

Formation Orienté Objet et Java de base

Atelier 2 : Les modificateurs

Encadré par M.BOULCHAHOU

Objectif de l'atelier.....	1
Création des packages	1
Création des classes.....	2
Les constructeurs.....	7
Import des classes dans une autre	10
Le modificateur private	11
Le modificateur public	14
Le modificateur protected	16
Le modificateur par défaut	16

Objectif de l'atelier

L'objectif de cet atelier est de créer des packages java, des classes java et de manipuler les modificateurs d'accès private, protected, public et default.

Vous allez apprendre à:

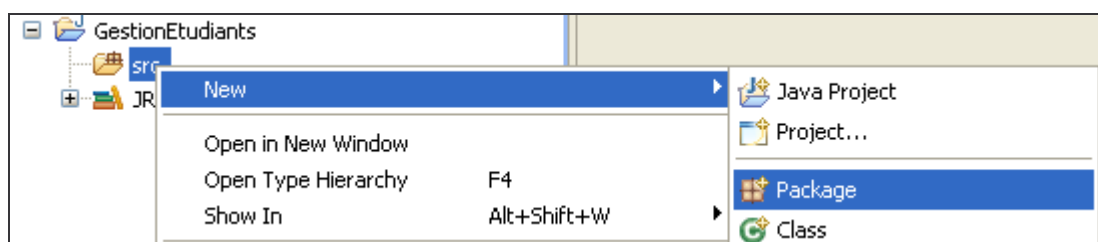
- *Créer des packages sous Eclipse après avoir créer un projet Java*
- *Créer des classes java à partir d'Eclipse*
- *Manipuler les modificateurs d'accès private, protected, public et default*

Création des packages

*Accéder à Eclipse et créer un projet Java sous le nom « GestionEtudiant ».
Créer deux packages :*

*ma.gov.fst.ecoles.etudiants
ma.gov.fst.ecoles.proffesseurs*

Pour le faire, cliquer droit sur src>new>package

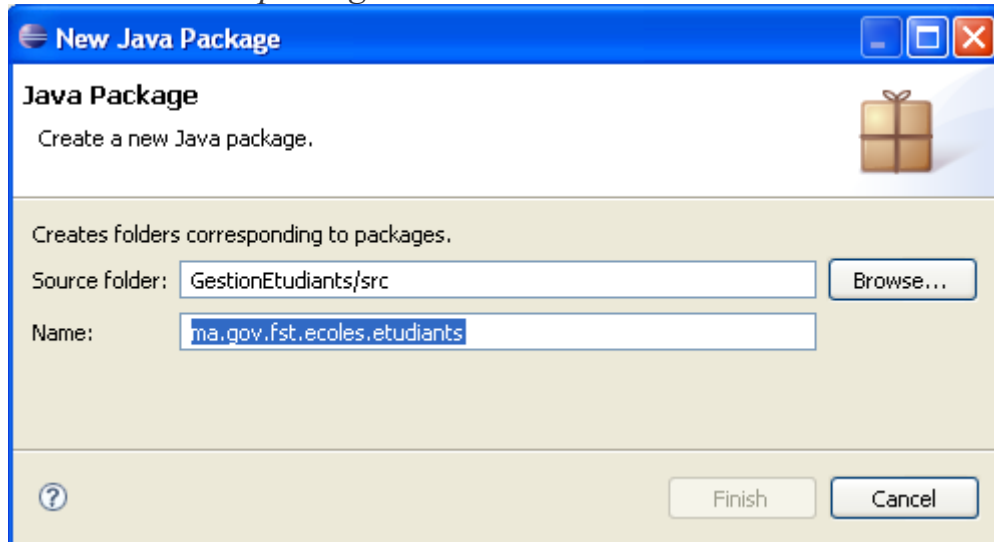


Formation Orienté Objet et Java de base

Atelier 2 : Les modificateurs

Encadré par M.BOULCHAHOU

Saisir le nom du package.



Vous aurez un projet comme celui-ci :

Règle de nommage des packages :

pays.domainestructure.structure.nomprojet.module

Exemple :

fr.gov.education.ecoles.etudiants

Création des classes

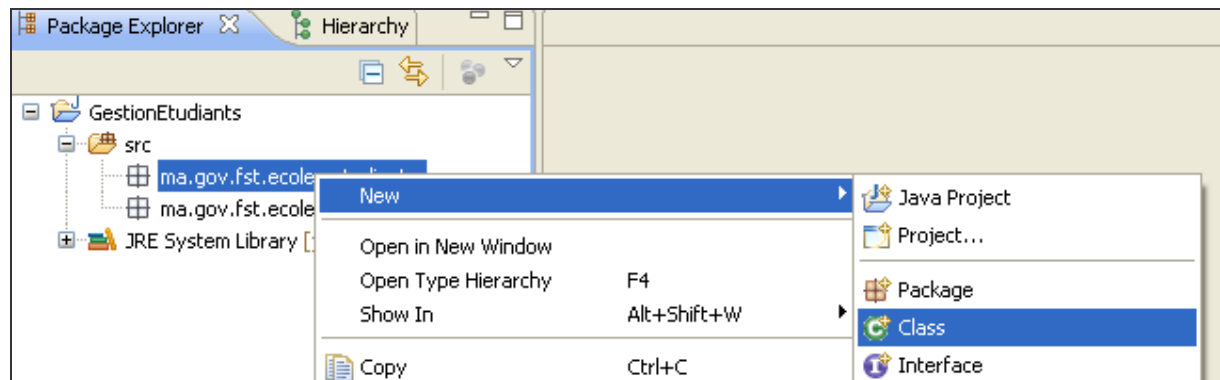
Créer une classe Etudiant.java dans le package ma.gov.fst.ecoles.etudiants. La classe java Etudiant doit contenir deux attributs nom, prenom comme des chaînes de caractères.

Pour créer une classe dans un package, en clique droit sur le package>new>Class.

