

İÇİNDEKİLER

1 GİRİŞ	1
Bu Kitap Kimler İçin?	1
Temel Kavramlar	6
iOS İşletim Sistemi	6
iOS Kısaltması Nereden Geliyor?	7
iOS İşletim Sisteminin Katmanları	8
1. Çekirdek İşletim Sistemi Katmanı (Core OS)	8
Security Framework	8
Accelerate Framework	8
External Accessory Framework	9
LibSystem Kütüphanesi	9
2. Çekirdek Hizmetler Katmanı (Core Services)	9
3. Medya Hizmetleri Katmanı (Media Services)	13
4. Cocoa Touch Katmanı	14
iOS Yazılım Geliştirme Kiti (iOS SDK)	16
iOS Yazılım Geliştirme Kiti Araçları	17
1. Xcode Entegre Geliştirme Ortamı (Xcode IDE)	17
2. iOS Simulator	18
3. Interface Builder	19
4. Instruments	20
5. Apple LLVM Derleyicisi	21
2 GELİŞTİRME ORTAMLARININ KURULUMU	23
iOS SDK ve Geliştirme Ortamlarının İndirilme Yöntemleri	23
1. Mac OS X AppStore Uygulaması	24
Araçlığıyla Xcode'un Son Stabil Sürümünü İndirmek	24
2. Apple Firmasının Geliştiriciler İçin Kullanıma Sunduğu developer.apple.com Sitesine Kaydolmak	25
3. iOS Developer Program'a Yıllık 99\$ Karşılığında Üye Olmak	26

X OBJECTIVE-C

iOS SDK ve Geliştirme Ortamlarının Kurulumu	28
İlk Objective-C Programımız	30

3 OBJECTIVE-C PROGRAMLAMA DİLİNE GİRİŞ 35

Objective-C Dilinin Tarihçesi	35
Objective-C Dilinin Özellikleri	36
Objective-C = (ANSI C + SmallTalk)	36
Objective-C Programlama Ortamının Bileşenleri	38
Nesne Yönelimli Programlama	38
Temel Nesne Yönelimli Programlama Kavramları	40
Encapsulation (Sarmalama)	40
Inheritance (Kalıtım ve Miras)	42
Member Variables (Üye Değişken)	44
Polymorphism (Çok Biçimlilik)	46
Objective-C Diliyle C Tabanlı Diğer Programlama Dillerinin Karşılaştırılması	48
C ve Objective-C	48
C++ ve Objective-C	48
Java ve Objective-C	50
Özet	51
Bundan Sonrası	52

4 C PROGRAMLAMA DİLİNE ÖZET BİR BAKIŞ 53

Bu Bölüme Neden İhtiyaç Duyuldu?	53
Bölümde Kullanılacak Olan Görseller Hakkında	54
C Programlama Dilinin Tarihçesi	54
C Programlama Dilinin Standartları	56
1) Kernighan & Ritchie Standardı - K&R C	57
2) ANSI C Standardı (C89)	57
3) ISO C Standardı (C90)	57
4) ISO 99 C Standardı (C99)	58
C programlama Dili Neden Bu Kadar Yaygın?	58

Bir C Dosyasının Yapısı	60
Xcode ve C Programlama Dili	62
Bir C Programının Adım Adım İncelenerek Yürütülmesi	64
1. Adım (10. satır)	66
2. Adım (12. satır)	67
3. Adım (13. satır)	68
4. Adım (14. satır)	68
5. Adım (15. satır)	70
6. Adım (3. satır)	70
7. Adım (5. satır)	71
8. Adım (6. satır)	71
9. Adım (7. satır)	71
10. Adım (15. satır)	72
11. Adım (16. satır)	72
12. Adım (17. satır)	73
Giriş/Çıkış (I/O) Fonksiyonları ve #include <stdio.h>	73
Derleme ve Bağlama İşlemleri	74
Derleme İşlemi (Compile)	75
Bağlama İşlemi (Link)	75
Derleyici (Compiler)	75
Temel C Programlama Kavramları	76
Değişkenler ve Değerleri	76
Değişken Nedir?	76
Değişkenlerin Tanımlanması	76
Değişkenlerin İsimlendirilmesi	77
Veri Tipleri	78
Temel Veri Tipleri	78
Temel Veri Tiplerinin Türevleri	82
Özel Değişken Tipleri	85
Sabit Değişkenler ve const Anahtar Kelimesi	85

