Swift Programlama Dili ile iOS Mobil Uygulama Geliştirme

MURAT ADIGÜZEL (murat.adiguzel@intertech.com.tr)

5 Nisan 2021

Koleksiyonlar

Diziler (Arrays), Sözlükler (Dictionaries) & Kümeler (Sets)

Koleksiyonlar

Big-O (Büyük-O) Notasyonu

“Ayşe”, “Ali”, “Kerem”

“Veli”, “Cenk”, “Zeynep”, “Mehmet”, “Gülizar”, “Ali”

“Ali”

Koleksiyonlar

Big-O (Büyük-O) Notasyonu

“Ayşe”, “Ali”, “Kerem”

“Veli”, “Cenk”, “Zeynep”, “Mehmet”, “Gülizar”, “Ali”

“Ali”

Koleksiyonlar

Big-O (Büyük-O) Notasyonu

“Ayşe”, “Ali”, “Kerem”

“Veli”, “Cenk”, “Zeynep”, “Mehmet”, “Gülizar”, “Ali”

“Ali”

O(1)

Koleksiyonlar

Big-O (Büyük-O) Notasyonu

“Ayşe”, “Ali”, “Kerem”

“Veli”, “Cenk”, “Zeynep”, “Mehmet”, “Gülizar”, “Ali”

“Ali”

Koleksiyonlar

Big-O (Büyük-O) Notasyonu

“Ayşe”, “Ali”, “Kerem”

“Veli”, “Cenk”, “Zeynep”, “Mehmet”, “Gülizar”, “Ali”

“Ali”

Koleksiyonlar

Big-O (Büyük-O) Notasyonu

“Ayşe”, “Ali”, “Kerem”

“Veli”, “Cenk”, “Zeynep”, “Mehmet”, “Gülizar”, “Ali”

O(n)

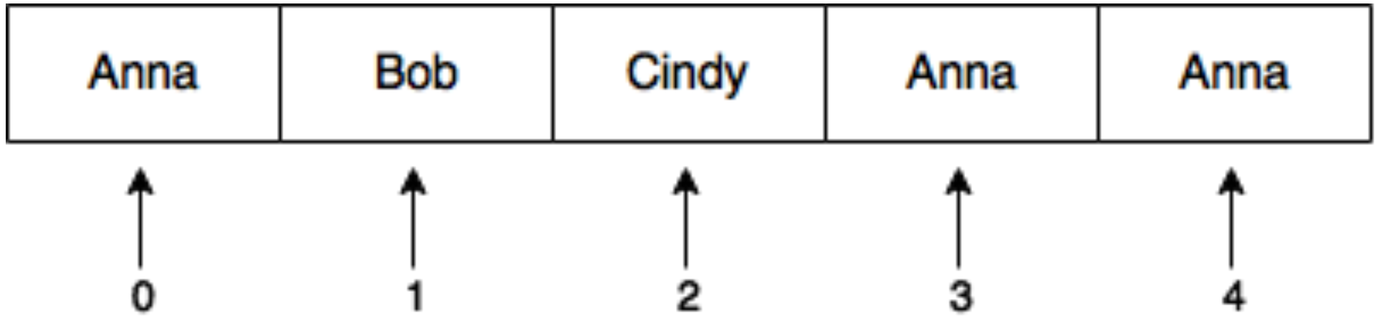
“Ali”

Koleksiyonlar

Diziler (Arrays)

Koleksiyonlar

Diziler (Arrays)



Koleksiyonlar

Dizi İşlemleri İçin Çalışma Süresi

- Diziler bellekte bitişik bir blok olarak saklanır.

Koleksiyonlar

Dizi İşlemleri İçin Çalışma Süresi

- Öğelere erişim

Koleksiyonlar

Dizi İşlemleri İçin Çalışma Süresi

- Öğelere erişim

O(1)

Koleksiyonlar

Dizi İşlemleri İçin Çalışma Süresi

- Öğe ekleme (Dizinin başına ekleme)

Koleksiyonlar

Dizi İşlemleri İçin Çalışma Süresi

- Öğe ekleme (Dizinin başına ekleme)

O(n)

Koleksiyonlar

Dizi İşlemleri İçin Çalışma Süresi

- Öğe ekleme (Dizinin ortasına ekleme)

Koleksiyonlar

Dizi İşlemleri İçin Çalışma Süresi

- Öğe ekleme (Dizinin ortasına ekleme)

O(n)

Koleksiyonlar

Dizi İşlemleri İçin Çalışma Süresi

- Öğe ekleme (Dizinin sonuna ekleme)

Koleksiyonlar

Dizi İşlemleri İçin Çalışma Süresi

- **Öğe ekleme (Dizinin sonuna ekleme)**

En iyi durumda ve ortalama **O(1)**

En kötü durumda **O(n)**

Koleksiyonlar

Dizi İşlemleri İçin Çalışma Süresi

- Öğe silme

Koleksiyonlar

Dizi İşlemleri İçin Çalışma Süresi

- Öğe silme

O(1) veya **O(n)**

Koleksiyonlar

Dizi İşlemleri İçin Çalışma Süresi

- Öğe arama

Koleksiyonlar

Dizi İşlemleri İçin Çalışma Süresi

- Öğe arama

O(n)

Koleksiyonlar

Egzersizler (Diziler)

1. `players` dizisindeki `"Dan"` öğesinin konumunu belirlemek için `firstIndex(of:)` yöntemini kullanın.