Formation Orienté Objet et Java de base

Atelier 2 : Les modificateurs

Encadré par M.BOULCHAOUB



Vous aurez ensuite l'écran suivant, donner le nom Etudiant et cocher la case de la méthode main pour la créer automatiquement avec la création de la classe:

Formation Orienté Objet et Java de base

Atelier 2 : Les modificateurs

Encadré par M.BOULCHAHOUB

New Java Class

Create a new Java class.

Source folder:

Package:

☐ Enclosing type:

Name:

Modifiers: ☒ public ☐ default ☐ private ☐ protected
☐ abstract ☐ final ☐ static

Superclass:

Interfaces:

Which method stubs would you like to create?

☐ public static void main(String[] args)

☐ Constructors from superclass

☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))

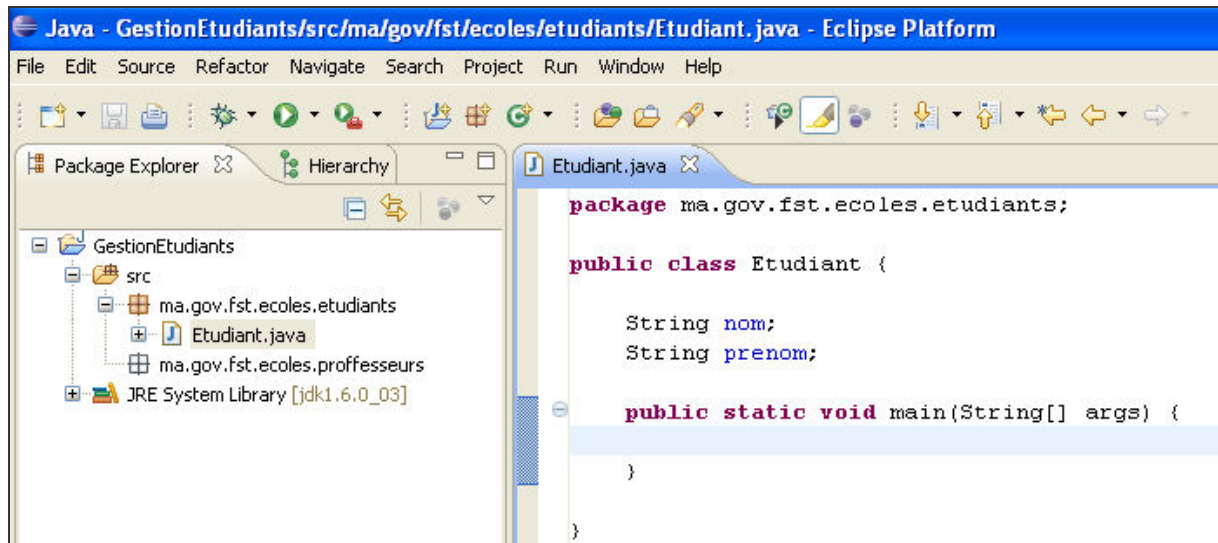
☐ Generate comments

Dans le projet « GestionEtudiants » vous aurez :

Formation Orienté Objet et Java de base

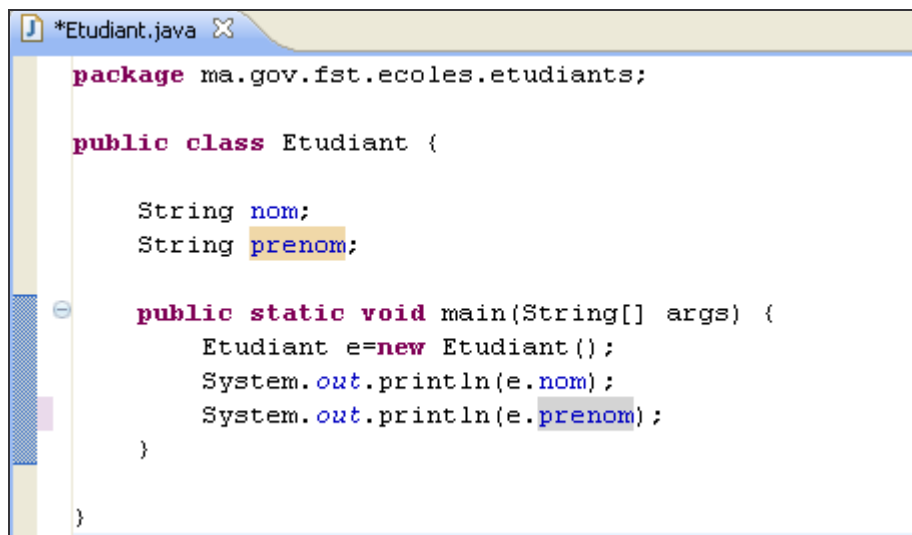
Atelier 2 : Les modificateurs

Encadré par M.BOULCHAHOUB



Instancier un objet de type Etudiant dans la méthode main. Cette instanciation se fait par l'appel au constructeur pas défaut (constructeur proposé par la JVM ne prenant aucun paramètre et qui n'apparait pas parmi les méthodes de la classe Etudiant).

Afficher le nom et le prénom de l'objet instancié. Un objet d'une classe s'appelle aussi instance de classe.



L'accès à un attribut ou à une méthode d'un objet se fait par :

nomObjet.nomAttribut pour les attributs

nomObjet.nomMethode pour les méthodes

Formation Orienté Objet et Java de base

Atelier 2 : Les modificateurs

Encadré par M.BOULCHAHOUB

Règles de nommage :