XII OBJECTIVE-C

Statik Değişkenler ve static Anahtar Kelimesi	86
volatile Anahtar Kelimesi	88
extern Anahtar Kelimesi	89
register Anahtar Kelimesi	91
Temel Girdi Çıktı Komutları (printf ve scanf)	92
printf Fonksiyonu	93
Biçim Niteleyicilerinin printf Fonksiyonu İçindeki Kullanımı	97
Özel Karakterler (Escape Sequences)	98

4 C PROGRAMLAMA DİLİ - İLERİ KULLANIMLAR 189

Bu Bölüme Neden İhtiyaç Duyuldu?	189
Tip Dönüşümleri (Type Casting)	190
Dönüşüm (cast) Operatörü	190
main.c (Bölüm5 Örnek1)	190
Tip Çevrimleri (Type Conversion)	193
main.c (Bölüm5 Örnek2)	194
#define Ön İşlemci Komutu	196
Fonksiyonlar	197
main.c (Bölüm5 Örnek3)	199
Fonksiyon Prototipi (İmzası)	200
main.c (Bölüm5 Örnek4)	201
main.c (Bölüm5 Örnek5)	201
Hiçbir Dönüş Tipi Olmayan (void) Fonksiyonlar	202
main.c (Bölüm5 Örnek6)	203
İç İçer Fonksiyonlar	203
main.c (Bölüm5 Örnek7)	205
Main Fonksiyonu ve Program Parametreleri	207
main.c (Bölüm5 Örnek8)	208
Fonksiyonlara Dizi Göndermek	209
main.c (Bölüm5 Örnek9)	209
Değişken Kapsamları (Scope)	211

Yerel Değişkenler	211
Global Değişkenler	212
main.c (Bölüm5 Örnek10)	212
İşaretçi (Pointer) Kavramı	214
İşaretçi Deklarasyonu	214
Adres Operatörü (&)	215
Erişim Operatörü (*)	215
main.c (Bölüm5 Örnek11)	216
Neden Tek Bir Tip İşaretçi Yok? (İşaretçi Aritmetiği)	218
main.c (Bölüm5 Örnek12)	219
İşaretçiler Aracılığıyla Dizi Yönetimi	221
main.c (Bölüm5 Örnek13)	221
Bir işaretçiyi Fonksiyona Parametre Olarak Göndermek	224
Değer Yöntemiyle Çağrı (Call By Value)	224
main.c (Bölüm5 Örnek14)	224
Referans Yöntemiyle Çağrı (Call By Reference)	225
main.c (Bölüm5 Örnek15)	225
Swap İşlemi	226
main.c (Bölüm5 Örnek16)	226
Ortalama Bulan Programın (Bölüm5 Örnek9) İşaretçi Notasyonu	
Kullanan Versiyonu	227
main.c (Bölüm5 Örnek17)	227
NULL Değeri ve İşaretçiler	228
main.c (Bölüm5 Örnek18)	229
Genel Amaçlı (void) İşaretçiler	229
main.c (Bölüm5 Örnek19)	230
Fonksiyon İşaretçileri	231
main.c (Bölüm5 Örnek20)	232
main.c (Bölüm5 Örnek21)	233
Dinamik Bellek Yönetimi	235

XIV OBJECTIVE-C

malloc() fonksiyonu	235
free() fonksiyonu	235
sizeof() fonksiyonu	235
main.c (Bölüm5 Örnek22)	236
İşaretçi Değişkenlerin Birbirine Atanması Durumu	238
main.c (Bölüm5 Örnek23)	238
Farklı Bellek Alanlarının	241
Farklı İşaretçiler Aracılığıyla Yönetimi	241
main.c (Bölüm5 Örnek24)	241
Dizilerde Dinamik Hafıza Yönetimi	244
Tek Boyutlu Dizilerde Dinamik Hafıza Yönetimi	244
main.c (Bölüm5 Örnek25)	245
İki Boyutlu Dizilerde Dinamik Hafıza Yönetimi	247
main.c (Bölüm5 Örnek26)	249
Diğer Hafıza Yönetimi Fonksiyonları (calloc ve realloc)	253
main.c (Bölüm5 Örnek27)	254
Veri Yapıları (Structs)	256
main.c (Bölüm5 Örnek28)	257
Veri Yapılarının İşaretçiler Aracılığıyla Kullanımı (Struct Pointers)	259
main.c (Bölüm5 Örnek29)	261
typedef Anahtar Kelimesi	262
main.c (Bölüm5 Örnek30)	262
Bundan Sonrası	264
6 OBJECTIVE-C PROGRAMLAMA DİLİ	265
Ne Durumdayız?	265
Objective-C ile C Programlama Diline Yapılan Eklentiler	266
Yeni Ön İşlemci (Preprocessor) Direktifleri	266
Sade Eklentiler	267
SmallTalk Söz Dizimi	268
Mesaj Kavramı	269