Koleksiyonlar

Egzersizler (Diziler)

1. Oyuncuların adlarını ve puanlarını yazdıran bir `for-in` döngüsü yazın.

Koleksiyonlar

Egzersizler (Diziler)

1. Aşağıdakilerden hangileri geçerli ifadelerdir?

```
let array1 = [Int]()

let array2 = []

let array3: [String] = []
```

Koleksiyonlar

Egzersizler (Diziler)

2. `let array4 = [1, 2, 3]`

   dizisi için aşağıdakilerden hangileri geçerli ifadelerdir?

```
print(array4[0])

print(array4[5])

array4[1...2]

array4[0] = 4

array4.append(4)
```

Koleksiyonlar

Egzersizler (Diziler)

3. Belirli bir tamsayının ilk varlığını bir tamsayı dizisinden kaldıran bir fonksiyon yazın. Fonksiyonun imzası şu şekildedir:

```
func removingOnce(_ item: Int, from array: [Int]) -> [Int]
```

Koleksiyonlar

Egzersizler (Diziler)

4. Belirli bir tamsayının tüm varlıklarını bir tamsayı dizisinden kaldıran bir fonksiyon yazın. Fonksiyonun imzası şu şekildedir:

```
func removing(_ item: Int, from array: [Int]) -> [Int]
```

Koleksiyonlar

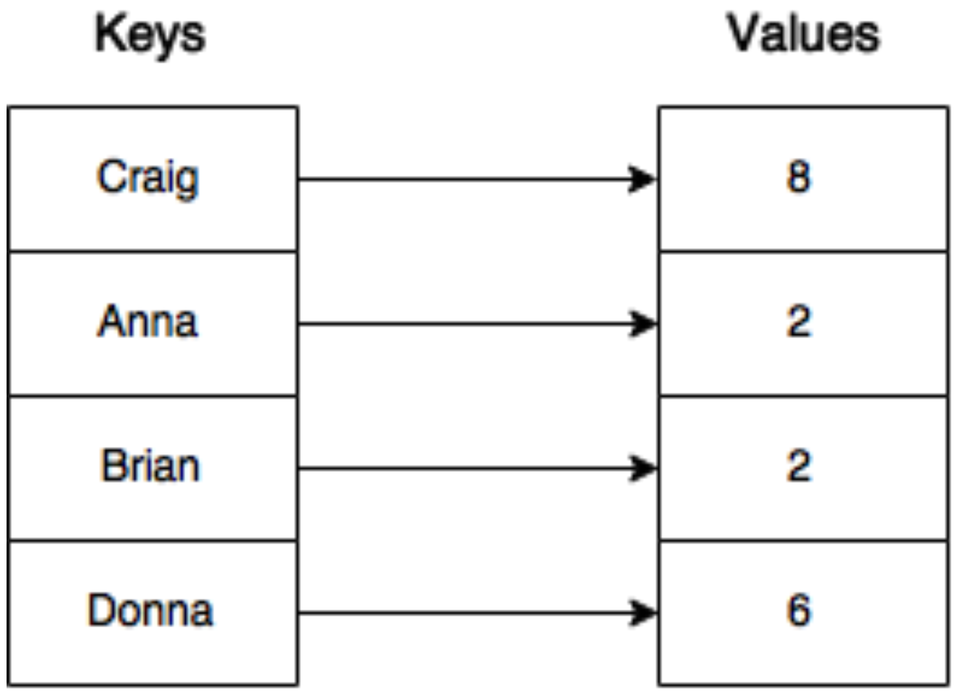
Egzersizler (Diziler)

5. Diziler, orijinal diziyle aynı öğeleri ters sırada tutan bir diziyi döndüren **`reversed()`** yöntemine sahiptir. **`reversed()`** kullanmadan aynı şeyi yapan bir fonksiyon yazın. Fonksiyonun imzası şu şekildedir:

```
func reversed(array: [Int]) -> [Int]
```

Koleksiyonlar

Sözlükler (Dictionaries)



Koleksiyonlar

Sözlük İşlemleri İçin Çalışma Süresi

- Öğelere erişim

Koleksiyonlar

Sözlük İşlemleri İçin Çalışma Süresi

- Öğelere erişim

O(1)

