

TEB



iOS Programlama Eğitimi

5. Gün

ARC



- Automatic Reference Counting
- Swift'te hafıza yönetimi
- ARC, artık bir nesneye ihtiyaç kalmayınca onun kapladığı hafızayı serbest bırakır. (Geri verir)

ARC Nasıl Çalışır



```
class Person {  
    let name: String  
    init(name: String) {  
        self.name = name  
        println("\(name) is being initialized")  
    }  
    deinit {  
        println("\(name) is being deinitialized")  
    }  
}
```

ARC Nasıl Çalışır



```
var reference1: Person?  
var reference2: Person?  
var reference3: Person?
```

ARC Nasıl Çalışır



```
reference1 = Person(name: "John Appleseed")  
// prints "John Appleseed is being initialized"
```

ARC Nasıl Çalışır



```
reference2 = reference1  
reference3 = reference1
```

ARC Nasıl Çalışır



```
reference1 = nil  
// prints nothing
```

```
reference2 = nil  
// prints nothing
```

```
reference3 = nil  
// prints "John Appleseed is being deinitialized"
```

Strong Reference Cycle



```
class Person {  
    let name: String  
    init(name: String) { self.name = name }  
    var apartment: Apartment?  
    deinit { println("\(name) is being deinitialized") }  
}
```

```
class Apartment {  
    let number: Int  
    init(number: Int) { self.number = number }  
    var tenant: Person?  
    deinit { println("Apartment #\(number) is being  
deinitialized") }  
}
```


Strong Reference Cycle



```
var john: Person?  
var number73: Apartment?
```

Strong Reference Cycle



```
john = Person(name: "John Appleseed")  
number73 = Apartment(number: 73)
```

Strong Reference Cycle



var john

strong

<Person instance>

name: "John Appleseed"

apartment: nil

var number73

strong

<Apartment instance>

number: 73

tenant: nil

Strong Reference Cycle



```
john!.apartment = number73  
number73!.tenant = john
```

Strong Reference Cycle



var john

strong

<Person instance>

name: "John Appleseed"

apartment: <Apartment instance>

var number73

strong

<Apartment instance>

number: 73

tenant: <Person instance>

strong

strong

Strong Reference Cycle



```
john = nil  
// prints nothing
```

```
number73 = nil  
// prints nothing
```

Strong Reference Cycle



var john

var number73

<Person instance>

name: "John Appleseed"

apartment: **<Apartment instance>**

strong

<Apartment instance>

number: 73

tenant: **<Person instance>**

strong



Strong Reference Cycle



- Soru: Nasıl çözülür?
- Cevap: Weak Reference

Strong Reference Cycle



```
class Person {  
    let name: String  
    init(name: String) { self.name = name }  
    var apartment: Apartment?  
    deinit { println("\(name) is being deinitialized") }  
}
```

```
class Apartment {  
    let number: Int  
    init(number: Int) { self.number = number }  
    var tenant: Person?  
    deinit { println("Apartment #\(number) is being  
deinitialized") }  
}
```

Weak References



```
class Person {  
    let name: String  
    init(name: String) { self.name = name }  
    var apartment: Apartment?  
    deinit { println("\(name) is being deinitialized") }  
}
```

```
class Apartment {  
    let number: Int  
    init(number: Int) { self.number = number }  
    weak var tenant: Person?  
    deinit { println("Apartment #\(number) is being  
deinitialized") }  
}
```

Weak References



```
var john: Person?  
var number73: Apartment?  
  
john = Person(name: "John Appleseed")  
number73 = Apartment(number: 73)  
  
john!.apartment = number73  
number73!.tenant = john
```

Weak References



var john

strong



<Person instance>

name: "John Appleseed"

apartment: <Apartment instance>

var number73

strong



<Apartment instance>

number: 73

tenant: <Person instance>

strong



weak



Weak References

```
john = nil  
// prints "John Appleseed is being deinitialized"
```

var john

var number73

strong

<Person instance>

name: "John Appleseed"

apartment: <Apartment instance>

strong

<Apartment instance>

number: 73

tenant: <Person instance>

weak

Weak References



```
number73 = nil  
// prints "Apartment #73 is being deinitialized"
```

var john

var number73

<Person instance>

name: "John Appleseed"

apartment: <Apartment instance>

<Apartment instance>

number: 73

tenant: <Person instance>



Weak References



- Reference Cycle kırıldı

Weak References



- Reference Cycle kırıldı
- Not: Sadece değişkenler weak olarak tanımlanabilir, sabitler weak olamaz.

Unowned References



```
class Customer {
    let name: String
    var card: CreditCard?
    init(name: String) {
        self.name = name
    }
    deinit { println("\(name) is being deinitialized") }
}

class CreditCard {
    let number: UInt64
    unowned let customer: Customer
    init(number: UInt64, customer: Customer) {
        self.number = number
        self.customer = customer
    }
    deinit { println("Card #\(number) is being deinitialized") }
}
```

Unowned References



```
var john:Customer? = Customer(name: "John Appleseed")  
  
john!.card =  
CreditCard(number:1234_5678_9012_3456,customer:john!)
```

Unowned References



var john

strong

<Customer instance>

name: "John Appleseed"

card: <CreditCard instance>

strong

<CreditCard instance>

number: 1234_5678_9012_3456

customer: <Customer instance>

unowned

Unowned References



```
john = nil
// prints "John Appleseed is being deinitialized"
// prints "Card #1234567890123456 is being deinitialized"
```

var john

<Customer instance>

name: "John Appleseed"

card: **<CreditCard instance>**

strong



unowned

<CreditCard instance>

number: 1234_5678_9012_3456

customer: **<Customer instance>**