Runtime	270
Karakter Katarlarının Ele Alınış Biçimi ve NSString Sınıfı	270
NSLog	270
Bölüm6 Örnek1: main.m	271
Yeni Anahtar Kelimeler	273
#import	273
id	274
BOOL Veri Tipi	275
nil	276
sel	276
Özel Nümerik Veri Tipleri	277
NSInteger	278
NSUInteger	278
CGFloat	278
Objective-C Anahtar Kelimeleri	279
Sınıf, Nesne, Kategori ve Protocol Tanımlarken Kullanılan Direktifler	279
Encapsulation (Sarmalama) İçin Kullanılan Anahtar Kelimeler	279
Hata Ayıklama (Exception Handling) İçin Kullanılan Anahtar Kelimeler	281
Protokol (Protocol) Tanımına Özel Anahtar Kelimeler	282
Kategori (Category) Tanımına Özel Anahtar Kelimeler	282
Dille İlgili Diğer Anahtar Kelimeler	282
Dikkat Etmeniz Gereken Diğer Anahtar Kelimeler	283
Metot ve Fonksiyon Kavramları Üzerine	284
Üye Değişkeni Kavramı Üzerine	284
Objective-C Uygulamalarının Genel Yapısı	284
Ogrenci.h	288
Ogrenci.m	288
Ogrenci.h	289
Ogrenci.m	290
main.m	291



XVI OBJECTIVE-C

Bundan Sonrası 295

7 SINIFLAR, NESNELER VE KALITIM KAVRAMI 297

Ne Durumdayız? 297

Giriş 297

Nesne ve Sınıf Kavramları 298

Nesne (Object) Kavramı 298

Sınıf (Class) Kavramı 300

Sınıf ve Nesne Arasındaki İlişki 301

Nesne Yönelimli Programanın Temel Özellikleri 303

Kendi Nesnelerimizi Tasarlamak 305

Arayüz (Interface) Dosyasının Yapısı 309

@interface Anahtar Kelimesi 309

NObject Sınıfı ve Kök (Root) Sınıflar 309

Üye Değişkenler 310

Metotlar 312

Sınıf Metotları ve Nesne (Instance) Metotları 313

Objective-C Sınıflarında Bulunan Metotların Genel Sözdizimi 314

1) Hiç Parametre Almayan Metotlar 315

2) Tek Parametre Alan Metotlar 315

3) Birden Fazla Parametre Alan Metotlar 317

Uygulama (Implementation) Dosyasının Yapısı 318

self Anahtar Kelimesi 324

alloc Metodu 326

Değer Atayıcı Metotlar (Initializers) 327

Kendi Değer Atayıcı Metotlarımızı 328

Tasarlamak (Designated Initializers) 328

dealloc Metodu 330

İsmlendirme Kuralları 331

Kalıtım Kavramı 332

Ogrenci Sınıfının Arayüz Dosyası (Ogrenci.h) 335



Oğrenci Sınıfının Uygulama (Implementation) Dosyası (Oğrenci.m)	336
Personel Sınıfının Arayüz Dosyası (Personel.h)	340
Personel Sınıfının Uygulama (Implementation) Dosyası (Personel.m)	341
Uygulama Dosyası (main.m)	343
Metot Geçersiz Kılma (Override) İşlemi	347
Metotların Aşırı Yüklenmesi	348
Bundan Sonrası	349
8 OBJECTIVE-C 'DE HAFIZA YÖNETİMİ	351
Ne Durumdayız?	351
Giriş	352
Primitif Değişkenler ile Nesne Değişkenlerin Bellekteki Görünümü	359
Hafıza Yönetimi Yöntemleri	362
Hafızanın Elle Yönetilmesi (Manual Memory Management) ve Referans Sayma (Reference Counting) Yöntemi	363
Referans Sayma Yöntemi	364
Başboş Referans Kavramı (Dangling Pointer)	367
Ownership (Sahiplik) Kavramı	367
Elverişli Metotlar (Convenience Methods)	368
Bir Metottan Nesne Döndürmek	369
autorelease Mesajı	371
Nesneleri Oluşturduğumuz Anda autorelease Mesajı Göndermek	372
İleride Otomatik Olarak Silinecek Nesnelere Havuzu (Autorelease Pool)	372
Yöntem 1: NSAutoreleasePool Sınıfını Kullanmak	372
Drain Mesajı	374
Yöntem 2: @autoreleasepool Direktifini Kullanmak	375
İç İç Havuzlar (Nested Autorelease Pools)	376
Havuz Tanımlamadan autorelease Çağırarak	376
Retain Mesajı ve RetainCount Kavramı	376
Retain Mesajı	377
RetainCount	377

