

□ Syntaxe Objective C - Méthodes privées

- Il n'existe pas de support direct des méthodes privées au niveau du langage
- La solution passe par l'utilisation d'une "Category" spéciale appelée "Class Extensions"
- Une catégorie permet de rajouter des méthodes à une classe existante (y compris à une classe dont on ne possède pas le code source)
 - permet d'étendre les fonctionnalités d'une classe sans avoir recours à l'héritage
 - permet également de répartir le code source d'une classe importante sur plusieurs fichiers ou d'ajouter des fonctionnalités à une classe d'un framework depuis un autre framework

□ Syntaxe Objective C - Méthodes privées

□ Category - exemple

- UITableView (définie dans le framework UIKit) ajoute des propriétés row et section à la classe NSIndexPath (définie dans le framework Foundation) pour manipuler plus facilement cet index dans le contexte des vues de type liste

```
// This category provides convenience methods to make it easier to use an
NSIndexPath to represent a section and row
@interface NSIndexPath (UITableView)

+ (NSIndexPath *)indexPathForRow:(NSInteger)row inSection:(NSInteger)section;

@property(nonatomic, readonly) NSInteger section;
@property(nonatomic, readonly) NSInteger row;

@end
```

□ Syntaxe Objective C - Méthodes privées

□ Class extension (catégorie anonyme)

- A la différence d'une catégorie classique le compilateur vérifie que les méthodes déclarées dans la catégorie sont implémentées

```
@interface MyClass : NSObject  
@property (readonly) float value;  
  
@end
```



```
@interface MyClass ()  
@property (readwrite) float value;  
  
@end  
  
@implementation MyClass  
  
...  
  
@end
```



□ Exercice

- Créer une application dont l'interface graphique présente un "slider" et 4 labels
- Le slider permet de modifier l'âge d'un objet de type Person
- Le label de l'âge et du droit de vote doivent se mettre à jour en temps réel
- L'âge ne peut prendre que des valeurs entières
- Dans un deuxième temps l'âge ne pourra prendre que des valeurs multiples de 5 et le curseur devra se trouver à la position correspondante

Person
<code>NSString *name;</code>
<code>UInt8 age;</code>
<code>– (id) initWithAge:(UInt8)newAge;</code>
<code>+(Person *)createPersonWithAge:(UInt8)newAge;</code>
<code>– (BOOL)canLegallyVote;</code>



Penser à vérifier les fuites mémoire

□ Exercice

- Créer un nouveau projet et réutiliser la classe Person
- Ajouter une propriété spouse de type Person à la classe Person
 - Donner l'attribut retain à cette propriété
- Créer une vue contenant un bouton qui crée 2 personnes mariées l'une à l'autre à chaque appui
- Analyser les fuites mémoires (afficher les "retain cycles")