Koleksiyonlar

Sözlük İşlemleri İçin Çalışma Süresi

- Öğe ekleme

Koleksiyonlar

Sözlük İşlemleri İçin Çalışma Süresi

- Öğe ekleme

O(1)

Koleksiyonlar

Sözlük İşlemleri İçin Çalışma Süresi

- Öğe silme

Koleksiyonlar

Sözlük İşlemleri İçin Çalışma Süresi

- Öğe silme

$$O(1)$$

Koleksiyonlar

Sözlük İşlemleri İçin Çalışma Süresi

- Öğe arama

Koleksiyonlar

Sözlük İşlemleri İçin Çalışma Süresi

- Öğe arama

O(1)

Koleksiyonlar

Egzersizler (Sözlükler)

1. Aşağıdakilerden hangileri geçerli ifadelerdir?

```
let dict1: [Int, Int] = [:]

let dict2 = [:]

let dict3: [Int: Int] = [:]
```

Koleksiyonlar

Egzersizler (Sözlükler)

2. `let dict4 = ["One": 1, "Two": 2, "Three": 3]`

   sözlüğüne göre aşağıdakilerden hangisi geçerli ifadelerdir?

```
dict4[1]

dict4["One"]

dict4["Zero"] = 0

dict4[0] = "Zero"
```

Koleksiyonlar

Egzersizler (Sözlükler)

3. Anahtar olarak iki harfli eyalet kodları ve değerler olarak tam eyalet adları içeren bir sözlük verildiğinde, adları sekiz karakterden uzun olan tüm eyaletleri yazdıran bir fonksiyon yazın. Örneğin;

   ```
   ["NY": "New York", "CA": "California"]
   ```

   sözlüğü için çıktı `"California"` olacaktır.

Koleksiyonlar

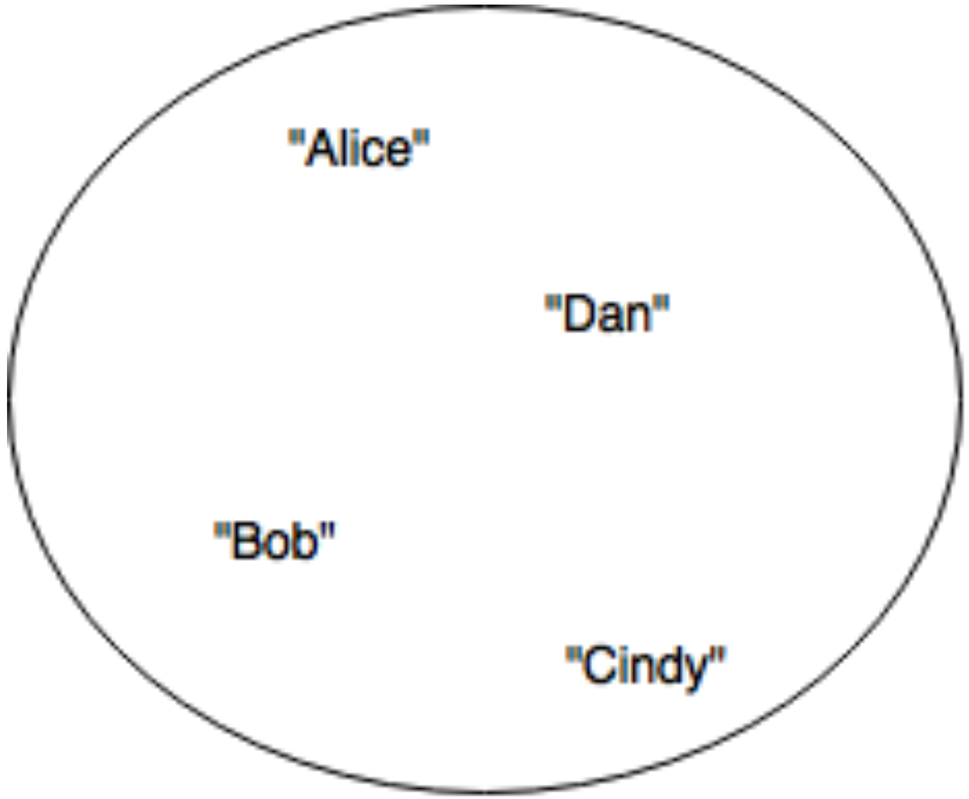
Egzersizler (Sözlükler)

4. İki sözlüğü tek bir sözlükte birleştiren bir fonksiyon yazın. Her iki sözlükte de ortak bir anahtar varsa, ilk sözlükteki çifti yok sayın. Fonksiyonun imzası şudur:

```
func merging(_ dict1: [String: String], with dict2: [String: String]) -> [String: String]
```

Koleksiyonlar

Kümeler (Sets)



Koleksiyonlar

Küme İşlemleri İçin Çalışma Süresi

- Öğe arama, silme, ekleme

Koleksiyonlar

Küme İşlemleri İçin Çalışma Süresi

- Öğe arama, silme, ekleme

$$O(1)$$

Koleksiyonlar

Egzersizler (Genel)

1. `var array5 = [1, 2, 3]`

   dizisi için aşağıdakilerden hangileri geçerli ifadelerdir?