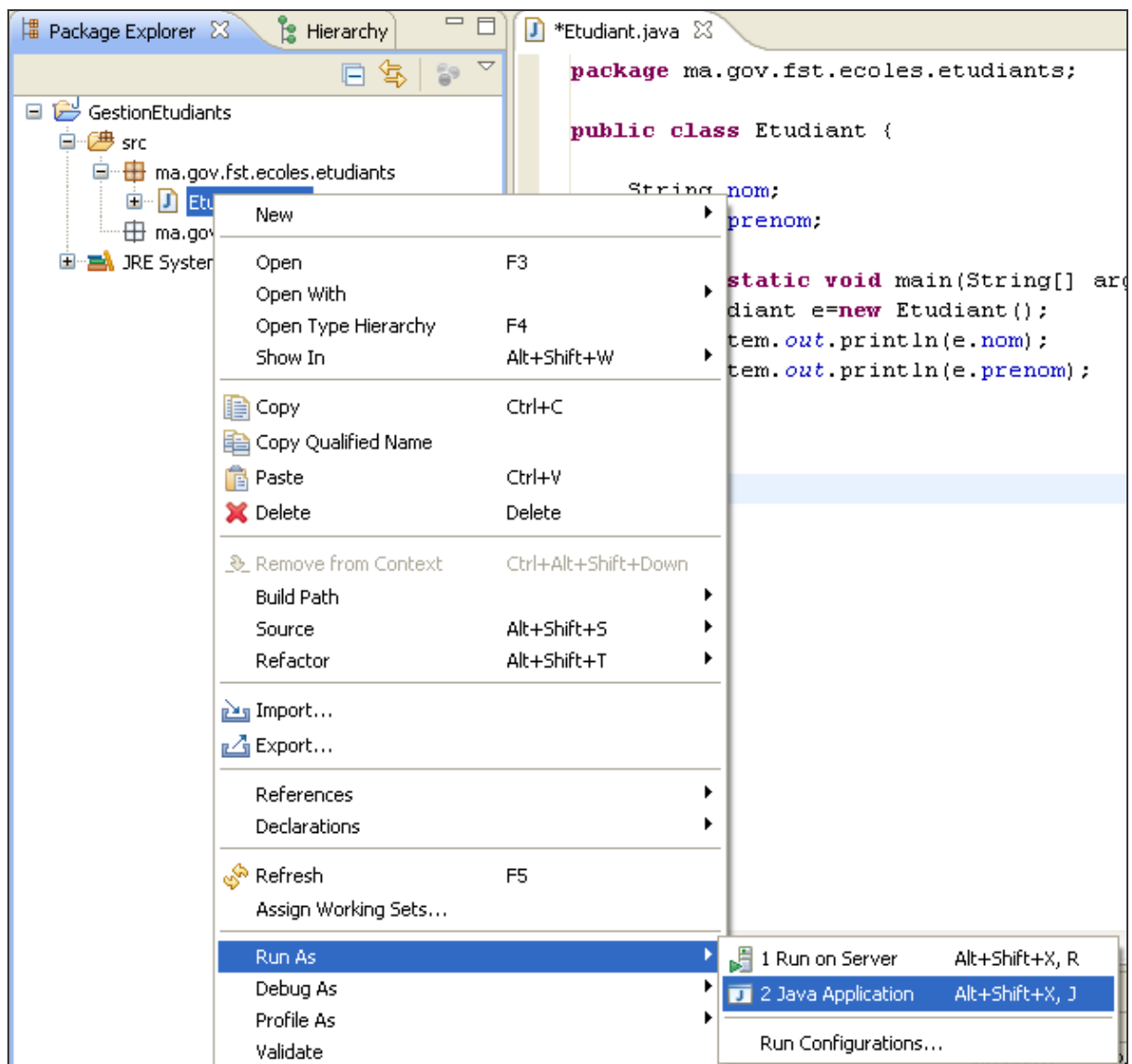
Les classes doivent commencer par une majuscule.

Les packages sont en minuscule.

Les attributs doivent commencer par une minuscule.

Les méthodes doivent commencer par une minuscule.

Exécuter la classe Etudiant (peut être exécutée car elle contient la méthode main). Pour exécuter une classe à partir de Eclipse, on clique droit sur la classe>run as>Java application

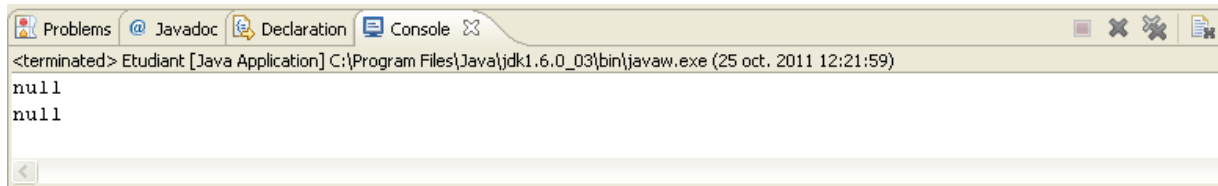


Vous aurez le résultat suivant sur la console.

Formation Orienté Objet et Java de base

Atelier 2 : Les modificateurs

Encadré par M.BOULCHAOUB



```
<terminated> Etudiant [Java Application] C:\Program Files\Java\jdk1.6.0_03\bin\javaw.exe (25 oct. 2011 12:21:59)
null
null
```

Les constructeurs

*Créer un constructeur de la classe Etudiant qui prend le nom comme paramètre.
Créer un autre constructeur de la classe Etudiant qui prend le nom et le prénom
comme paramètres.*

*Un constructeur est une méthode spéciale qui a le même nom que sa classe, elle
ne retourne aucun paramètre.*

*Une classe peut contenir une multitude de constructeurs selon le besoin
exprimé.*

