

Formation 'Développeur Java Web Mobile'

Module 1 – Séance 12



Module 1 36h

CONCEPTS OBJETS ÉLÉMENTAIRES

- Séance 1 Le métier de Développeur
- Séance 2 5 atouts Java + 5 repères clés développeur Java
- Séance 3 Installation et analyse du JDK
- Séance 4 Premier programme java + Syntaxe (instruction, méthode)
- Séance 5 Syntaxe Java (création de methodes)
- Séance 6 Syntaxe Java (boucles)
- Séance 7 Syntaxe Java (types primitifs)
- Séance 8 Prise en main outil Eclipse + 15 bonnes pratiques
- Séance 9 Génération documentation : outil javadoc.exe
- Séance 10 Création de classes et d'Objets
- Séance 11 Encapsulation (propriétés private, protected)
- Séance 12 Constructeur (définition + codage)

Sommaire Module 1 – Séance 12

- **Analyse Javadoc: constructeurs JFrame** **3**
- **Définition d'un constructeur** **4**
- **Mise en oeuvre constructeurs** **5**
- **Exemple codage Constructeurs** **8**
- **Questions** **10**

Javadoc : Analyse Constructeurs classe JFrame

Constructeur utilisé
pour DemoFenetre1



Constructeur utilisé
pour DemoFenetre2



Constructor Summary

Constructors

Constructor and Description

JFrame()

Constructs a new frame that is initially invisible.

JFrame(GraphicsConfiguration gc)

Creates a Frame in the specified GraphicsConfiguration of a screen device and a blank title.

JFrame(String title)

Creates a new, initially invisible Frame with the specified title.

JFrame(String title, GraphicsConfiguration gc)

Creates a JFrame with the specified title and the specified GraphicsConfiguration of a screen device.

Method Summary

All Methods

Static Methods

Instance Methods

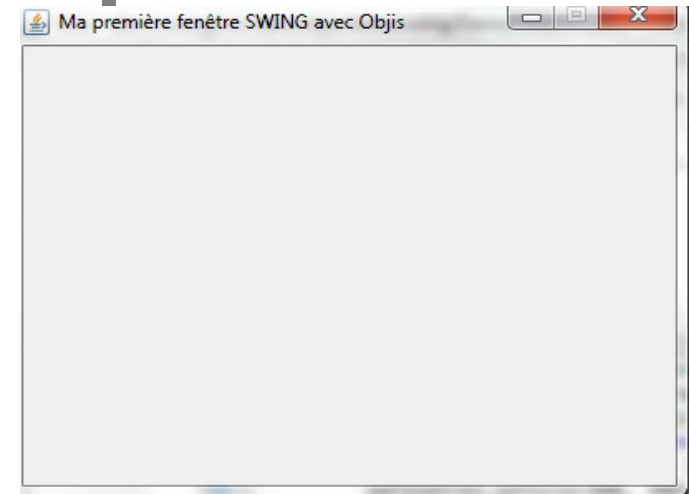
Concrete Methods

Constructeur

- Un constructeur est une méthode.
- Un constructeur a le même nom que la classe.
- Il peut y avoir plusieurs constructeurs par classe.
- Un constructeur a une visibilité 'public'.
- Un constructeur n'a pas de type de retour.
- Un constructeur permet d'initialiser les propriétés d'un objet.

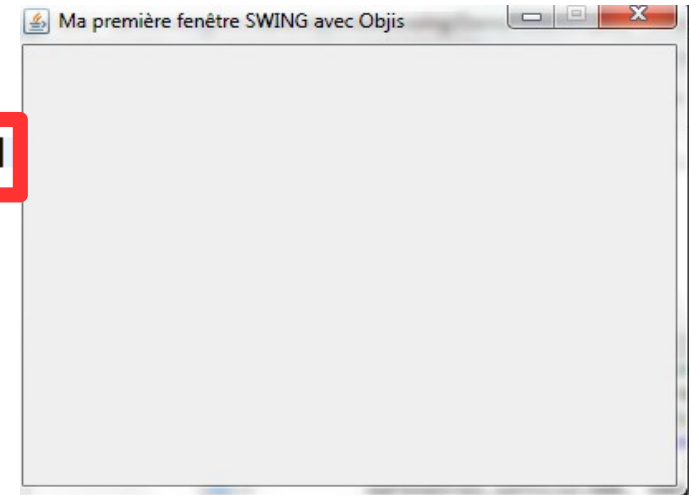
Analyse code DemoFenetre1

```
DemoFenetre1.java ✕  
  
package com.objis.demoswing;  
  
import javax.swing.JFrame;  
  
public class DemoFenetre1 {  
  
    public static void main(String[] args) {  
        // TODO 1. Instancier une JFrame  
        JFrame maFenetre1 = new JFrame();  
  
        // TODO 2. Modifier le titre de la fenêtre  
        maFenetre1.setTitle("Ma première fenêtre SWING avec Objis");  
  
        // TODO 3. Modifier la taille (400 * 300)  
        maFenetre1.setSize(400, 300);  
  
        // TODO 4. taille non modifiable par l'utilisateur  
        maFenetre1.setResizable(false);  
  
        // TODO 5. Un click sur croix entraine fermeture de la fenêtre  
        maFenetre1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
  
        // TODO 6. Centrer la fenêtre par rapport à l'écran de l'ordinateur  
        maFenetre1.setLocationRelativeTo(null);  
  
        // TODO 7. Afficher la JFrame  
        maFenetre1.setVisible(true);  
  
    }  
}
```



