Class Methods

בשיעור זה נכיר

- הגדרת ושימוש מתודות מחלקה
- הגדרת ושימוש משתני מחלקה
- המתודה load של NSObject
- sharedInstance

Class Methods

גם מחלקה היא אובייקט, וגם איתה ניתן לדבר.
על כן ישנן מתודות מחלקה.

על פי רוב מתודות מחלקה הינן מתודות "סטטיות" שתוכנן ופעולתן אינן משתנות על פי עצמים כאלה ואחרים.

ופעולתה קשורה לכל עצמי המחלקה או לאף אחד מהם.

הצהרה על מתודת מחלקה תפתח במילה class func

Class Methods

דוגמא

```
class AirPlane{
    class func getMaxHeight() -> Double{
        return 5000
    }

    class func canFlightInZone(#zoneID : UInt) -> Bool{
        if zoneID > 1000{
            return true
        } else {
            return false
        }
    }

    class func saySomething(){
        print("something")
    }
}
```

```
AirPlane.saySomething()

let maxHeight = AirPlane.getMaxHeight()

let zone : UInt = 997
let canFlight = AirPlane.canFlightInZone(zoneID: zone)
```

Class Methods

משתני מחלקה - ניתן להגדיר משתנים גלובליים ברמת המחלקה.

משתני מחלקה נגישים עבור מתודות מחלקה.

וכן נגישים לקריאה חיצונית ובפרט על ידי עצמי המחלקה.

יש להוסיף את המילה class לפני המשתנה.
שימו לב: נכון להיום ב xCode6 beta 3, הדבר טרם נתמך

```
class AirPlane{
    class var num : Int = 5          ! Class variables not yet supported
    class func getMaxHeight() -> Double{
```

Class Methods

כל האמור לעיל, רלוונטי גם ל Value-Types כדוגמת struct
אלא שבמקום המילה class
יש להשתמש במילה static

```
struct GameLevel{
    static let minCreditsInLevel : Int = 3
    static let maxCreditsInLevel : Int = 15

    static func getCredits() -> Int{
        let range : Int = maxCreditsInLevel - minCreditsInLevel
        var credits : Int = Int(rand())%range

        credits += minCreditsInLevel

        return credits
    }
}

let credits = GameLevel.getCredits()
```

NSObject's load

למחלקה NSObject ישנה מתודה שמקבלת קריאה עוד לפני שהאפליקציה עולה.
היא שימושית מאד בשביל אתחולים עצמאיים של מחלקות, על השימושים השונים.
אין צורך לשלוח קריאה ל load.
היא נקראת לבד.

```
class AppManager : NSObject{
    override class func load(){
        super.load()
        self.perfromActionsForFirstLoad()
    }

    class func perfromActionsForFirstLoad(){

    }
}
```

Shared-Instance

בכל אפליקציה שהיא מורכבת דיה, ראוי כי תהיה לפחות מחלקה אחת שמנהלת פעולות באפליקציה.
בדרך כלל מדובר על חלק מה model

עצם מהמחלקה המנוהלת, אם כן אמור להיות יחיד במינו, ונרצה תמיד לעבוד אל מול אותו עצם ואל מול ערכיו.

על כן נדרש לנהל shared-instance, יש הקוראים אותו singleton

נראה דוגמא

```
var LDB_instance : LocalDatabaseManager?

class LocalDatabaseManager : NSObject{
    class func sharedInstance() -> LocalDatabaseManager{
        if LDB_instance == nil{
            LDB_instance = LocalDatabaseManager()
        }
        return LDB_instance!
    }

    init(){
        //open database file
    }

    deinit{
        //close database file
    }

    func addObject(obj : AnyObject){

    }

    func updateObject(obj: AnyClass, atIndex index : UInt){

    }

    func removeObject(obj: AnyClass, atIndex index : UInt){

    }
}
```

Class Methods

שאלות?