

Разработване на приложения за iOS



Петко Пенчев, TapBits.com

Foundation Framework

- **NSObject**
- основен клас за почти всеки обект в iOS SDK
- основополагащ за memory management

Foundation Framework

- **NSString**

```
// замества добре познатият ни от C  
char *str;
```

```
NSString *str = @"това е стринг"; // unicode, immutable
```

```
NSString *str = [NSString stringWithFormat:@"Тегло %d кг", weight];
```

- **NSMutableString**

```
NSMutableString *mutString = [[NSMutableString alloc] initWithString:@"0."];  
[mutString appendString:digit];
```

Foundation Framework

- **NSNumber**

object wrapper: int, float, double, BOOL

```
BOOL flag = YES;  
NSNumber *n = [NSNumber numberWithBool:flag];
```

...

```
NSNumber *num = [NSNumber numberWithFloat:36.5];
```

```
float f = [num floatValue];  
int i = [num intValue];
```

...

```
int pushBadgeCount;  
[NSNumber numberWithInt:pushBadgeCount];
```

Foundation Framework

- **NSNumber**

object wrapper: int, float, double, BOOL

```
BOOL flag = YES;  
NSNumber *n=[NSNumber numberWithInt:flag];
```

...

```
NSNumber *num = [NSNumber numberWithInt:12];
```

```
float f = [num floatValue];  
int i = [num intValue];
```

много удобен за преобразуване на primitive types в NSObject

Foundation Framework

- **NSValue**

object wrapper за необектни типове

```
CGRect r = CGRectMake(10.0, 10.0, 20.0, 20.0);  
NSValue *val = [NSValue valueWithCGRect:r];
```

- **NSData**

съхраняване и извличане на данни от всякакъв вид и тип

Foundation Network

- **NSDate**

Дата и час ...NSCalendar, NSDateFormatter, NSDateComponents

```
NSLog(@"current date: %@", [[NSDate date] description]);
```

```
...
```

```
NSDateFormatter* formatter = [[[NSDateFormatter alloc] init] autorelease];  
[formatter setDateFormat:@"MM_yyyy"];
```

```
NSDate *date = [[NSDate alloc] initWithTimeIntervalSinceNow:0];  
// NSDate *date = [NSDate date];
```

```
ProductId = [NSString stringWithFormat:@"prefix%@", [formatter stringFromDate: date]];  
NSLog(@"ProductId: %@", ProductId);
```

```
[date release];
```

```
...
```

КОЛЕКЦИИ

- **NSArray**

Индексно/подредена колекция от обекти .. [0,1,2 ... N]
Immutable

+ (id) arrayWithObjects:(id) firstObject, ...

- (int) count;
- (id) objectAtIndex:(int) index;
- (void) makeObjectsPerformSelector:(SEL) aSelector;
- (NSArray *) sortedArrayUsingSelector:(SEL) aSelector;
- (id) lastObject;

- **NSMutableArray**

- (id) initWithCapacity:(NSUInteger) numItems;
- (void) addObject:(id) anObject;
- (void) insertObject:(id) anObject atIndex:(int) index;
- (void) removeObjectAtIndex:(int) index;

```
NSMutableArray *arr=[[NSMutableArray alloc] initWithCapacity:0];
```

```
[arr insertObject:AnyNSObject atIndex:0];
```

```
...
```

```
[arr release];
```

Речници

- **NSDictionary**

Асоциативен масив, речник или hash table .. key => value
Immutable

Ако реализираме собствен key class трябва да имплементираме

- (NSUInteger)hash
- (BOOL)isEqual:(NSObject *)obj

Най-често за ключ използваме: NSString *Key

- (int)count;
- (id)objectForKey:(id)key;
- (NSArray *)allKeys;
- (NSArray *)allValues;

Пример:

```
id value = [mydict objectForKey:@"name"];
```

Речници

- **NSMutableDictionary**

Създаване:

- + (id)dictionaryWithCapacity:(NSUInteger)numItems
- (id)initWithCapacity:(NSUInteger)numItems

Добавяне на елементи:

- (void)setObject:(id)anObject forKey:(id)key;
- (void)addEntriesFromDictionary:(NSDictionary *)otherDictionary;
- setValue:forKey:
- setDictionary:

Изтриване на елементи:

- (void)removeObjectForKey:(id)key;
- removeAllObjects
- removeObjectForKey:

Множества

- **NSSet**

Неподредена колекция от обекти

- (int)count;
- (BOOL)containsObject:(id)anObject;
- (id)anyObject;
- (void)makeObjectsPerformSelector:(SEL)aSelector;
- (id)member:(id)anObject;

- **NSMutableSet**

Създаване:

- + setWithCapacity:
- initWithCapacity:

Работа с множества:

- unionSet:
- minusSet:
- intersectSet:
- setSet:

Добавяне и премахване на елементи:

- addObject:
- filterUsingPredicate:
- removeObject:
- removeAllObjects
- addObjectFromFromArray:

Обхождане

Пример:

```
// Дефинираме NSArray е масив от NSString обекти
NSArray *myArray = [NSArray arrayWithObjects:@"Едно", @"Две", @"Три", nil];
for (NSString *string in myArray) {
    double value = [string doubleValue];
}
```

Пример:

```
NSSet *mySet = ...; // може да е NSArray също
for (id obj in mySet) {
    ...
}
```

Пример:

```
NSDictionary *myDictionary = ...; // масив от id
for (id key in myDictionary) {
    id value = [myDictionary objectForKey:key];
}
```

СПИСЪЦИ СЪС “СВОЙСТВА”

- Property List

Колекция от колекции

NSArray, NSDictionary, NSNumber, NSString, NSData, ..

