

# Домашнее задание №1

## 1. Подготовка

Как было сказано на лекции, на протяжении всего курса мы будем разрабатывать приложение-чат. И для начала тебе нужно создать проект и настроить его:

- iOS Deployment Target - 13.0
- Devices - iPhone (iPad можно по желанию)
- Device Orientation - Portrait (Landscape по желанию)
- Добавь AppIcons по своему усмотрению, используя подготовленный набор в архиве с заданием.
- Так как наше приложение не будет поддерживать многооконный режим (сцены), нужно удалить SceneDelegate, чтобы события жизненного цикла обрабатывались в AppDelegate :
  - Удалить файл `SceneDelegate.swift`
  - В файле `Info.plist` удалить `Application Scene Manifest`
  - В классе `AppDelegate` :
    - удалить методы `UISceneSession Lifecycle`
    - добавить свойство `var window: UIWindow?`

## 2. Жизненный цикл приложения

Теперь нужно закрепить знания о жизненном цикле приложения и понять, как приложение переходит из одного состояния в другое. Тебе нужно будет вывести в консоль сообщения для каждого состояния. Для этого:

- В файле `AppDelegate.swift` в каждый метод жизненного цикла добавь вывод в консоль названия этого метода. Для этого можно использовать литерал `#function`. К выводу названия метода добавь вывод состояний, при переходе между которыми он вызывается. Вывод в консоли должен выглядеть примерно так:

```
Application moved from <стейт из которого переходим> to <стейт в который переходим>:  
<название метода>
```

- Сверни/разверни и закрой приложение - в консоль должны выводиться названия вызываемых при этом методов `AppDelegate`.

### 3. Жизненный цикл UIViewController

Также на лекции мы рассмотрели жизненный цикл `UIViewController`. Для того, чтобы лучше его понять:

- Добавь во ViewController следующие методы:
  - `viewWillAppear`
  - `viewDidAppear`
  - `viewWillLayoutSubviews`
  - `viewDidLayoutSubviews`
  - `viewWillDisappear`
  - `viewDidDisappear`
- Не забудь проставить вызовы `super` методов (прим. `super.viewWillAppear(animated)`)
- В каждый метод добавь вывод в консоль название вызываемого метода (аналогично, как это было сделано ранее в `AppDelegate`).

### 4. Git

Выполненное домашнее задание нужно будет выложить в свой репозиторий на Github Classroom. Для этого у тебя должны быть начальные знания `git` и понимание, как с ним работать. Если ты еще не знаком(а) с `git`:

- Прочитай основную информацию [здесь](#)
- Прочитай про основные команды, которые тебе потребуются:
  - `git add`
  - `git commit`
  - `git push`
  - `git pull`
  - `git merge`
- [Прочитай про](#) `.gitignore` :

- `.gitignore` файл можно сгенерировать с помощью сервиса [gitignore.io](https://gitignore.io)
- добавь файл `.gitignore` в свой проект.

## 5\* Задание со звездочкой

Как можно заметить, в домашнем задании тебе придется **логировать** события жизненного цикла. Попробуй реализовать решение, позволившее бы включить и отключить вывод логов в консоль перед компиляцией (в интерфейсе приложения не надо добавлять никаких кнопок для этого).

## 6. Как получить максимальный балл?

Для этого нужно соблюсти следующее:

- Проект должен собираться без ошибок и warning'ов
- Проект не должен падать (крашиться)
- Файл `.gitignore` заполнен правильно
- События жизненного цикла логируются корректно
- Выполнено задание со звездочкой