

Module 1 - Informatique - Algorithmique et Programmation Objet

Travaux Dirigés (8), Licence 1ère Année

Portées des Attributs, Méthodes et Classes

Exercices

- 1.1.** Créez une classe **Etudiant** ayant deux attributs :
- **nom** une chaîne de caractère (String) qui contiendra le nom de l'étudiant et
 - **prenom** une chaîne de caractère (String) qui contiendra le prénom de l'étudiant.
 - Un constructeur sans paramètre qui initialise les deux attributs **nom** et **prenom** avec la chaîne "**inconnu**".

Exemple

```

1 public class Etudiant {
2     String nom;
3     String prenom;
4
5     Etudiant(){
6         nom = "Inconnu";
7         prenom = "Inconnu";
8     }
9 }

```

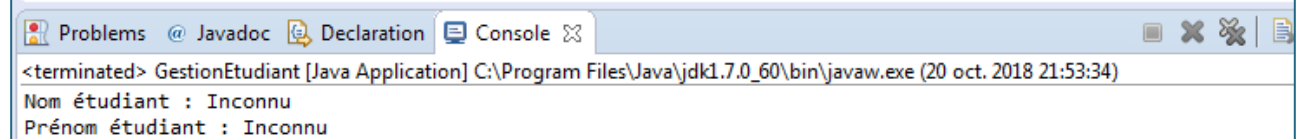
- 1.2.** Créez une classe **GestionEtudiant** avec une méthode **main()**. Dans cette dernière créez un objet **Etudiant** et affichez son nom et prénom (attributs **nom** et **prenom**).

Exemple

```

1 public class GestionEtudiant {
2     public static void main(String[] args) {
3         Etudiant etu = new Etudiant();
4         System.out.println("Nom étudiant : " + etu.nom);
5         System.out.println("Prénom étudiant : " + etu.prenom);
6     }
7 }

```



Problems @ Javadoc Declaration Console

<terminated> GestionEtudiant [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (20 oct. 2018 21:53:34)

Nom étudiant : Inconnu
Prénom étudiant : Inconnu

- 1.3.** Ajouter à votre classe **Etudiant** :
- un nouveau constructeur acceptant un paramètre chaîne de caractères **n** avec lequel il initialise l'attribut **nom**. L'attribut **prenom** sera initialisé à "**Inconnu**" par appel au constructeur déjà existant (sans paramètre).
Note : l'appel au constructeur sans paramètre devra être fait en premier.
 - un nouveau constructeur acceptant deux paramètres chaînes de caractères **n** et **p** avec lesquels il initialise les attributs **nom** et **prenom**. Ce constructeur fera appel au constructeur précédent (avec un paramètre) pour initialiser l'attribut **prenom**.
Note : l'appel au constructeur avec un paramètre devra être fait en premier.

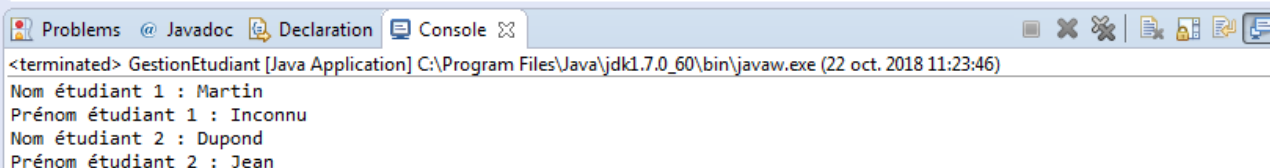
Dans votre méthode **main()**, créez deux objets **Etudiant** chacun avec un des deux constructeurs et affichez leurs attributs **nom** et **prenom**.

Exemple

```

1 public class GestionEtudiant {
2     public static void main(String[] args) {
3         Etudiant etu1 = new Etudiant("Martin");
4         System.out.println("Nom étudiant 1 : " + etu1.nom);
5         System.out.println("Prénom étudiant 1 : " + etu1.prenom);
6         Etudiant etu2 = new Etudiant("Dupond", "Jean");
7         System.out.println("Nom étudiant 2 : " + etu2.nom);
8         System.out.println("Prénom étudiant 2 : " + etu2.prenom);
9     }
10 }

```



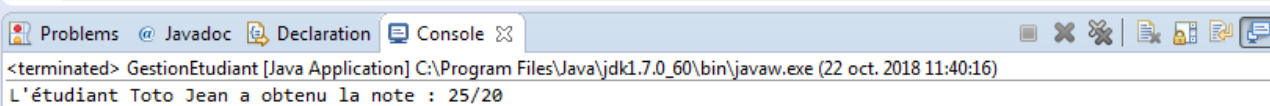
- 1.4. Ajouter à votre classe **Etudiant** un attribut **note** (entier). À partir du **main()**, créez un objet de nom "**Toto**" et de prénom "**Jean**". Affectez la valeur **25** à son attribut **note** et affichez-la.

