iPhone SDK Tutorial

Chapter 2 최소한의 개념

Junil Ko (goya@vinia.net)

1. 역사

- **Objective C**는 **C**의 슬림-슈퍼셋
  스몰 톡(Small Talk)에 뿌리를 두고 있음.
- **C** 소스와 혼용가능
  C소스와 혼용가능 하며 헤더 파일은 “*.h” 구현파일은 “*.m” 사용.
- **C++** 소스와 혼용가능
  C++ 소스와 혼용가능 하며 헤더 파일은 “*.h” 구현파일은 “*.mm” 사용.
- **Objective C 2.0**
  Property Access, Garbage Collection, Enumeration의 속도향상
- **Objective C 2.1**
  Block 개념과 Thread 관리의 향상

GOYA

2. OOP

GOYA

2-1. Header File

```
//
//  GYVoiceRecorder.h
//  VoiceRecorder
//
//  Created by Junil Ko on 11/25/09.
//  Copyright 2009 __MyCompanyName__. All rights reserved.
//
#import <Foundation/Foundation.h>
#import <UIKit/UIKit.h>
#import <AudioToolbox/AudioQueue.h>
#import <AudioToolbox/ExtendedAudioFile.h>
#import <AudioToolbox/AudioServices.h>

@interface GYVoiceRecorder : NSObject {
    BOOL bRecording;

    AudioStreamBasicDescription audFormat;
    AudioQueueRef recQueueObject;  // 녹음 큐
    /* AudioQueueRef playQueueObject;  // 재생 큐 */

}
@property (readwrite) AudioFileID audioFileID;

- (void)sessionInit;
- (void)writeMagicCookie;
- (UInt32)getQueueMiliSeconds:(AudioQueueRef) queue;

@end
```

GOYA

2-1. Implementation File

```
//
//  GYVoiceRecorder.m
//  VoiceRecorder
//
//  Created by Junil Ko on 11/25/09.
//  Copyright 2009 __MyCompanyName__. All rights reserved.
//

#import "GYVoiceRecorder.h"

@implementation GYVoiceRecorder

@synthesize packetDescriptions;
@synthesize m_recMiliSeconds;

- (id)init {
    if ( [super init] )    {

#ifdef TARGET_OS_IPHONE
        [self sessionInit];
#endif

    }
    return self;
}
```

GOYA

2-1. Class & Instance

```
GYSwitch *mySwitch = [[GYSwitch alloc] init];
```

```
- (id)init {
    if ( [super init] )    {
        NSArray *filePaths = NSSearchPathForDirectoriesInDomains
            (NSDocumentDirectory, NSUserDomainMask,YES);

        CFStringRef fileString = (CFStringRef) [NSString stringWithFormat: @"%@/Recording.caf",
                                                [filePaths objectAtIndex: 0]];
        [filePaths release];

        fileURL =    (NSURL*)CFURLCreateWithFileSystemPath
            (NULL,fileString,kCFURLPOSIXPathStyle,false);

#ifdef TARGET_OS_IPHONE
        [self sessionInit];
#endif

    }
    return self;
}
```

GOYA

2-1. Class & Instance

C

```
DrawMyGradient(myView, aquaGradient, 0.0, 1.0, YES);
```

C++

```
myView->drawGradient(aquaGradient, 0.0, 1.0, YES);
```

Objective C

```
[myView drawMyGradient:aquaGradient fromValue:0.0 toValue:1.0 opaque:YES];
```

```
- (void)drawMyGradient:(CGGradientRef)gradient
                fromValue:(float)start
                    toValue:(float)end
                    opaque:(BOOL)opaque {

}
```

