

Protocol

בשיעור זה נכיר

- מהו פרוטוקול
- שימוש בפרוטוקול
- כתיבת פרוטוקול

Protocol

הגדרה - פרוטוקול

פרוטוקול הוא אוסף הצהרות ותכונות לא ספציפי למחלקה מסויימת או למבנה מסויים

חלק מהצהרות הפרוטוקול הינן חובה
וחלק הינן רשות

המחלקה הרוצה לאמץ (conform) פרוטוקול מסויים, צריכה להצהיר עליו ולממש אותו.

ניתן להתייחס לאובייקט מבלי לציין את מחלקתו, אלא רק בציון הפרוטוקול אותו הוא מממש

Protocol

פרוטוקולים נפוצים

- UITextFieldDelegate ●
- UITableViewDataSource ●
- UITableViewDelegate ●
- UIImagePickerControllerDelegate ●
- UIApplicationDelegate ●

Protocol

הפרוטוקול מגדיר יכולות למחלקה המממשת.

צורת העבודה היא כדלהלן.

אובייקט א מחזיק משתנה ממחלקה לא ידועה, המצהיר על פרוטוקול מסויים.
אובייקט ב מצהיר על הפרוטוקול ומממש אותו כראות עיניו.

אובייקט א שולח קריאות מתוך הפרוטוקול לאובייקט ב ומתייחס לתשובותיו במידת הצורך כראות עיניו.

המחלקה המממשת יכולה לנטר אירועים ולקבוע תהליכים עבור האובייקט הקורא.

Protocol

נראה דוגמא לפרוטוקול

```
protocol FighterInspector{  
    var insperctorID : Int {get set}  
    func fighterCanTakeOff(fighter : Fighter) -> Bool  
    func fighterCanBeOnHeight(fighter : Fighter, height : Double) -> Bool  
    func fighterChangedToHeight(fighter : Fighter, height : Double)  
}
```

Protocol

```
class Fighter{
  //...
  var insperctor : FighterInspector?
  var height : Double = 0.0{
    didSet{
      insperctor?.fighterChangedToHeight(self, height: self.height)
    }
  }

  func takeOff(){
    if (insperctor?.fighterCanTakeOff(self)){
      //taking off
    }
  }

  func getToHeight(height : Double){
    if (insperctor?.fighterCanBeOnHeight(self, height: height)){
      //getting to height
    }
  }
}
```

ניתן דוגמא למחלקה Fighter

שיש לה property שלא ידוע מאיזו מחלקה הוא,

אך הוא אמור לממש את

הפרוטוקול FighterInspector

אם אינו nil

Protocol

והנה Person אשר מצהיר ומממש
את הפרוטוקול FighterInspector

```
class Person : FighterInspector{  
    var insperctorID : Int = 5000  
    func fighterCanTakeOff(fighter : Fighter) -> Bool{  
        return true  
    }  
  
    func fighterCanBeOnHeight(fighter : Fighter, height : Double) -> Bool{  
        return (height < 2000)  
    }  
  
    func fighterChangedToHeight(fighter : Fighter, height : Double){  
        print("height is \"(height)\")  
    }  
}
```

Protocol

במידה ותחשוק נפשכם במתודות ותכונות שאינם בעלי חובת מימוש
תאלצו ליצור פרוטוקול Objective-C שכל המשתנים שבו הינם ממחלקות של Objective C
נראה דוגמאונת

```
@objc protocol SampleProtocol{  
    //Methods  
    func someRequiredMethodWithObject(value : NSObject) -> Bool  
    @optional func someOptionalMethodWithNumber(number : NSNumber) -> NSArray  
  
    //Properties  
    var someProperty : Int {get}  
    @optional var someOptionalArray : NSArray {get}  
}
```


Protocol

נבצע תרגיל.

מסך שמתקשר עם המסך הקודם לו, רגע לפני שהוא חוזר, או בעת אירוע מסויים.
כמובן, באמצעות פרוטוקול

Protocol

שאלות?