Exemple

```

10 Etudiant etu = new Etudiant("Toto", "Jean");
11 etu.note = 25;
12 System.out.println("L'étudiant " + etu.nom + " " + etu.prenom + " a obtenu la note : " + etu.note + "/20");

```



Note : l'exemple montre que lorsqu'un attribut (**note**) d'un objet d'une classe est accessible à partir d'une autre classe, il peut être modifié sans aucun contrôle. Il en est de même pour les attributs **nom** et **prenom**. Pour éviter ces manipulations incontrôlées, il faut déclarés les attributs concernés en privés (mot clé **private**) de façon qu'ils ne soient accessibles qu'à partir de leur classe. Pour pouvoir accéder à leur valeur et les modifier, il est nécessaire d'ajouter des méthodes qui accomplissent ces tâches.

- 1.5. Modifiez votre classe **Etudiant** comme suit :
 - Déclarez les attributs **nom**, **prenom** et **note** en privés (précédez leur déclaration par le mot clé **private**).
 - Ajoutez une méthode **String getNom()** qui retourne la valeur de l'attribut **nom**.
 - Ajoutez une méthode **String getPrenom()** qui retourne la valeur de l'attribut **prenom**.
 - Ajoutez une méthode **void initNote(int n)** qui initialise l'attribut **note** avec la valeur **n** si et seulement si **0 ≤ n ≤ 20**.
 - Ajoutez une méthode **int getNote()** qui retourne la valeur actuelle de l'attribut **note**.

Dans votre **main()**, créez un nouvel objet **Etudiant**, initialisez son attribut **note** avec une valeur en dehors de l'intervalle **[0, 20]** dans un premier temps puis avec une note dans l'intervalle **[0, 20]** et, dans les deux cas, affichez les nom, prénom et note du nouvel objet créé.

Exemple

```

10 Etudiant etu = new Etudiant("Toto", "Jean");
11 etu.initNote(25);
12 System.out.println("L'étudiant " + etu.getNom()+" "+etu.getPrenom()+" a obtenu la note : "+etu.getNote()+"/20"
13 etu.initNote(15);
14 System.out.println("L'étudiant " + etu.getNom()+" "+etu.getPrenom()+" a obtenu la note : "+etu.getNote()+"/20"

```

1.6. Ajouter à votre classe **Etudiant** un attribut **effectif** (entier) qui va comptabiliser le nombre d'objets **Etudiant** créés. Enfin, modifiez en conséquence vos constructeurs pour que cet attribut soit incrémenté à chaque création d'objet **Etudiant**.

Note : attention à n'incrémenter l'attribut **effectif** qu'une fois lors de la création d'un objet. Rappelez-vous qu'il vous a été demandé que les constructeurs s'appellent entre eux.

Dans votre **main()**, créez trois objets **Etudiant** puis affichez la valeur de l'attribut **effectif**.

Exemple

```

1 public class GestionEtudiant {
2     public static void main(String[] args) {
3         Etudiant etu1 = new Etudiant("Toto", "Jean");
4         System.out.println("Effectif pour l'objet \" + etu1.getNom() + "\" = \" + etu1.effectif);
5         Etudiant etu2 = new Etudiant("Dupont", "Pierre");
6         System.out.println("Effectif pour l'objet \" + etu2.getNom() + "\" = \" + etu2.effectif);
7         Etudiant etu3 = new Etudiant("Martin", "Marc");
8         System.out.println("Effectif pour l'objet \" + etu3.getNom() + "\" = \" + etu3.effectif);

```

Note : l'exemple montre que chaque objet créé a son propre attribut **effectif**. Afin qu'il soit "partagé" entre tous les objets (ou qu'il ne soit lié à aucun objet en particulier), il faut le déclarer comme attribut de classe (précédez sa déclaration par le mot clé **static**).