- Удобство при съхранение и зареждане на данни

// plist е NSArray или NSDictionary

```
[plist writeToFile:(NSString *)path atomically:(BOOL)];
```

// ще създаде и запише plist в path като .plist файл

Логване и дебъгване

```
NSLog(@"log: %@ ", myString);
```

```
NSLog(@"log: %f ", myFloat);
```

```
NSLog(@"log: %i ", myInt);
```

```
...
```

```
currentScoreLabel.text = [NSString stringWithFormat:@"%d", currentScore];
```

```
...
```

```
NSString *str = [[NSNumber numberWithInt:myInt] stringValue]
```

```
...
```

Повече детайли от

<http://developer.apple.com/library/mac/#documentation/Cocoa/Conceptual/Strings/Articles/formatSpecifiers.html>

NSZombieEnabled - много добра имплементация на система за дебъгване

<http://cocoa-nut.de/?p=16>

Логване и дебъгване

- MLog class

```
#import "MLog.h"
...
MLogString(@"hp at start: %d", hp_at_start);
...

2008-11-09 09:32:00.432 MyApp[21179:20b]
EventsItemViewController.m:163 viewDidAppear:
```

Декларация и методи

```
#import <UIKit/UIKit.h>

#ifdef REMOTELOG 1
#ifdef EXTENSIVELOG 1

#define MLogString(s,...) \
[MLog logFile:__FILE__ lineNumber:__LINE__ \
format:(s),##__VA_ARGS__]

@interface MLog : NSObject {
    BOOL __MLogOn;
}

+(void)logFile:(char*)sourceFile lineNumber:(int)lineNumber
        format:(NSString*)format, ...;
+(void)setLogOn:(BOOL)logOn;
@end
```

Форматирание NSNumberFormatter

```
NSString *tempStr = @"8,765.4";
NSNumberFormatter * myNumFormatter = [[NSNumberFormatter alloc] init];
[myNumFormatter setLocale:[NSLocale currentLocale]];
[myNumFormatter setFormatterBehavior:NSNumberFormatterBehavior10_4];
[myNumFormatter setNumberStyle:NSNumberFormatterDecimalStyle];
NSNumber *tempNum = [myNumFormatter numberFromString:tempStr];
NSLog(@"string '%@' gives NSNumber '%@' with intValue '%"i",
      tempStr, tempNum, [tempNum intValue]);
[myNumFormatter release];
...

NSDate* date= [NSDate dateWithTimeIntervalSinceNow:0];
NSDateFormatter* formatter = [[[NSDateFormatter alloc] init] autorelease];
NSTimeZone *gmtTimeZone = [NSTimeZone timeZoneWithName:@"Europe/London"];
                        //[NSTimeZone timeZoneForSecondsFromGMT:60*60];
[formatter setTimeZone:gmtTimeZone];

NSString* datastr = @"...";
[formatter setDateFormat:@"yyyy-MM-dd HH:mm"];
NSDate* dateb = [formatter dateFromString: datastr];
...
```

Memory Management

Коректно:

```
- (NSDictionary *)testVariableValues {
    NSMutableDictionary *returnValue = [[NSMutableDictionary alloc] init];
    [returnValue setObject:[NSNumber numberWithInt:3.5] forKey:@"x"];
    [returnValue setObject:[NSNumber numberWithInt:23.8] forKey:@"y"];
    return [returnValue autorelease];
}
```

Добре:

```
- (NSDictionary *)testVariableValues {
    NSMutableDictionary *returnValue = [NSMutableDictionary dictionary];
    [returnValue setObject:[NSNumber numberWithInt:3.5] forKey:@"x"];
    [returnValue setObject:[NSNumber numberWithInt:23.8] forKey:@"y"];
    return returnValue;
}
```

По-добре:

```
- (NSDictionary *)testVariableValues {
    return [NSDictionary dictionaryWithObjectsAndKeys:[NSNumber numberWithInt:3.5], @"x",
        [NSNumber numberWithInt:23.8], @"y", nil];
}
```

Memory Management

Λοшо:

```
- (NSString *)tenThousandGs {  
    NSString *s = "";  
    for (int i = 0; i < 10000; i++) s = [s stringByAppendingString:@"O"];  
    return s;  
}
```

Добро:

```
- (NSString *)tenThousandGs {  
    NSMutableString *s = [[NSMutableString alloc] init];  
    for (int i = 0; i < 10000; i++) [s appendString:@"O"];  
    return [s autorelease];  
}
```

По-добре:

```
- (NSString *)tenThousandGs {  
    NSMutableString *s = [NSMutableString string];  
    for (int i = 0; i < 10000; i++) [s appendString:@"O"];  
    return s;  
}
```