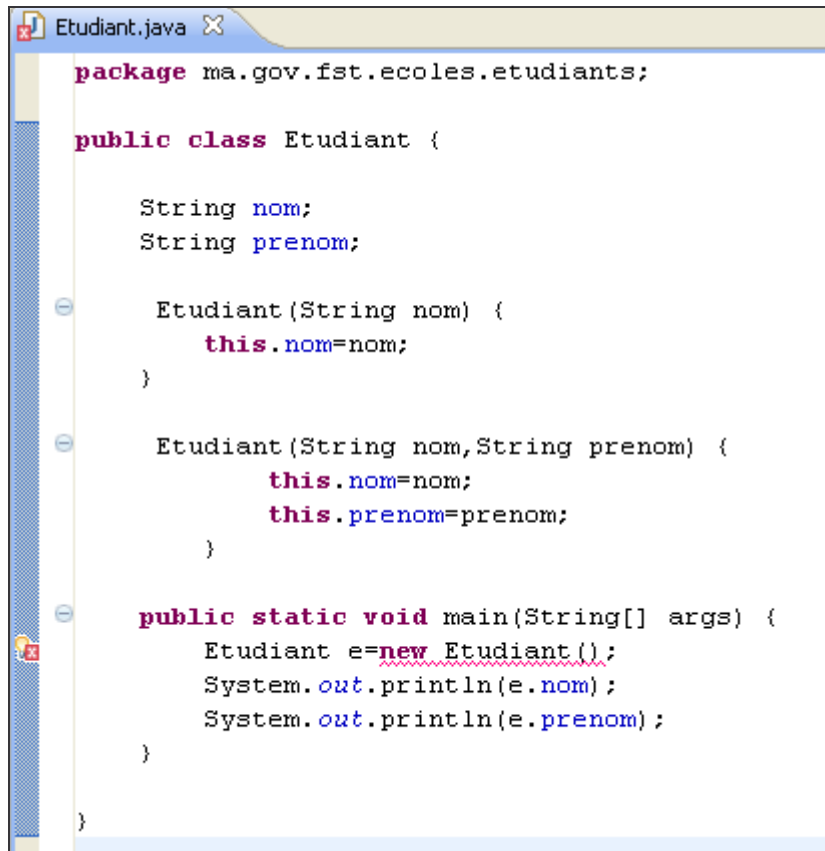
Le rôle des constructeurs est de créer des objets à partir d'une classe

Vous aurez, une classe Etudiant qui ressemble à cela :

Formation Orienté Objet et Java de base

Atelier 2 : Les modificateurs

Encadré par M.BOULCHAHOUB



```
package ma.gov.fst.ecoles.etudiants;

public class Etudiant {

    String nom;
    String prenom;

    Etudiant(String nom) {
        this.nom=nom;
    }

    Etudiant(String nom,String prenom) {
        this.nom=nom;
        this.prenom=prenom;
    }

    public static void main(String[] args) {
        Etudiant e=new Etudiant();
        System.out.println(e.nom);
        System.out.println(e.prenom);
    }
}
```

Une fois le développeur crée son propre constructeur, le constructeur par défaut livré par la JVM n'est plus disponible

La règle ci-dessous justifie l'erreur de compilation constatée au niveau de la classe Etudiant.

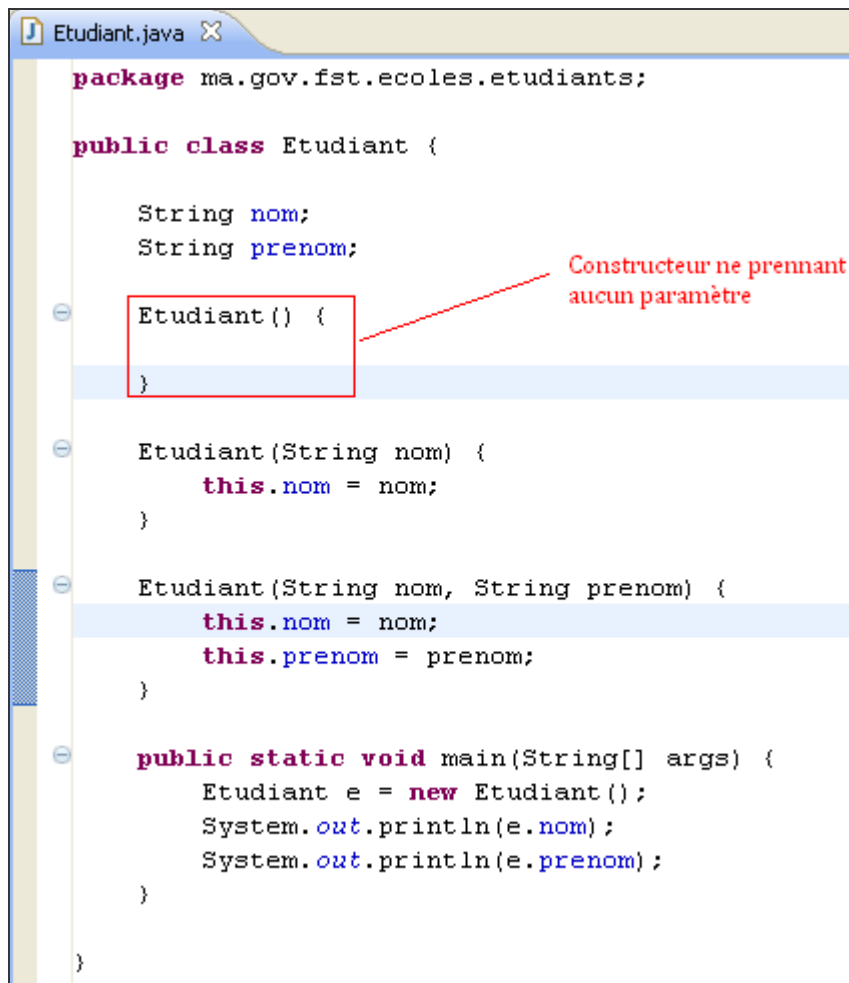
Pour corriger l'erreur deux façons sont possibles :

1^{ère} façon : Faire apparaître le constructeur par défaut au niveau de la classe Etudiant, comme suit :

Formation Orienté Objet et Java de base

Atelier 2 : Les modificateurs

Encadré par M.BOULCHAOUB



```
Etudiant.java X
package ma.gov.fst.ecoles.etudiants;

public class Etudiant {

    String nom;
    String prenom;

    Etudiant() {
    }

    Etudiant(String nom) {
        this.nom = nom;
    }

    Etudiant(String nom, String prenom) {
        this.nom = nom;
        this.prenom = prenom;
    }

    public static void main(String[] args) {
        Etudiant e = new Etudiant();
        System.out.println(e.nom);
        System.out.println(e.prenom);
    }
}
```

2^{ème} façon :

Utiliser les constructeurs créés par le développeur pour faire l'instanciation des objets de la classe Etudiant.