1.7. Modifiez votre classe **Etudiant** de façon que l'attribut **effectif** soit déclaré en **static**. Exécutez à nouveau votre **main()** précédent.

Exemple (Exécution du même main() que précédemment)

```

1 public class Etudiant {
2     private String nom;
3     private String prenom;
4     private int note;
5     static int effectif = 0;

```

Note : un attribut de classe, **static**, a une "existence" en dehors de la création de tout objet (instanciation) de cette classe. Il devient "accessible" même si aucune instance (objet) de la classe n'a été créée.

- 1.8.** Affichez la valeur de l'attribut **effectif** de la classe **Etudiant** avant création d'une quelconque instance de la classe puis après création d'une instance. **Note** : un attribut de classe est accessible en le préfixant par le nom de la classe suivi d'un point (**Etudiant.effectif**).

Exemple

```

1 public class GestionEtudiant {
2     public static void main(String[] args) {
3         System.out.println("Valeur de l'attribut effectif de la classe Etudiant = " + Etudiant.effectif);
4         Etudiant etu1 = new Etudiant("Toto", "Jean");
5         System.out.println("Effectif pour l'objet \" + etu1.getNom() + "\" = " + etu1.effectif);
6         System.out.println("Valeur de l'attribut effectif de la classe Etudiant = " + Etudiant.effectif);
7     }
}

```

Problems @ Javadoc Declaration Console

<terminated> GestionEtudiant [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (22 oct. 2018 13:57:10)

Valeur de l'attribut effectif de la classe Etudiant = 0
 Effectif pour l'objet "Toto" = 1
 Valeur de l'attribut effectif de la classe Etudiant = 1

Remarque : si un attribut de classe (**static**) est déclaré également privé, alors il ne devient "accessible" que par une méthode de la classe. Cette méthode doit nécessairement être aussi une méthode de classe (**static**). Comme pour un attribut de classe, une méthode de classe a une "existence" en dehors de toute création d'objet (d'instance) de la classe. Elle peut donc être invoquée même si aucun objet de la classe n'a été créé.

- 1.9.** Dans la classe **Etudiant**, ajoutez une méthode **static int getEffectif()** qui retourne la valeur de l'attribut **effectif**. Testez-la dans votre **main()**.

Exemple

```

1 public class GestionEtudiant {
2     public static void main(String[] args) {
3         System.out.println("Valeur de l'attribut effectif de la classe Etudiant = " + Etudiant.getEffectif());
4         Etudiant etu1 = new Etudiant("Toto", "Jean");
5         System.out.println("Effectif pour l'objet \" + etu1.getNom() + "\" = " + etu1.getEffectif());
6         System.out.println("Valeur de l'attribut effectif de la classe Etudiant = " + Etudiant.getEffectif());
7     }
}

```

Problems @ Javadoc Declaration Console

<terminated> GestionEtudiant [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (22 oct. 2018 14:15:49)

Valeur de l'attribut effectif de la classe Etudiant = 0
 Effectif pour l'objet "Toto" = 1
 Valeur de l'attribut effectif de la classe Etudiant = 1

Nous voulons maintenant ajouter un attribut **identifiant** (int) à la classe **Etudiant** qui va contenir l'identifiant d'un objet **Etudiant**. Celui-ci sera construit à partir de l'attribut **nom** en sommant chacun des caractères de ce dernier (**identifiant** = somme des **nom.charAt(i)** pour tout **i**). L'attribut **identifiant** ne doit être initialisé que si l'objet créé l'est avec un attribut **nom** défini (autre que "Inconnu").

- 1.10.** Dans la classe **Etudiant**, ajoutez l'attribut **identifiant** et une méthode **int initIdentifiant(String n)** qui retourne un entier correspondant à la somme des caractères du paramètre **n** (lors de l'appel, le paramètre effectif sera l'attribut **nom**). Vous modifierez les constructeurs adéquats qui permettent l'initialisation de l'attribut **identifiant** avec la valeur retournée par la méthode. Ajoutez une méthode **int getIdentifiant()** qui retourne la valeur de l'attribut **identifiant** d'un objet. Testez-les dans votre **main()**.

