

Apple Swift kurzus

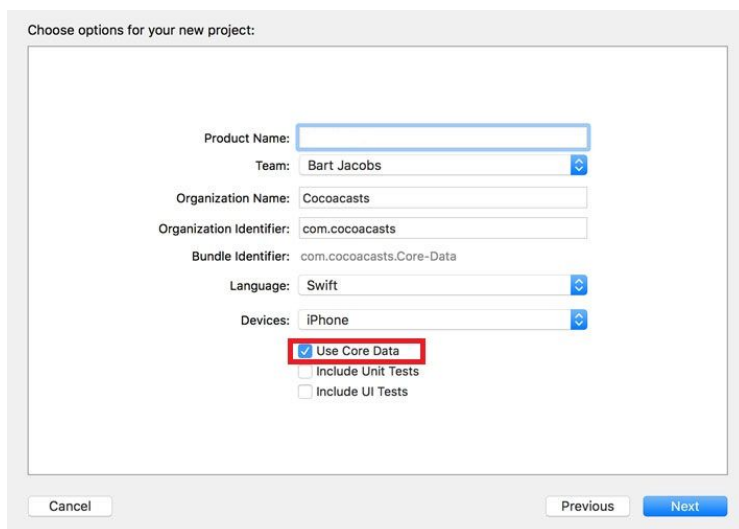
9. gyakorlat

Core Data:

- egy keretrendszer, amely menedzseli az alkalmazás model rétegbeli objektumait.
- rendelkezésre bocsát egy általánosított és automatizált megoldást az objektumok életciklusát, objektumgráfok menedzsmentjét ellátó feladatokra
- meg van valósítva a perzisztencia
- nincs szükség SQL kódok írására

Alkalmazás létrehozása:

- új projekt esetén pipáljuk be a **Use Core Data** négyzetet
~ így egy kibővített projektet kapunk (bővített *AppDelegate.swift*-tel és egy *.xcdatamodeld* fájlal)



~ az új projektünk neve semmiképp ne CoreData vagy Core Data vagy hasonlók legyenek, mert ebben az esetben nem találja a megfelelő importálandó lib-et

Core Data használata:

- a Core Data használatához szükséges importálni a CoreData lib-et
import CoreData
- az adatbázisunkhoz hozzunk létre egy adatmodellt
~ nyissuk meg a **ProjectName.xcdatamodeld** fájlt
~ hozzunk létre egy új entitást (Add Entity gomb)
~ adjunk attribútumot az entitásunkhoz (+), állítsuk be az adattag típusát (Integer, String, stb...)
- programkódban hozzunk létre egy új tömböt
var listItems = [NSObject]()
~ a tömb típusa NSObject elemeket tároló tömb, mert minden Core Data-ba bekerülő objektum típusa NSObject (továbbá így elérhető az adatbázis kód szinten is)
- adatbázisba **mentés** a kódban:
~ példányosítsunk egy **AppDelegate** objektumot, hogy kontextust hozzassunk létre, illetve használhassuk az AppDelegate objektumait és metódusait
let appDelegate = UIApplication.shared.delegate as! AppDelegate
~ hozzuk létre a kontextust - (ez kvázi "jegyzetfüzetként" szolgál, amin módosításokat végzünk)
let managedContext = appDelegate.persistentContainer.viewContext
~ keressük ki a megfelelő entitást az adatbázisból
let entity = NSEntityDescription.entity(forEntityName: "EntityName", in: managedContext)!

~ a kigyűjtött entitás alapján hozzunk létre egy új objektumot

```
let entityName = NSManagedObject(entity: entity,  
    insertInto: managedContext)
```

~ az elkészült objektumnak, ha szeretnénk valamelyik attribútumát módosítani, akkor az attribútum nevére, mint kulcsra kell hivatkoznunk (pl.: ha a neve **stringAttr**, és a típusa **string**, akkor így)

```
entityName.setValue("someString", forKeyPath: "stringAttr")
```

~ a kontextust el kell menteni (ezzel perzisztálunk)

```
do {  
    try managedContext.save()  
    listItems.append(entityName)  
} catch let error as NSError {  
    print("Could not save. \(error)")  
}
```

- adatbázisból olvasás a kódban

~ példányosítsunk egy **appDelegate** objektumot, hogy kontextust hozzassunk létre, illetve használhassuk az AppDelegate objektumait és metódusait

```
let appDelegate = UIApplication.shared.delegate as! AppDelegate
```

~ hozzuk létre a kontextust - (ez kvázi "jegyzetfüzetként" szolgál, amin módosításokat végzünk)

```
let managedContext = appDelegate.persistentContainer.viewContext
```

~ lekérdezéshez készítsunk egy Fetch request-et

```
let fetchRequest = NSFetchRequest<NSManagedObject>(entityName:  
    "entityName")
```

~ hajtsuk végre a fetch-elést, majd az eredményt tegyük bele az NSManagedObject-eket tartalmazó tömbünkbe (**listItems**)

```
do {  
    listItems = try managedContext.fetch(fetchRequest)  
} catch let error as NSError {  
    print("Could not fetch. \(error)")  
}
```

- adatbázisból törlés a kódban

~ példányosítsunk egy **appDelegate** objektumot, hogy kontextust hozzassunk létre, illetve használhassuk az AppDelegate objektumait és metódusait

```
let appDelegate = UIApplication.shared.delegate as! AppDelegate
```

~ hozzuk létre a kontextust - (ez kvázi "jegyzetfüzetként" szolgál, amin módosításokat végzünk)

```
let managedContext = appDelegate.persistentContainer.viewContext
```

~ ha a birtokunkban van a konkrét **NSManagedObject** objektum, amelyiket törölni szeretnénk, akkor ez egyből át is adható a **delete()** metódusnak paraméterként (pl.: a **listItems** tömb 6. elemét szeretném törölni az adatbázisból (*indexelés 0-tól indul!*))

```
managedContext.delete(listItems[5])
```

~ érdemes aktualizálni a **listItems** tömböt is

```
listItems.remove(at: 5)
```

~ a perzisztáláshoz itt is el kell menteni a kontextust

```
do {  
    try managedContext.save()  
} catch let error as NSError {  
    print("Could not save. \(error)")  
}
```