouverEtudiants(Etudiant[], String)**. On considère que le nombre maximal d'étudiants sera fixé à 30 étudiants.

   - Renseignez-vous sur comment convertir une valeur entière vers une chaine de caractères en utilisant la méthode **Integer.toString(int)**. Vous pourrez la trouver sur : https://docs.oracle.com/javase/7/docs/api/java/lang/Integer.html .



toString

`public String toString()`

Returns a `String` object representing this `Integer`'s value. The value is converted to signed decimal representation and returned as a string, exactly as if the integer value were given as an argument to the `toString(int)` method.

Overrides:

`toString` in class `Object`

Returns:

a string representation of the value of this object in base 10.

   - De quelles méthodes additionnelles avez-vous besoin dans la classe Etudiant ? (donnez juste la signature et leur fonctionnement, le code ne sera pas nécessaire.) Mettez à jour votre diagramme de classe.
   - Écrivez la méthode trouverEtudiants(Etudiant[] mesEtudiants, String maCaracteristique) (à mettre dans la classe **MonCode**). Attention : puisque cette méthode sera appelée dans la méthode principale, qui est statique, votre méthode devra également être statique !