Exemple

```

1 public class GestionEtudiant {
2     public static void main(String[] args) {
3         Etudiant etu1 = new Etudiant();
4         System.out.println("Identifiant de \" + etu1.getNom() + "\" = " + etu1.getIdentifiant());
5         Etudiant etu2 = new Etudiant("Toto");
6         System.out.println("Identifiant de \" + etu2.getNom() + "\" = " + etu2.getIdentifiant());
7         Etudiant etu3 = new Etudiant("Dupond", "Jean");
8         System.out.println("Identifiant de \" + etu3.getNom() + "\" = " + etu3.getIdentifiant());
9     }
}

```

Problems @ Javadoc Declaration Console

<terminated> GestionEtudiant [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (22 oct. 2018 15:20:16)

Identifiant de "Inconnu" = 0
 Identifiant de "Toto" = 422
 Identifiant de "Dupond" = 618

Remarque : tels que sont définis l'attribut **identifiant** et la méthode **initIdentifiant()**, il est permis d'invoquer cette dernière depuis une autre classe. Il est alors possible de modifier l'attribut **identifiant** d'un quelconque objet depuis l'extérieur de la classe **Etudiant**. On peut également supposer que l'identifiant d'un étudiant est établi une fois pour toute (ce qui est votre cas durant toute votre scolarité dans un même établissement) et qu'il n'a pas à être modifié par la suite. Ce qui n'est pas aussi le cas ici. L'exécution suivante illustre ces différentes situations :

Exemple

```

1 public class GestionEtudiant {
2     public static void main(String[] args) {
3         Etudiant etu = new Etudiant("Dupond", "Jean");
4         System.out.println("Identifiant de \" + etu.getNom() + "\" = " + etu.getIdentifiant());
5         etu.identifiant = 777;
6         System.out.println("Identifiant de \" + etu.getNom() + "\" = " + etu.getIdentifiant());
7         etu.identifiant = etu.initIdentifiant("Toto");
8         System.out.println("Identifiant de \" + etu.getNom() + "\" = " + etu.getIdentifiant());
9     }
}

```

Problems @ Javadoc Declaration Console

<terminated> GestionEtudiant [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (22 oct. 2018 16:33:14)

Identifiant de "Dupond" = 618
 Identifiant de "Dupond" = 777
 Identifiant de "Dupond" = 422

1.11. Dans la classe **Etudiant**, déclarez l'attribut **identifiant** comme étant une constante (déclaration précédée du mot clé **final**). Une fois l'attribut **identifiant** initialisé, aucune modification ne sera possible et toute tentative entraînera une erreur. Déclarez également la méthode **int initIdentifiant(String n)** comme étant privée (déclaration précédée du mot clé **private**) qui la rendra "visible" (et donc invocable) que de l'intérieur de la classe **Etudiant**.

Note : l'attribut **identifiant** doit être initialisé lors de toute instantiation de la classe **Etudiant**. Dans notre cas, il n'est possible de le faire que si l'objet créé l'est avec une valeur pour l'attribut **nom**. Le constructeur sans paramètre ne pourra donc pas l'initialiser et sera par conséquent supprimé. Il y a lieu de modifier également le constructeur avec un paramètre (suppression de **this()** ; qui correspond à l'appel du constructeur sans paramètre qui n'existe plus).

Exemple

```

1 public class GestionEtudiant {
2     public static void main(String[] args) {
3         Etudiant etu = new Etudiant("Dupond", "Jean");
4         etu.identifiant = 777;
  
```

Problems @ Javadoc Declaration Console

<terminated> GestionEtudiant [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (22 oct. 2018 16:42:15)
 Exception in thread "main" java.lang.Error: Unresolved compilation problem:
 The final field Etudiant.identifiant cannot be assigned
 at GestionEtudiant.main(GestionEtudiant.java:4)

Exemple

```

1 public class GestionEtudiant {
2     public static void main(String[] args) {
3         Etudiant etu = new Etudiant("Dupond", "Jean");
4         etu.identifiant = etu.initIdentifiant("Toto");
5     }
  
```

Problems @ Javadoc Declaration Console

<terminated> GestionEtudiant [Java Application] C:\Program Files\Java\jdk1.7.0_60\bin\javaw.exe (22 oct. 2018 16:45:07)
 Exception in thread "main" java.lang.Error: Unresolved compilation problem:
 The method initIdentifiant(String) from the type Etudiant is not visible
 at GestionEtudiant.main(GestionEtudiant.java:4)