XVIII OBJECTIVE-C

Nesneleri Kopyalamak	379
Copy Metodunun Kullanımı	381
MutableCopy Metodu	385
Elle Hafıza Yönetiminde Dikkat Edilmesi Gereken Noktalar	386
Dengesiz Retain Release Çağruları	386
Retain Döngüsü	386
Garbage Collection (Çöp Toplayıcı)	387
Memory Leak (Hafıza Sızıntısı) Kavramı	389
Automatic Reference Counting (Otomatik Referans Sayma)	389
Bundan Sonrası	394

9 SARMALAMA (ENCAPSULATION) KAVRAMI VE PROPERTY KULLANIMI 395

Ne Durumdayız?	395
Giriş	395
Sarmalama (Encapsulation)	395
Prensip 1: Tasarladığınız Sınıfın Durum Bilgisine Erişimi Mümkün Olduğunca Kısıtlamaya Çalışınız.	396
Prensip 2: Eğer başkasının tasarladığı bir sınıfı kullanıyorsanız ve bu sınıf size hem kendi durum bilgisine erişim imkanı hem de metotlar aracılığıyla durum bilgisine erişim imkanı sunuyor,sa, mümkün olduğu kadar metot kullanımını tercih ediniz.	396
Üye Değişkenlere Erişimin Kısıtlanması - Erişim Direktifleri	400
@public Değişkenlere İşaretçi (Pointer) Yöntemiyle Erişilmesi	404
Üye Metotlara Erişimin Kısıtlanması	406
Değer Okuyucu (Accessor) ve Değer Atayıcı (Mutattor) Metotlar	406
Değer Okuyucu (Accessor) ve Değer Atayıcı (Mutattor) Metotların İsmlendirilmesi	407
Değer Okuyucu (Getter) Metotların Genel İçeriği	409
Değer Atayıcı (Setter) Metotların Genel İçeriği	409
Direkt Değer Atama (Assign)	409
Sahiplik Yoluyla Değer Atama (Retain)	411

Kopyalama Yoluyla Değer Atama (Copy)	411
Defansif Kopyalama	411
Hangi Yöntem Ne Zaman Kullanılmalı, Nelere Dikkat Edilmeli	412
Property (Özellik) Kavramı	416
Property ile Üye Değişken İsimlerinin Farklı Olması	418
@property Direktifinin Yapısı	420
@dynamic Anahtar Kelimesi	421
Dot Syntax (Nokta Notasyonu) Kullanımı	425
Bundan Sonrası	429

10 ÇOK BİÇİMLİLİK (POLYMORPHISM), DİNAMİK BAĞLAMA (DYNAMIC BINDING), HATA YÖNETİMİ VE PROTOKOL KAVRAMLARI 431

Ne Durumdayız?	431
Statik Tipleme (Static Typing)	431
Dinamik Tipleme (Static Typing)	432
Id Tipi	432
main.m (Bölüm10 Örnek1)	432
Dinamik Bağlama (Dynamic Binding)	433
main.m (Bölüm10 Örnek2)	434
Çok Biçimlilik (Polymorphism)	435
Statik Tipleme ve Dinamik Tipleme Arasındaki Fark	435
main.m (Bölüm10 Örnek2) - Yeni versiyon	437
Dinamik Bağlama Kullanılan Uygulamalarda Çalışma Zamanı Hatalarını Engellemek	439
Yöntem 1: NSObject Sınıfının Sağladığı Metotları Kullanmak	439
Class Nesnesi (Class Object)	440
Class Tipi	441
main.m (Bölüm10 Örnek3)	442
main.m (Bölüm10 Örnek4)	443
Selector (Seçici) Kavramı	446
Bölüm10 Örnek5 nesneBilgisiYazdır metodu (main.m)	448