Analyse code DemoFenetre2 (même rendu visuel)

```
DemoFenetre2.java
1 package com.objis.demoswing;
2
3 import javax.swing.JFrame;
4
5 public class DemoFenetre2 {
6
7     public static void main(String[] args) {
8         // TODO 1. Créer une JFrame
9         JFrame maFenetre1 = new JFrame("Ma première fenêtre SWING avec Objis");
10
11         // TODO 2. Modifier le titre de la fenêtre
12         //maFenetre1.setTitle();
13
14         // TODO 3. Modifier la taille (400 * 300)
15         maFenetre1.setSize(400, 300);
16
17         // TODO 4. taille non modifiable par l'utilisateur
18         maFenetre1.setResizable(false);
19
20         // TODO 5. Un click sur croix entraîne fermeture de la fenêtre
21         maFenetre1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
22
23         // TODO 6. Centrer la fenêtre par rapport à l'écran de l'ordinateur
24         maFenetre1.setLocationRelativeTo(null);
25
26         // TODO 7. Afficher la JFrame
27         maFenetre1.setVisible(true);
28
29     }
30 }
31
```



Constructeur

- Un constructeur est la méthode d'initialisation d'un objet
 - Le nom de cette méthode est le même que le nom de la classe et elle n'a pas de type de retour
 - Il sert à initialiser l'état de l'objet qu'il crée
 - Une classe peut comporter plusieurs constructeurs ayant des arguments différents
- On fait appel au constructeur d'un objet avec le mot clé *new*
 - *new Employe("Bernard")* : indique que l'on appelle un constructeur de la classe Employe, qui prend un argument de type *String*

Instanciation

- Le constructeur permet de définir un état de l'objet suite à sa création
- Cela permet de gérer l'état en plusieurs lignes (en passant par des setters)
- Instanciation se fait avec le mot clé **new**
- Dans 'MonApplication.java', coder :
 - `Employe employe1 ;` // 1. déclaration
 - `employe1 = new Employe('Pierre','DUPOND') ;` // 2. instanciation
 - `employe1.demandeFormation() ;` // 3. utilisation

Exemple de constructeur Et utilisation mots clés 'this' et 'super'

```
public class Employe
{
    private String nom;
    private String prenom;

    public Employe(){
        super();
    }

    public Employe(String nom){
        super();
        this.nom = nom; // this désigne l'objet courant (cette instance d'Employe)
    }

    public Employe(String nom, String prenom){
        super();
        this.nom = nom;
        this.prenom = prenom;
    }
}
```

Fait référence
au Constructeur de la
classe parente
ici de la classe Object)

Fait référence à
l'instance courante

Demo 1

```
MonApplication.java x
1 public class MonApplication {
2
3     public static void main (String arg[]) {
4         System.out.println ("Bienvenue dans mon application de gestion ressources humaines");
5
6         Employe employe1 = new Employe() ;
7
8         System.out.println ("Employe 1 : " + employe1.getNom() + " " + employe1.getPrenom() ) ;
9
10        employe1.setPrenom("Serge") ;
11        employe1.setNom("KOMI") ;
12
13        System.out.println ("Employe 1 : " + employe1.getNom() + " " + employe1.getPrenom() ) ;
14    }
15 }
```

Demo 2

```
MonApplication2.java
1 public class MonApplication2 {
2
3     public static void main (String arg[]) {
4         System.out.println ("Bienvenue dans mon application de gestion ressources humaines");
5
6         Employe employe1 = new Employe("KOMI", "Serge") ;
7
8         System.out.println ("Employe 1 : Nom " + employe1.getNom() + " prenom : " + employe1.getPrenom() ) ;
9     }
10 }
```

Questions Module1-Séance12

- **Citez les 5 repères clés du développeur Java**
- **Qu'est ce qu'un Constructeur ?**
- **Citez 2 caractéristiques clés constructeur ?**
- **A qui sert un constructeur ?**
- **Possible d'instancier une classe qui n'a pas de constructeur ?**
- **Utilité d'avoir plusieurs Constructeurs dans une même classe ?**
- **Combien de constructeur pour classe File ?**
- **Mettez en oeuvre 2 constructeurs de File**
- **Qu'est ce que le mot clé super ?**
- **Proposez un code complet classe Voiture**