



Le langage Java

Constructeurs

Objectifs Pédagogiques

À l'issue de cette formation, vous serez en mesure de :

- ✓ Comprendre la structure d'un objet
- ✓ Savoir instancier un objet

Programme détaillé ou sommaire

Création d'objet/instance

Rôle du constructeur

Utilisation d'un constructeur

Exemple de constructeurs

Mot clé this

Appel d'un autre constructeur avec this(...)

Destruction d'objets

Création d'objet/instance

Mot-clé : **new** doit être suivi par le nom de la classe et des parenthèses ()

Exemple : `new AdressePostale () ;`

Le mot clé **new** précède l'utilisation d'un constructeur:

Exemple : `AdressePostale adresse = new AdressePostale () ;`

Rôle du constructeur

Il sert à créer des objets à partir d'une classe.

Le constructeur fait partie de la catégorie des méthodes (d'où les parenthèses).

```
public class Emprunt {  
    public Emprunt() {  
        System.out.println("Coucou je viens d'être invoqué");  
    }  
}
```

```
Emprunt emp1 = new Emprunt();
```

```
Emprunt emp2 = new Emprunt();
```

Obligatoire :
Le
constructeur a
le même nom
que la classe
!

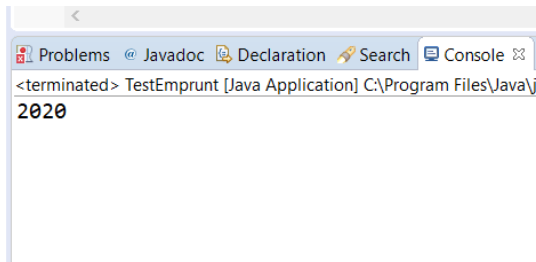
```
Console Problems Javadoc Declaration Search Servers History Git Staging Call Hierarchy  
<terminated> TestEmprunt (2) [Java Application] C:\Program Files\Java\jdk-11.0.7\bin\javaw.exe (6 oct. 2021 à 16:57:04)  
Coucou je viens d'être invoqué  
Coucou je viens d'être invoqué
```

Rôle du constructeur

Lorsque la classe a des variables, il permet de les initialiser

```
public class Emprunt {  
  
    int annee;  
  
    Emprunt() {  
        annee = 2020;  
    }  
}
```

```
public class TestEmprunt {  
  
    public static void main(String[] args) {  
        Emprunt emp = new Emprunt();  
        System.out.println(emp.annee);  
    }  
}
```



3 types de constructeurs

Les 3 types de constructeurs:

- ❑ Le constructeur **par défaut**
- ❑ Le constructeur **sans paramètre**
- ❑ Le constructeur **avec paramètre**

Le constructeur par défaut

Si vous n'avez pas mis de constructeur dans une classe, java insère un constructeur par défaut, appelé aussi constructeur implicite.

```
public class Emprunt {
```

```
}
```

```
public class TestEmprunt {
```

```
    public static void main(String[] args) {
```

```
        Emprunt emp = new Emprunt();
```

```
    }
```

```
}
```



Le constructeur sans paramètre

Le constructeur sans paramètre s'utilise comme le constructeur par défaut.

```
public class Emprunt {  
    int annee;  
  
    Emprunt() {  
        annee = 2020;  
    }  
}
```

```
public class TestEmprunt {  
  
    public static void main(String[] args) {  
        Emprunt emp = new Emprunt();  
    }  
}
```



Le constructeur avec paramètre

Le constructeur avec paramètre permet de passer des paramètres à un constructeur.

```
public class Emprunt {
```

```
    int annee;
```

```
    Emprunt(int nvAnnee) {
```

```
        annee = nvAnnee;
```

```
    }
```

```
}
```

```
public class TestEmprunt {
```

```
    public static void main(String[] args) {
```

```
        Emprunt emp = new Emprunt(2020);
```

```
    }
```

```
}
```

Le constructeur exige désormais
un paramètre de type entier



Le constructeur avec paramètre

Autre exemple qui ne compile pas...

```
public class Emprunt {
```

```
    int annee;
```

```
    Emprunt(int nvAnnee) {
```

```
        annee = nvAnnee;
```

```
    }
```

```
}
```

```
public class TestEmprunt {
```

```
    public static void main(String[] args) {
```

```
        Emprunt emp = new Emprunt();
```

```
    }
```

```
}
```

Le constructeur exige désormais
un paramètre de type entier



Création d'objet/instance

Le mot clé **new** précède l'utilisation d'un constructeur:

Exemple : `AdressePostale monAdresse = new AdressePostale();`

Un constructeur peut avoir des paramètres


Exemple: `AdressePostale monAdresse = new AdressePostale(5, "Bld", "Des maréchaux", 44100, "Nantes");`

Définition d'un constructeur

- **Même nom que la classe**
- Nombre quelconque d'arguments
- Plusieurs constructeurs par classe

Permet de valoriser les attributs de la classe.

```
public class Personne {  
  
    private String nom;  
    private String prenom;  
  
    public Personne(){  
  
    }  
  
    public Personne(String nvNom, String nvPrenom){  
        nom = nvNom;  
        prenom = nvPrenom;  
    }  
}
```



Utilisation d'un constructeur

➤ Avec l'opérateur new

- Si la classe possède plusieurs constructeurs, c'est au développeur d'en choisir un.

➤ Dans l'exemple ci-contre:

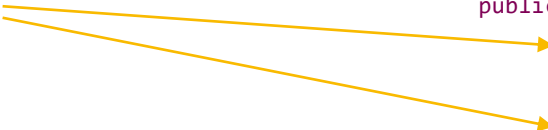
- ❑ la première personne **pers1** est construite avec le premier constructeur sans paramètre.

- Les attributs de **pers1** sont **non valorisés**

- ❑ La seconde personne **pers2** est construite avec le second constructeur.

- Les attributs de **pers2** sont **valorisés**

```
public class TestPersonne {  
  
    public static void main(String[] args) {  
        Personne pers1 = new Personne();  
  
        Personne pers2 = new Personne("Durand", "Paul");  
  
    }  
}
```



Donner une valeur par défaut

- On peut donner une valeur par défaut à une variable dans un constructeur.

```
public class CartePostale {  
  
    Photo photo;  
    String texte;  
    Adresse adresse;  
  
    CartePostale() {  
        texte = "Nous passons de bonnes vacances";  
    }  
  
    CartePostale(String nvtexte) {  
        texte = nvtexte;  
    }  
}
```

Mot clé this

Pour lever les ambiguïtés de nommage, **this** désigne l'objet.

```
class CartePostale {  
    Photo image;  
    String texte;  
    Adresse adresse;  
  
    CartePostale() {  
        texte = "Nous passons de bonnes vacances";  
    }  
  
    CartePostale(String texte) {  
        → this.texte = texte;  
    }  
}
```


Destruction d'objet

Automatique en Java

une fois que l'objet n'est plus référencé
pas de destructeur (C++)

Effectuée par le **garbage collector**

libère la mémoire
processus de faible priorité

Annexes



Chaînage de constructeurs

Méthode **this(...)**

La méthode `this` permet d'appeler un autre constructeur.

```
public class CartePostale {  
    Photo photo;  
    String texte;  
    Adresse adresse;  
  
    CartePostale() {  
        this("Nous passons de bonnes vacances");  
    }  
    ↓  
    CartePostale(String texte) {  
        this.texte = texte;  
    }  
}
```

Atelier (TP)

OBJECTIFS : Créer des constructeurs et les utiliser

DESCRIPTION :

- Dans le TP n°2 vous allez devoir créer des constructeurs et les utiliser.