Formation Orienté Objet et Java de base

Atelier 2 : Les modificateurs

Encadré par M.BOULCHAOUB

```
Etudiant.java X
package ma.gov.fst.ecoles.etudiants;

public class Etudiant {

    String nom;
    String prenom;

    Etudiant(String nom) {
        this.nom = nom;
    }

    Etudiant(String nom, String prenom) {
        this.nom = nom;
        this.prenom = prenom;
    }

    public static void main(String[] args) {
        //Créer l'objet e1 en utilisant le constructeur qui prend un seul paramètre
        Etudiant e1 = new Etudiant("KADIRI");
        System.out.println(e1.nom);
        System.out.println(e1.prenom);

        //Créer l'objet e2 en utilisant le constructeur qui prend deux paramètres
        Etudiant e2 = new Etudiant("KADIRI", "Mohammed");
        System.out.println(e2.nom);
        System.out.println(e2.prenom);
    }
}
```

Import des classes dans une autre

Créer une classe Professeur dans le package ma.gov.fst.ecoles.professeurs.

Créer la méthode main dans la classe Professeur.

Instancier un objet de la classe Etudiant au niveau de la méthode main de la classe Professeur.

```
Etudiant.java X Professeur.java X
package ma.gov.fst.ecoles.professeurs;

public class Professeur {

    public static void main(String[] args) {
        Etudiant e = new Etudiant("SAHL", "Omar");
    }
}
```

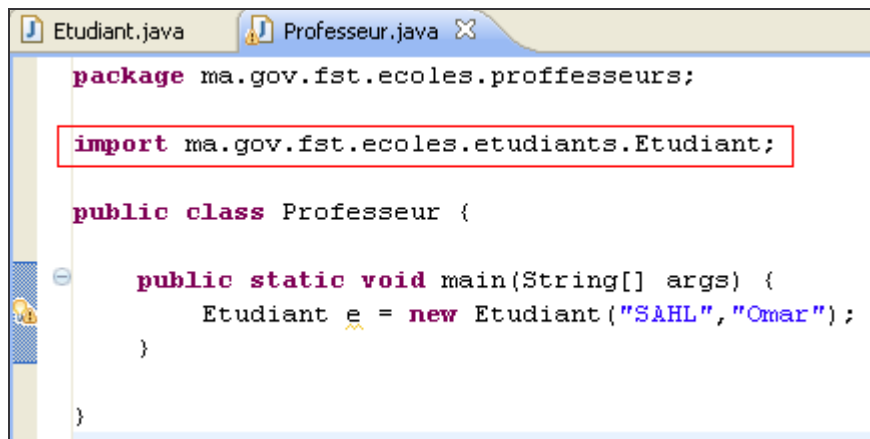
Formation Orienté Objet et Java de base

Atelier 2 : Les modificateurs

Encadré par M.BOULCHAHOUB

Noter l'erreur de compilation apparue suite à l'utilisation de la classe Etudiant dans la classe Professeur.

*Si une classe (**classe appelante**) utilise une autre classe (**classe appelée**) qui n'existe pas dans son package, la classe appelante doit importer la classe appelé.*



```
package ma.gov.fst.ecoles.proffesseurs;

import ma.gov.fst.ecoles.etudiants.Etudiant;

public class Professeur {

    public static void main(String[] args) {
        Etudiant e = new Etudiant("SAHL", "Omar");
    }
}
```

Le constructeur de la classe Etudiant doit être public pour qu'il soit visible dans la classe professeur. Bienvenue à la notion de visibilité et d'accessibilité...voir la suite de cet atelier.

