



Constructeurs

Objectifs

- Concept de constructeur
- Les différents constructeurs
- Appels des constructeurs

Constructeur

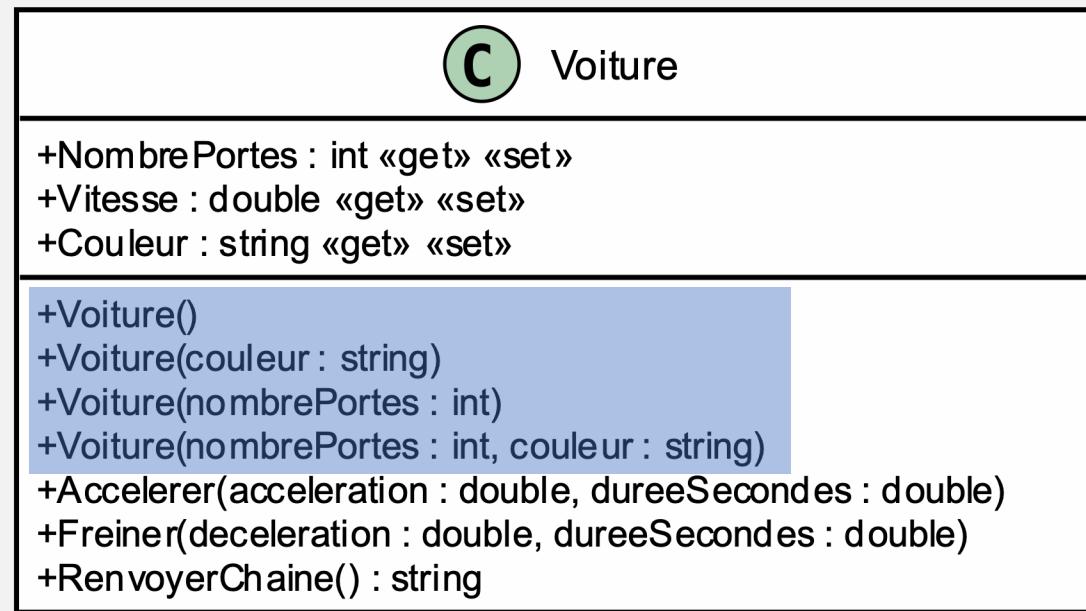
- Un constructeur est une méthode particulière qui :
 - Porte le nom de la classe
 - N'a pas de type de retour
 - Est exécutée pendant l'exécution du new juste après l'allocation de la mémoire
- Par défaut, le compilateur crée un constructeur par classe déclarée, c'est ce constructeur qui est exécuté quand vous faites : « new <NomDeLaClasse>() »
- Le constructeur par défaut est donc sans paramètre

Constructeur

- Les constructeurs avec paramètres sont appelés constructeurs d'initialisation
- La méthode « Initialiser » sera remplacée par ces constructeurs
- Dépendamment des langages, on parle aussi de :
 - Constructeur par copie : permet de copier un objet
 - Constructeur de conversion : permet de convertir un objet en un autre
- À la fin de l'exécution du constructeur, l'objet doit être dans un état cohérant

Classe des exemples

- Nous allons reprendre la classe Voiture décrite dans le module 02
- **Les préconditions ne sont pas écrites afin de simplifier le code affiché**



Constructeur par défaut

- Le but d'initialiser l'état de l'objet avec les valeurs par défaut

```
public Voiture()  
{  
    ;  
}
```

Équivalent du code
généré
automatiquement par
le compilateur

```
public Voiture()  
{  
    this.Couleur = "Noire";  
    this.NombrePortes = 4;  
    this.Vitesse = 0;  
}
```

Exemple de code du
constructeur par défaut
pour la classe Voiture
du Module 02

Constructeurs d'initialisation

```
public Voiture(string p_couleur)
{
    this.Couleur = p_couleur;
    this.NombrePortes = 4;
    this.Vitesse = 0;
}
```

```
public Voiture(int p_nombrePortes)
{
    this.Couleur = "Noire";
    this.NombrePortes = p_nombrePortes;
    this.Vitesse = 0;
}
```

```
public Voiture(int p_nombrePortes, string p_couleur)
{
    this.Couleur = p_couleur;
    this.NombrePortes = p_nombrePortes;
    this.Vitesse = 0;
}
```

Problème ici ?

- Beaucoup de répétition de code :
 - WET : We Enjoy Typing
 - Non respect du principe DRY (Don't Repeat Yourself)
- Comment résoudre ce problème ?
 - Appel d'une méthode « Initialiser » qui prend tous les paramètres
 - Utilisation du « this »
 - Utilisation des paramètres par défaut

Utilisation d'une méthode « Initialiser »

```
public Voiture()
{
    this.Initialiser(4, "Noire");
}

public Voiture(string p_couleur)
{
    this.Initialiser(4, p_couleur);
}

public Voiture(int p_nombrePortes)
{
    this.Initialiser(p_nombrePortes, "Noire");
}

public Voiture(int p_nombrePortes, string p_couleur)
{
    this.Initialiser(p_nombrePortes, p_couleur);
}

public void Initialiser(int p_nombrePortes, string p_couleur)
{
    this.Couleur = p_couleur;
    this.NombrePortes = p_nombrePortes;
    this.Vitesse = 0;
}
```

Utilisation du « this » – Simplification

```
public Voiture() : this(4, "Noire")
{ ; }

public Voiture(string p_couleur) : this(4, p_couleur)
{ ; }

public Voiture(int p_nombrePortes) : this(p_nombrePortes, "Noire")
{ ; }

public Voiture(int p_nombrePortes, string p_couleur)
{
    this.Couleur = p_couleur;
    this.NombrePortes = p_nombrePortes;
    this.Vitesse = 0;
}
```

Utilisation des paramètres par défaut

```
public Voiture(int p_nombrePortes = 4, string p_couleur = "Noire")
{
    this.Couleur = p_couleur;
    this.NombrePortes = p_nombrePortes;
    this.Vitesse = 0;
}
```

Quelle méthode utiliser ?

- Nous avons vu trois techniques :
 - Méthode « Initialiser » : à oublier car le langage propose des mécanismes standards
 - Utilisation du « this »
 - Utilisation des paramètres par défaut
- Vous pouvez utiliser l'une des deux dernières méthodes
 - Le choix dépend des pratiques de votre équipe projet
 - C'est généralement une question de gout

Appel des constructeurs 1 / 2

```
Voiture voiture1 = new Voiture();
```

```
Voiture voiture2 = new Voiture("Rouge");
```

```
Voiture voiture3 = new Voiture(3);
```

```
Voiture voiture4 = new Voiture(3, "Rouge");
```



```
public Voiture()
```



```
public Voiture(string p_couleur)
```



```
public Voiture(int p_nombrePortes)
```



```
public Voiture(int p_nombrePortes,  
               string p_couleur)
```

Appel des constructeurs 2 / 2

```
public Voiture(int p_nombrePortes = 4,  
               string p_couleur = "Noire")
```



```
Voiture voiture1 = new Voiture();  
  
Voiture voiture2 = new Voiture(p_couleur:"Rouge");  
  
Voiture voiture3 = new Voiture(3);  
  
Voiture voiture4 = new Voiture(p_nombrePortes: 3);  
  
Voiture voiture5 = new Voiture(3, "Rouge");
```