

## //classes

macOS  $\Rightarrow$  command Line tool

//

Properties:

```
let age = 10  
let name = "James"
```

Actions:

```
func sing()
```

Events:

```
onResample()
```

// main.swift (1)

```
import Foundation
```

```
let james = Musicians () // create object
```

```
james.age = 50
```

```
james.name = "James Hetfield"
```

```
james.instrument = "Guitar"
```

```
print(james.age)
```

// new file  $\Rightarrow$  swift  $\Rightarrow$  Musicians.swift (1)

```
import Foundation
```

```
class Musicians {
```

```
var name = ""
```

```
var age = 0
```

```
var instrument = ""
```

}

## // Musicians.swift (2)

```
import Foundation
```

enum MusicianType { //enums  $\Rightarrow$  easier to categorize things

```
case LeadGuitar  
case Vocalist  
case Drummer
```

}

```
class Musicians {
```

// property

```
var name: String  
var age: Int  
var instrument: String  
var type: MusicianType
```

## // initializers (constructor)

```
init (nameInit: String, ageInit: Int, instrumentInit: String, typeInit: MusicianType) {  
    name = nameInit  
    age = ageInit  
    instrument = instrumentInit  
    type = typeInit
```

}

```
func sing() {
```

```
    print("nothing else matters")
```

}

## // main.swift (2)

```
import Foundation
```

```
let james = Musicians (nameInit: "James", ageInit: 50, instrumentInit: "Guitar", typeInit: Drummer)
```

```
print(james.age)
```

Do not forget the dot

## // new file $\Rightarrow$ SuperMusician.swift $\Rightarrow$ inheritance

```
import Foundation
```

```
class SuperMusician: Musician {
```

```
func sing2() {
```

```
    print("enter night")
```

}

```
override func sing() {
```

```
    super.sing() // super is a Musicians class
```

```
    print("exit light")
```

}

## 11 main.swift (3)

import Foundation

```
let kirk = SuperMusicians(nameInit: "kirk", ageInit: 50, instrumentInit: "Guitar", typeInit: Drummer)
print(kirk.smg) // will print "nothing else matters" and "exit light"
```

## 11 Access levels

private → same class only, can not be used in the object, can not be inherited

fileprivate → same file, not only from the same class

internal → default one,

rarely used public → can be reached from everywhere, but can not be changed ⇒ mostly external library

open ⇒ reached from anywhere in the project, can be modified from everywhere

## 11 Simpson Book

### 1 DetailsVC.swift

import UIKit

```
class DetailsVC: UIViewController {
```

@IBOutlet weak var imageView: UIImageView!

↓ similar two

var selectedSimpson: Simpson?

override func viewDidLoad() {

super.viewDidLoad()

nameLabel.text = selectedSimpson?.name

jobLabel.text = selectedSimpson?.job

imageView.image = selectedSimpson?.image

}

### 11 ViewController.swift

var chosenSimpson: Simpson?

```
func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
```

chosenSimpson = mySimpsons[indexPath.row]

self.performSegue(withIdentifier: "toDetailVC", sender: nil)

}

```
override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
```

if segue.identifier == "toDetailsVC" {

let destinationVC = segue.destination as! DetailsVC

destinationVC.selectedSimpson = chosenSimpson

}

### 11 Simpson.swift

import Foundation

import UIKit // in order to have UIImage available

```
class Simpson {
```

var name: String

var job: String

var image: UIImage

```
init (simpsonName : String, simpsonJob : String, simpsonImage : UIImage) {  
    name = simpsonName  
    job = simpsonJob  
    image = simpsonImage
```

3

3