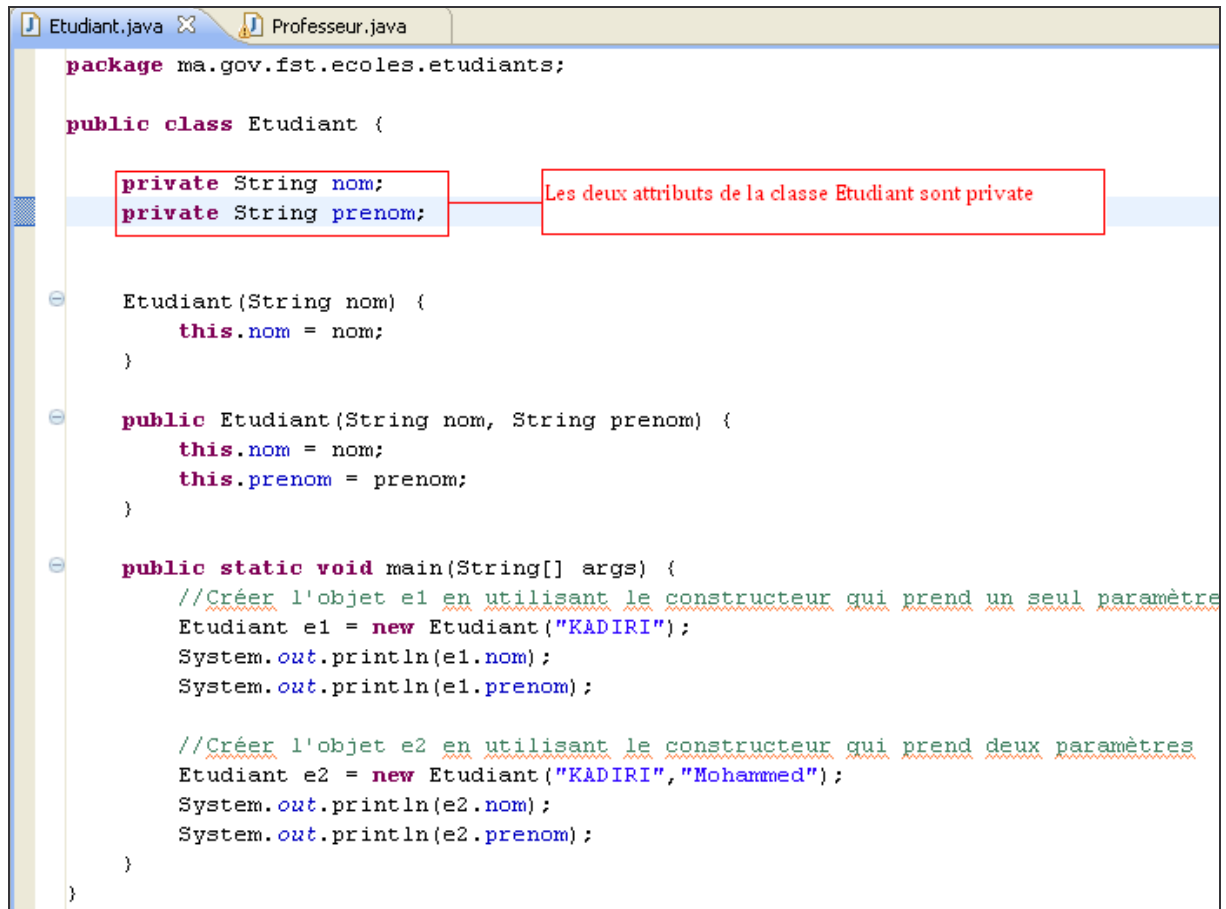
Le modificateur private

Faire précéder le nom et le prénom de la classe Etudiant.java par private.

Formation Orienté Objet et Java de base

Atelier 2 : Les modificateurs

Encadré par M.BOULCHAHOUB



```
package ma.gov.fst.ecoles.etudiants;

public class Etudiant {

    private String nom;
    private String prenom;

    Etudiant(String nom) {
        this.nom = nom;
    }

    public Etudiant(String nom, String prenom) {
        this.nom = nom;
        this.prenom = prenom;
    }

    public static void main(String[] args) {
        //Créer l'objet e1 en utilisant le constructeur qui prend un seul paramètre
        Etudiant e1 = new Etudiant("KADIRI");
        System.out.println(e1.nom);
        System.out.println(e1.prenom);

        //Créer l'objet e2 en utilisant le constructeur qui prend deux paramètres
        Etudiant e2 = new Etudiant("KADIRI", "Mohammed");
        System.out.println(e2.nom);
        System.out.println(e2.prenom);
    }
}
```

Remarquer que l'objet *e1* et *e2* accèdent aux attributs *nom* et *prénom* dans la méthode *main* de la classe *Etudiant*.

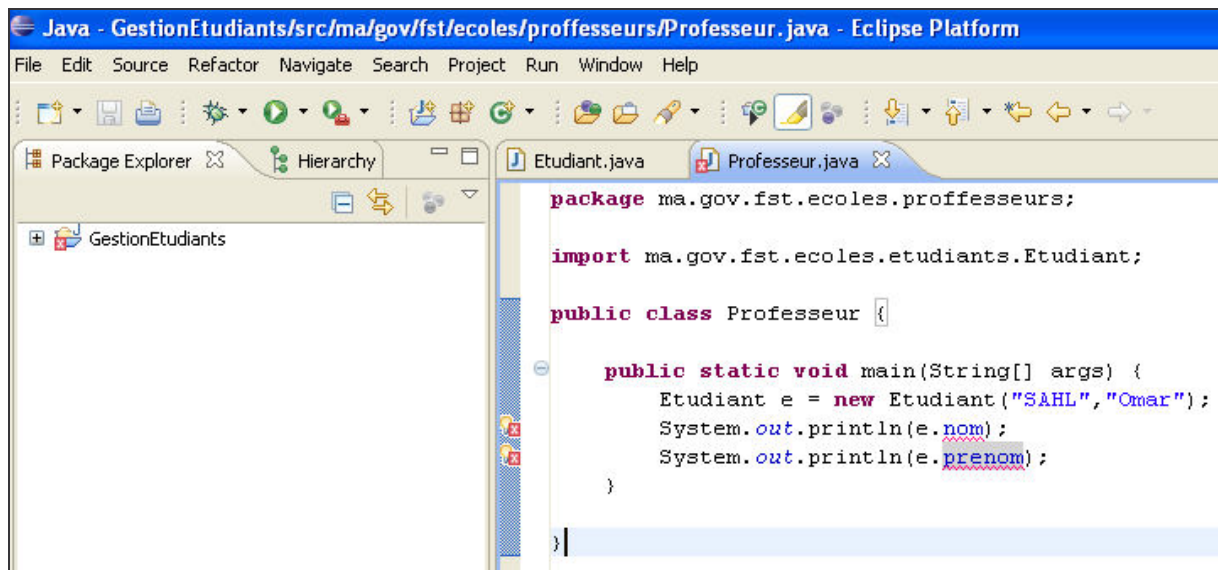
Un attribut ou une méthode marqués *private* n'est pas visible qu'à l'intérieure de sa classe.

Instancier un objet (*e*) de la classe *Etudiant* au niveau de la méthode *main* de la classe *Professeur*. Afficher le nom et le prénom de l'objet (*e*) nouvellement crée.

Formation Orienté Objet et Java de base

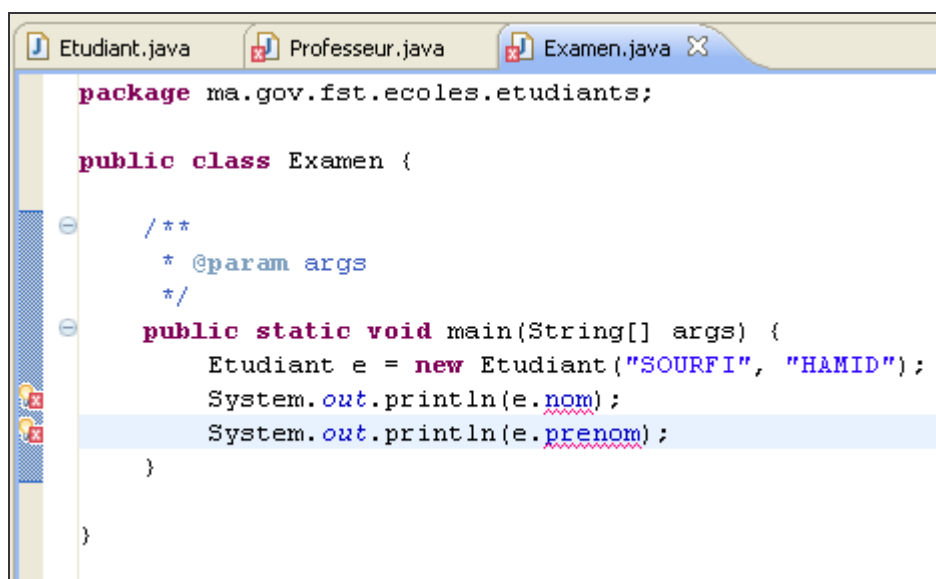
Atelier 2 : Les modificateurs

Encadré par M.BOULCHAOUB



Remarquer les erreurs de compilation au niveau de la classe Professeur. Le nom et le prénom de l'objet (e) ne sont pas visible/ne sont accessible à travers la classe Professeur car ils sont private.