2-1. Class & Instance

C

```
SubStringFromTo(sourceStr, &destStr, 2, 5);
```

C++

```
destStr = sourceStr->subStringFromTo(2, 5);
```

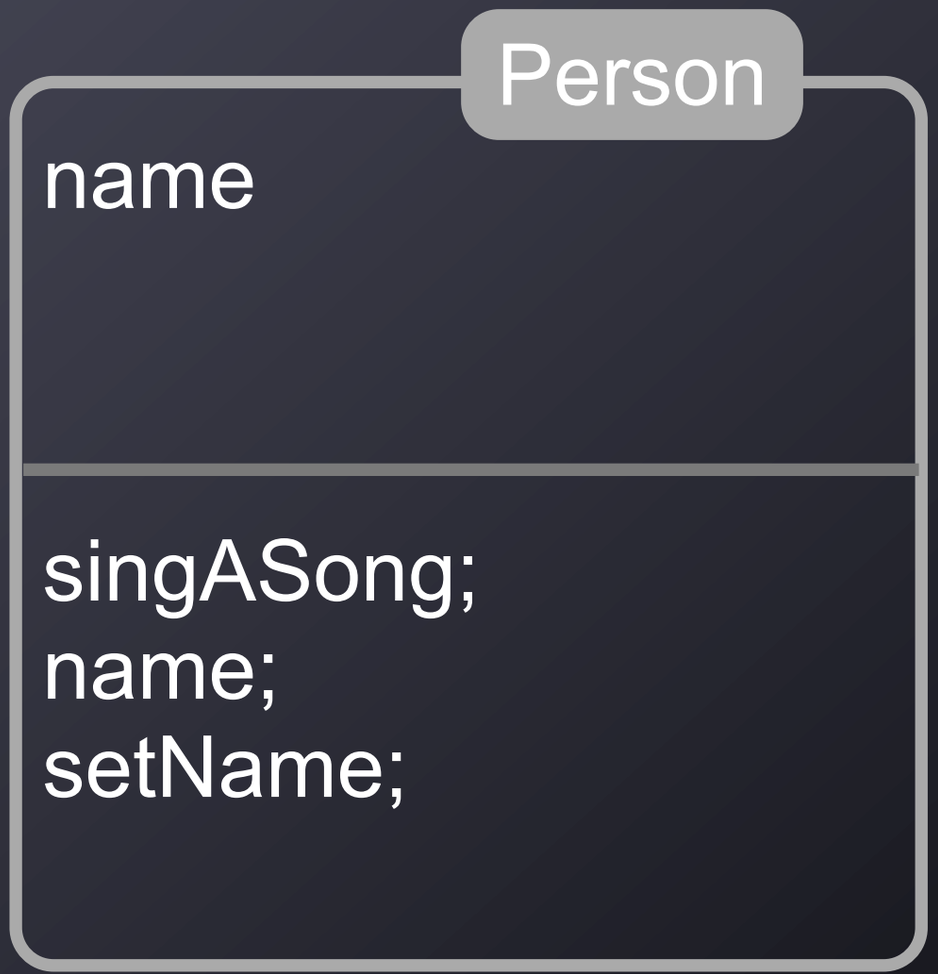
Objective C

```
destStr = [sourceStr subStringFrom:2 to 5];
```

```
- (NSString *)subStringFrom:(int)from to:(int)to
```

```
hashValue = [[sourceStr subStringFrom:2 to 5] hash];
```