   ```
   array5[0] = array5[1]
   ```

   ```
   array5[0...1] = [4, 5]
   ```

   ```
   array5[0] = "Six"
   ```

   ```
   array5 += 6
   ```

   ```
   for item in array5 { print(item) }
   ```

Koleksiyonlar

Egzersizler (Genel)

2. Bir dizinin orta elemanını döndüren bir fonksiyon yazın. Dizi boyutu çift olduğunda, ortadaki iki elemandan ikincisini döndürmelisiniz. Fonksiyonun imzası şu şekildedir:

```
func middle(_ array: [Int]) -> Int?
```

Koleksiyonlar

Egzersizler (Genel)

3. Bir tamsayı dizisindeki minimum ve maksimum değeri dönen bir fonksiyon yazın. Bu değerleri kendiniz bulun; **`min`** ve **`max`** yöntemlerini kullanmayın. Verilen dizi boşsa **`nil`** döndürün. Fonksiyonun imzası şu şekildedir:

```
func minMax(of numbers: [Int]) -> (min: Int, max: Int)?
```

Koleksiyonlar

Egzersizler (Genel)

4. `var dict5 = ["NY": "New York", "CA": "California"]`

   sözlüğü için aşağıdakilerden hangileri geçerli ifadelerdir?

   ```
   dict5["NY"]
   dict5["WA"] = "Washington"
   dict5["CA"] = nil
   ```

Koleksiyonlar

Egzersizler (Genel)

5. Bir metinde hangi karakterlerin geçtiğini ve bu karakterlerin her birinin ne sıklıkta ortaya çıktığını hesaplayan bir fonksiyon yazın. Sonucu sözlük olarak geri döndürün. Fonksiyonun imzası şudur:

```
func occurrencesOfCharacters(in text: String) -> [Character: Int]
```

İpucu 1: `String`, `for` ifadesiyle yineleyebileceğiniz bir karakter koleksiyonudur.

Bonus: Kodunuzu kısaltmak için sözlüklerin, sözlükte bulunmuyorsa varsayılan bir değer eklemenize izin veren özel bir **subscript** operatörü vardır. Örneğin, `dictionary["a", default: 0]`, yalnızca `nil` döndürmek yerine bulunmazsa, `"a"` karakteri ile bir `"a"` - `0` çifti oluşturur.

Koleksiyonlar

Egzersizler (Genel)

6. Bir sözlüğün tüm değerleri benzersizse `true` döndüren bir fonksiyon yazın. Benzersizliği test etmek için bir küme kullanın. Fonksiyonun imzası şudur:

```
func isInvertible(_ dictionary: [String: Int]) -> Bool
```

Koleksiyonlar

Egzersizler (Genel)

7. Aşağıdaki sözlüğe göre:

```
var nameTitleLookup: [String: String?] =

["Mary": "Engineer", "Patrick": "Intern", "Ray":
"Hacker"]
```

“Patrick" anahtarının değerini nil olarak ayarlayın ve "Ray" için anahtarı ve değeri tamamen kaldırın.

Koleksiyonlar

Özet (Diziler)

1. Aynı türden sıralı değer koleksiyonlarıdır.

2. Öğelere erişmek ve güncellemek için **subscript** veya birçok özellik ve yöntemden birini kullanabilirsiniz.

3. Sınırların dışında olan bir indekse erişim konusunda dikkatli olun.

Koleksiyonlar

Özet (Sözlükler)

1. Anahtar-değer çiftlerinin sıralanmamış koleksiyonlarıdır.
2. Anahtarların ve değerlerin tümü kendi içlerinde aynı türdendir.
3. Değerleri almak ve çift eklemek, güncellemek veya kaldırmak için **subscript** veya yöntemlerden birini kullanabilirsiniz.
4. Bir anahtar sözlükte değilse, arama `nil` değerini döndürür.
5. Bir sözlüğün anahtarı, `Hashable` protokolüne uyan bir tür olmalıdır.
6. `String`, `Int`, `Double` gibi temel Swift türleri `Hashable` protokolüne uygundur.

Koleksiyonlar

Özet (Kümeler)

1. Aynı türden benzersiz değerlerin sıralanmamış koleksiyonlarıdır.

2. Koleksiyona bir şeyin dahil edilip edilmediğini bilmeniz gerektiğinde çok kullanışlıdır.

3. Koleksiyondaki tüm öğeler `Hashable` protokolüne uygun bir türde olmalıdır.