Crée une classe Examen dans le même package que la classe Etudiant. Examen contiendra la méthode main dans laquelle on instancie un objet Etudiant.



Formation Orienté Objet et Java de base

Atelier 2 : Les modificateurs

Encadré par M.BOULCHAHOU

Remarquer les erreurs de compilation au niveau de la classe Examen. Le nom et le prénom de l'objet (e) ne sont pas visible/ne sont accessible à travers la classe Examen car ils sont private.

→ Un attribut, une méthode private seront visibles uniquement dans sa classe.
→ Un attribut, une méthode private ne seront pas visibles dans les classes du même package.
→ Un attribut, une méthode private ne seront pas visibles dans les classes des autres packages.
→ Une classe ne peut pas être private.

	<i>Méthode</i>	<i>Attribut</i>	<i>Classe</i>
Private	Visible uniquement dans la même classe	Visible uniquement dans la même classe	N'existe pas

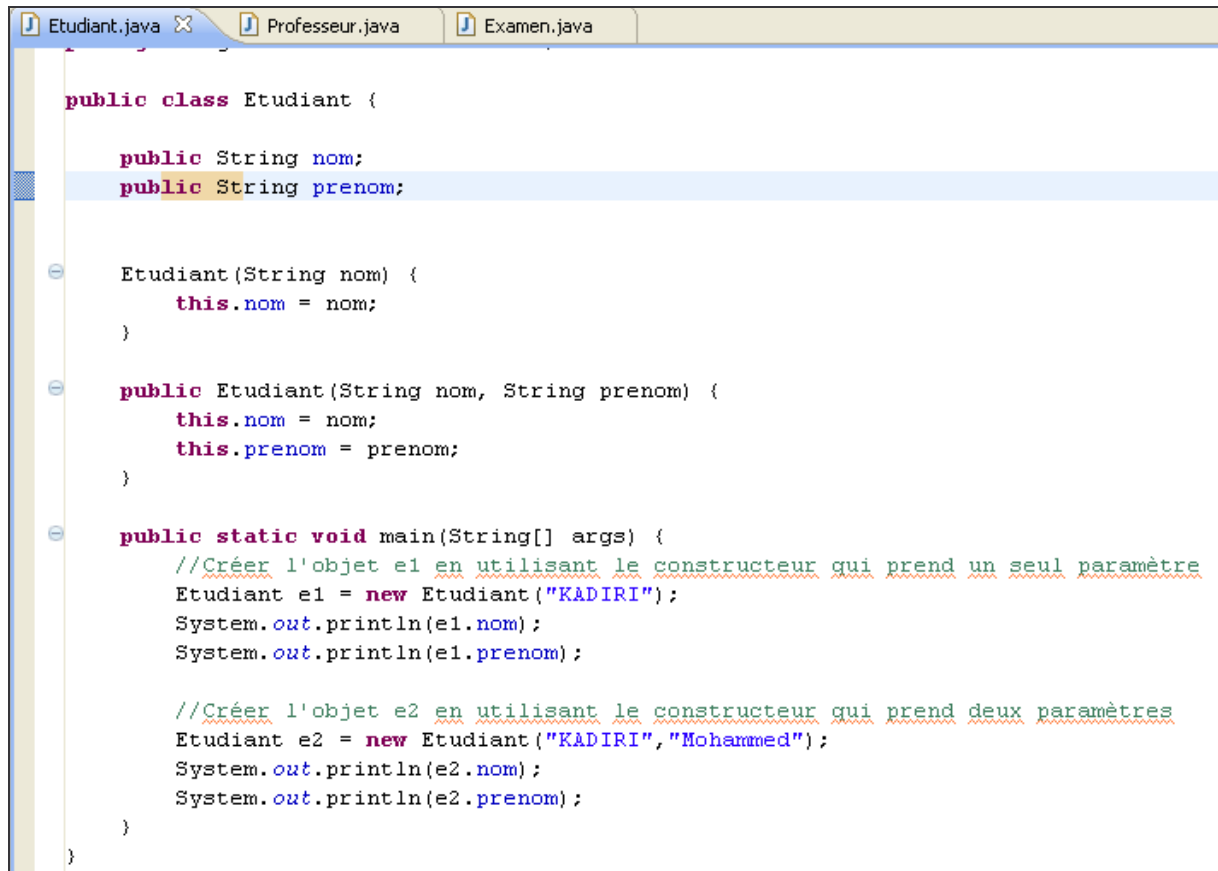
Le modificateur public

Marquer les attributs de la classe Etudiant en public.

Formation Orienté Objet et Java de base

Atelier 2 : Les modificateurs

Encadré par M.BOULCHAOUB



```
public class Etudiant {  
  
    public String nom;  
    public String prenom;  
  
    Etudiant(String nom) {  
        this.nom = nom;  
    }  
  
    public Etudiant(String nom, String prenom) {  
        this.nom = nom;  
        this.prenom = prenom;  
    }  
  
    public static void main(String[] args) {  
        //Créer l'objet e1 en utilisant le constructeur qui prend un seul paramètre  
        Etudiant e1 = new Etudiant("KADIRI");  
        System.out.println(e1.nom);  
        System.out.println(e1.prenom);  
  
        //Créer l'objet e2 en utilisant le constructeur qui prend deux paramètres  
        Etudiant e2 = new Etudiant("KADIRI", "Mohammed");  
        System.out.println(e2.nom);  
        System.out.println(e2.prenom);  
    }  
}
```

Remarquer que les erreurs de compilation sont disparues au niveau de la classe Professeur et au niveau de la classe Examen.

