

Aufgabe 1

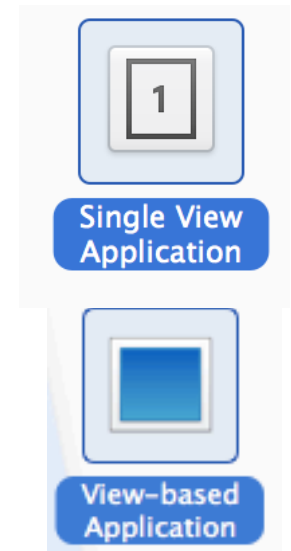
Erstellen einer Klasse

Ziel:

- ◆ Eine Klasse „Article“ ist zu erstellen.
 - Die Attribute der Klasse sollen bei Programmstart auf der Console ausgegeben werden.

Weg:

- ◆ Erstelle ein neues **View-basiertes Projekt** in Xcode.
(iOS → Application → View-based Application).
 - für iPhone
 - ohne Unit Tests
 - **ohne** Storyboards
- ◆ Erstelle eine **Klasse „Article“**
(iOS → Cocoa Touch → Objective-C Class).
 - Erzeuge hierzu in dem Ordner (Group), der so heißt wie dein Projekt, eine neue Datei.
 - Die Klasse soll von *NSObject* erben.





Aufgabe 1: Erstellen einer Klasse

◆ Article-Objekte sollen folgende **Attribute** haben:

- *titel* des Artikels als *NSString**,
- *text* des Artikels als *NSString**,
- *url* zum Artikel als *NSURL**-Objekt,
- *datum* der Veröffentlichung als *NSDate**-Objekt.

Hinweis: Klassen groß schreiben: *BeispielKlasse*

Methoden und Variablen klein: *beispielMethode*, *beispielVariable*

◆ Die Attribute sollen über **Getter** und **Setter** zugänglich sein.

- Versuche zuerst, einige Get- und Set-Methoden per Hand zu implementieren.
- Versuche außerdem, die Attribute per @property und @synthesize zu erstellen.

Aufgabe 1: Erstellen einer Klasse

- ◆ Importiere den **Header** deiner Klasse **in das AppDelegate**.
- ◆ Erstelle in der Methode „**didFinishLaunchingWithOptions**“ des **AppDelegate** Objekte deiner Klasse „Article“.
 - Lasse dir einige Werte auf der Console mit **NSLog(...)** ausgeben.

Hinweis: **Kompilieren** mit **[cmd]+[b]**

Starten mit **[cmd]+[r]**

- ◆ Schreibe eine eigene **Init-Methode** in deine Klasse „Article“, die als zusätzliche Parameter *titel*, *url* und *text* enthält.
 - Erstelle deine Objekte mit dieser Init-Methode.
- ◆ Implementiere eine **Methode**, die eine Beschreibung des Artikels zurückgibt („Name: Ein Artikel, erstellt: 10.10.2012 etc.“)
 - **NSString** bietet hilfreiche Methoden dafür an (siehe Xcode-Doku).

Hinweis: **Dokumentation** zu Elementen im Quellcode: **[alt]+Klick** auf das Element bzw. **[strg]+[alt]+[cmd]+[/]** bei selektiertem Element.

Aufgabe 2

Erstellen einer View

Ziel

- ◆ Eine **Oberfläche** (View) für das Anlegen von Artikeln ist zu erstellen.
- ◆ Ein Artikel soll folgende **Attribute** haben:
 - *Titel* des Artikels,
 - *Text* des Artikels,
 - *URL* zum Artikel,
 - *Datum* (optional).
- ◆ Der **ViewController ist der MVC-Controller** und soll das Modell (den Artikel) verwalten.
- ◆ Beim Drücken auf den **Speichern-Button** sollen die Daten aus der UI in das Modell (Artikel) übertragen und auf der Console ausgegeben werden.



mögliche Oberfläche

Weg

- ◆ Öffne die bereits zu deinem View-basierten Projekt angelegte **Nib-Datei** (.xib) und erstelle die Oberfläche.
 - **Achtung:** Nicht die MainWindow.xib verändern!
- ◆ Verknüpfe **grafische Elemente** über **IBOutlets** mit dem Quellcode.
 1. Dekлариere und synthetisiere die IBOutlets im Quellcode.
 2. Verknüpfe die grafischen Elemente im InterfaceBuilder mit den IBOutlets.
 3. Die IBOutlets gehören in den MVC-Controller (ViewController).
- ◆ Verknüpfe den Save-**Button** über eine **IBAction** mit dem Quellcode.
 1. Dekлариere und implementiere die IBAction im Quellcode.
 2. Verknüpfe das grafische Element im InterfaceBuilder mit der IBAction
 3. In der IBAction sollen die Daten aus den Outlets in das Modell übertragen werden und anschließend auf der Console ausgegeben werden. → Dazu braucht der View-Controller eine Referenz auf das Modell (Ein Artikel-Objekt)
 4. Die IBActions gehören in den ViewController.

- ◆ Die **Tastatur** soll beim drücken auf den return-Button wieder ausgeblendet werden.
 - Erstelle dazu eine IBAction für die Textfelder.
 - Verknüpfe das Event „**Did End On Exit**“ der Textfelder mit der IBAction.
 - Verschicke aus der IBAction die Nachricht „**resignFirstResponder**“ an das jeweilige Textfeld.
 - Hinweis: Das Textfeld übergibt beim Aufruf der Action-Methode eine Referenz auf sich selbst.

- ◆ Optional: Ein **Date-Picker** kann über **eine weitere View** eingebunden werden.
 - Die neue View + ViewController kann über „**presentModalViewController**“ angezeigt werden.

```
[self presentModalViewController:datePickerController animated:YES];
```