XX OBJECTIVE-C

Yöntem 2: İstisna Yönetimini Kullanmak	449
İstisna (Exception) Kavramı	449
main.m (Bölüm10 Örnek6)	449
@try @catch @finally Bloğu	450
Yöntem 3: Protokol (Protocol) Kavramını Kullanmak	451
Protokol Kavramı	451
Yaklaşım 1: Aynı Mesaja Cevap Vermesini İstedığınız Sınıfların O Mesajı İçeren Bir Protokolü Uygulamasını Sağlamak	454
Yaklaşım 2: Genel Amaçlı Metot Parametrelerinin	455
Bundan Sonrası	458
11 KATEGORİLER VE BLOKLAR	459
Giriş	459
Kategori (Category) Kavramı	459
main.m (Bölüm11 Örnek1)	461
Bloklar (Blocks)	462
main.m (Bölüm11 Örnek2)	464
typedef Anahtar Kelimesinin Bloklarla Beraber Kullanımı	465
main.m (Bölüm11 Örnek3)	465
Blok İçerisinde Yerel Değişkenlere Erişim	466
main.m (Bölüm11 Örnek4)	467
Blok Tarafından Değiştirilebilir Yerel Değişkenler Tanımlamak ve __block İfadesinin Kullanımı	468
main.m (Bölüm11 Örnek5)	468
Blokların Fonksiyonlara Parametre Olarak Gönderilmesi	470
main.m (Bölüm11 Örnek6)	471
12 KOLLEKSİYONLAR (COLLECTIONS)	475
Giriş	475
C Dizilerinde Nesne Kullanımı	475
main.m (Bölüm12 Örnek1)	476

Objective-C'ye Özel Veri Sınıfları	478
NSArray	478
main.m (Bölüm12 Örnek2)	479
Xcode İçinden Dokümantasyona Hızlı Ulaşım	480
Koleksiyonları Oluştururken Elverişli Metotların Kullanılması	482
main.m (Bölüm12 Örnek3)	483
Koleksiyonların Elemanlarına Döngü Yardımıyla Erişim	484
Dizinin Eleman Sayısını Bulma	484
Normal For Döngüsüyle Dizi Elemanlarına Erişim	484
Fast Enumeration Yöntemiyle Dizi Elemanlarına Erişim	484
main.m (Bölüm12 Örnek4)	485
Mutable / Immutable (Değiştirilebilir / Değiştirilemez Kavramları)	486
NSMutableArray	486
Diziye Farklı Veri Tipinde Bir Eleman Ekleme	487
main.m (Bölüm12 Örnek5)	488
Nesnelerin Yazdırılması	489
NSDictionary ve NSMutable Dictionary	491
Eleman Ekleme	493
Eleman Çıkarma	493
Eleman Değerini Güncelleme	494
Döngü Aracılığıyla NSDictionary	494
Nesnesinin Elemanlarına Erişim Sağlanması	494
main.m (Bölüm12 Örnek6)	495
NSSet	496
NSSet Sınıfının Sağladığı Yardımcı Metotların Kullanımı	497
main.m (Bölüm12 Örnek7)	498
NSMutableSet	501
Eleman Ekleme	501
Eleman Çıkarma	501
Yardımcı Metotların Kullanımı	501

XXII OBJECTIVE-C

main.m (Bölüm12 Örnek8)	502
NSCountedSet	504
main.m (Bölüm12 Örnek9)	505
NSData ve NSMutableData	506
main.m (Bölüm12 Örnek10)	507
Bundan Sonrası	508
13 DİĞER FOUNDATION SINIFLARI VE DOSYA İŞLEMLERİ	509
Giriş	509
NSString ve NSMutableString Kullanımı	509
main.m (Bölüm13 Örnek1)	513
NSNumber Sınıfı ve Kutulama Kavramı	516
main.m (Bölüm13 Örnek2)	517
NSDate ve NSCalendarDate Sınıflarının Kullanımı	519
main.m (Bölüm13 Örnek3)	520
Objective-C'de Dosyaya Okuma ve Yazma	522
main.m (Bölüm13 Örnek4)	524
Bundan Sonrası	526