

## Extensibilité avec les delegates et les events

### Introduction

Cette fiche vous donne un plan d'action pour écrire des classes extensibles grâce aux delegates et aux events en CSharp.

### Plan d'action

#### Etape 1 : Commencez par choisir un type de mécanisme d'extensibilité

Vous pouvez choisir parmi les mécanismes suivants :

- Héritage : technique intéressante uniquement si la nouvelle classe propose des comportements similaires et nécessite donc d'hériter d'une classe parente.
- Surcharge de méthode : technique simple permettant d'ajouter une méthode avec des paramètres différents. Attention à ne pas faire trop de surcharges car cela rend le travail du développeur plus difficile par la suite (il ne saura pas forcément quelle surcharge il devra utiliser).
- Delegate pour une méthode : mécanisme intéressant pour mettre en place des points d'extensions ou des filtres pour une méthode.
- Event pour une classe : mécanisme intéressant pour mettre en place des points d'extensions ou des filtres au niveau d'une classe (cela pourra être utilisé par plusieurs méthodes de cette classe par exemple)

#### Etape 2 : Coder le mécanisme en commençant par préparer le terrain

- Commencer par écrire la classe ou la méthode en fonction du mécanisme à mettre en place.
- Ajouter ensuite l'événement ou le delegate si vous avez choisi cette technique.

#### Etape 3 : Terminer et finaliser le mécanisme

- Si vous avez choisi les delegates ou les events, il faudra peut-être écrire une classe spéciale pour échanger des paramètres. Vous pouvez écrire cette classe à cette étape.

#### Etape 4 : Faites le bilan

Vérifiez que le code que vous avez écrit est compréhensible pour un autre développeur.

Est-ce que le mécanisme que j'ai implémenté est intuitif ? Est-ce qu'il est documenté ?

Article d'origine : <http://www.dotnetdojo.com/extensibilite-delegates-delegates-multicast-events-csharp/>

[Dotnetdojo.com](#) – Améliorer votre potentiel de développeur.