Les attributs nom et prenom sont maintenant visibles dans les classe Professeur et Examen car les deux attributs sont publics.

→ Un attribut, une méthode publique seront visibles dans leurs classes.
→ Un attribut, une méthode publique seront également visibles dans toutes les classes du projet.

Formation Orienté Objet et Java de base

Atelier 2 : Les modificateurs

Encadré par M.BOULCHAHOU

	<i>Méthode</i>	<i>Attribut</i>	<i>Classe</i>
public	Visible dans toutes les classes du projet	Visible dans toutes les classes du projet	Visible dans toutes les classes du projet

Le modificateur protected

Cette visibilité est fortement liée à la notion de l'héritage qui n'est pas encore abordée.

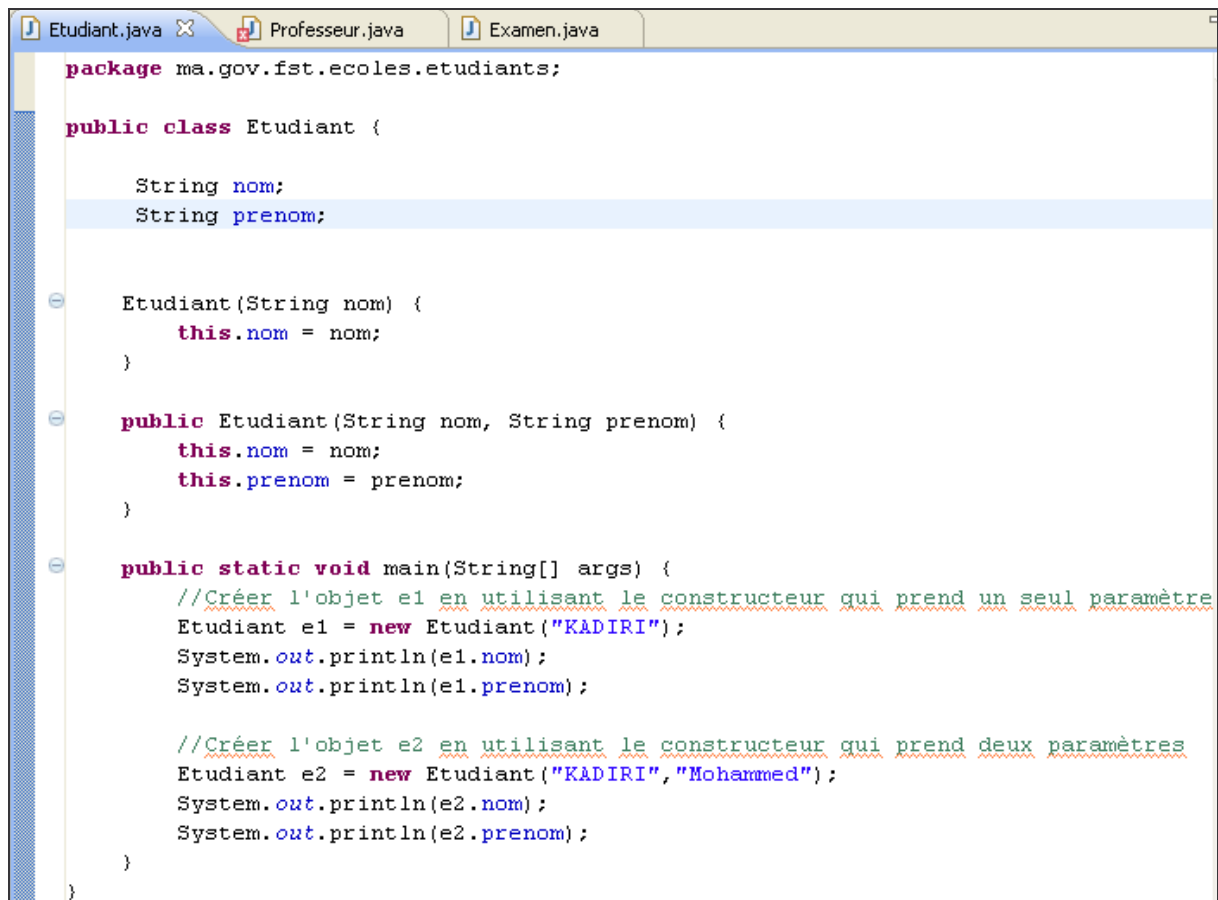
Le modificateur par défaut

Supprimer le public pour les attributs nom et prénom. Les attributs nom et prénom sont dits deux attributs avec la visibilité par défaut. (Si un attribut n'est pas précédé par private, public ou protected ce dernier a la visibilité par défaut).

Formation Orienté Objet et Java de base

Atelier 2 : Les modificateurs

Encadré par M.BOULCHAHOUB



```
package ma.gov.fst.ecoles.etudiants;

public class Etudiant {

    String nom;
    String prenom;

    Etudiant(String nom) {
        this.nom = nom;
    }

    public Etudiant(String nom, String prenom) {
        this.nom = nom;
        this.prenom = prenom;
    }

    public static void main(String[] args) {
        //Créer l'objet e1 en utilisant le constructeur qui prend un seul paramètre
        Etudiant e1 = new Etudiant("KADIRI");
        System.out.println(e1.nom);
        System.out.println(e1.prenom);

        //Créer l'objet e2 en utilisant le constructeur qui prend deux paramètres
        Etudiant e2 = new Etudiant("KADIRI", "Mohammed");
        System.out.println(e2.nom);
        System.out.println(e2.prenom);
    }
}
```

Remarquer que l'erreur de compilation existe maintenant uniquement dans la classe Professeur qui a un package différent de celui de la classe Etudiant.

→ Un attribut, une méthode ayant la visibilité par défaut, seront visibles dans leurs classes et dans les classes qui existent dans le même package.

	Méthode	Attribut	Classe
default	Toutes les classes du même package	Toutes les classes du même package	Toutes les classes du même package

FIN DE L'ATELIER