3. Cocoa Class 만들기



p.29 내용

예제 따라하기

GOYA

3. Cocoa Class 만들기

p.30 내용
예제 따라

%@	Cocoa Object
%d	Integer
%f	Float
%s	Char

__FUNCTION__	실행 중인 Method Name
__LINE__	실행 중인 줄 번호
__FILE__	실행 중인 파일 이름

GOYA

4. Header 파일 포함

*.h

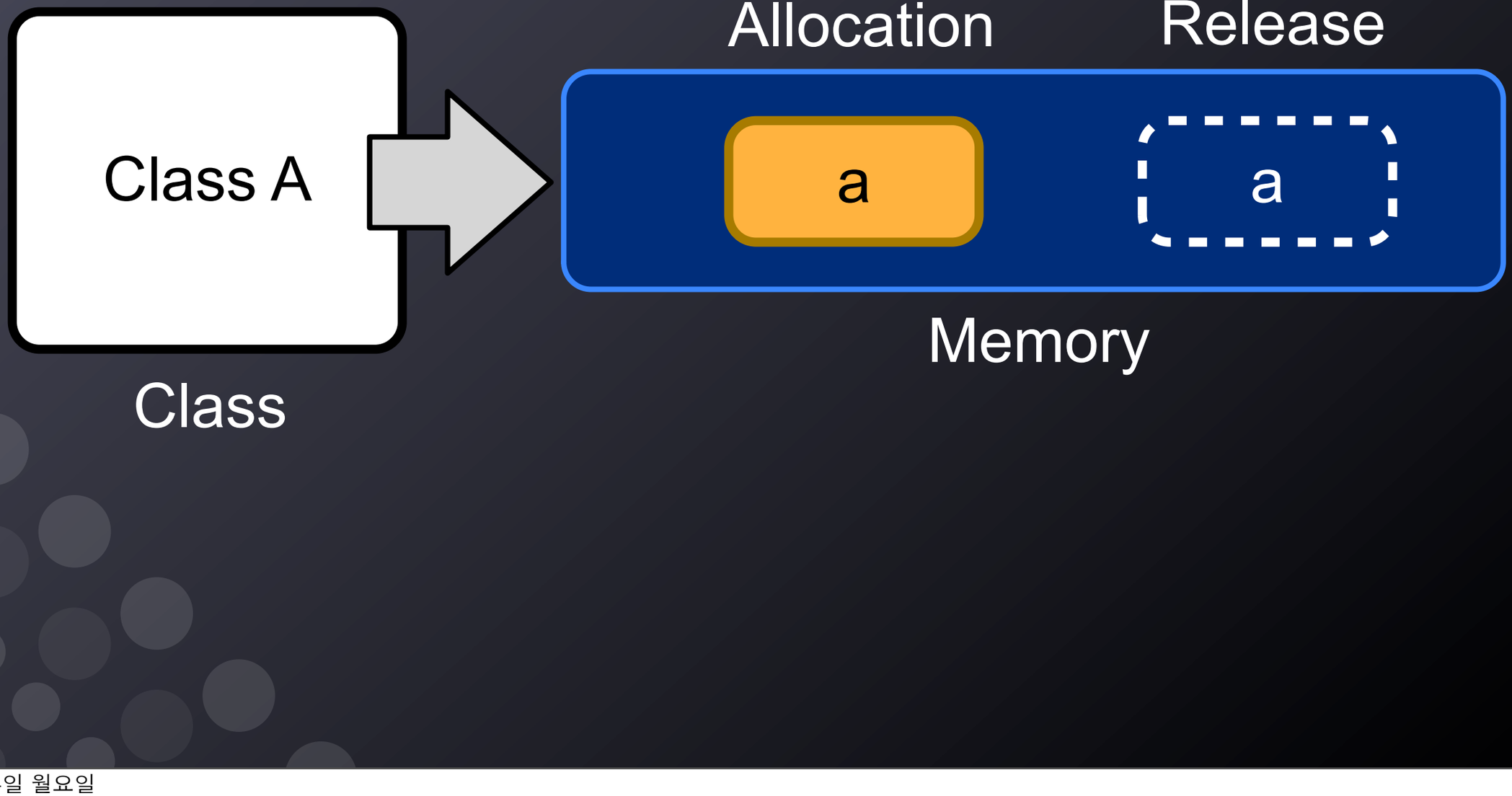
```
@class Pet;
@interface Person : NSObject <NSCoding>
{
...
}
...
@end
```

*.m

```
#import “Pet.h”
@implement Person
...
@end
```

GOYA

5. Memory 관리



5-1. Retain Count

```
NSString *retainedString = [[NSString alloc] initWithString:@”LiveCoder”];
// retainedString’s retain count = 1

[retainedString retain];
// retainedString’s retain count = 2

[retainedString release];
// retainedString’s retain count = 1

[retainedString release];
// retainedString’s retain count = 0

[retainedString release];
// error
```

GOYA

5.2 Class Method

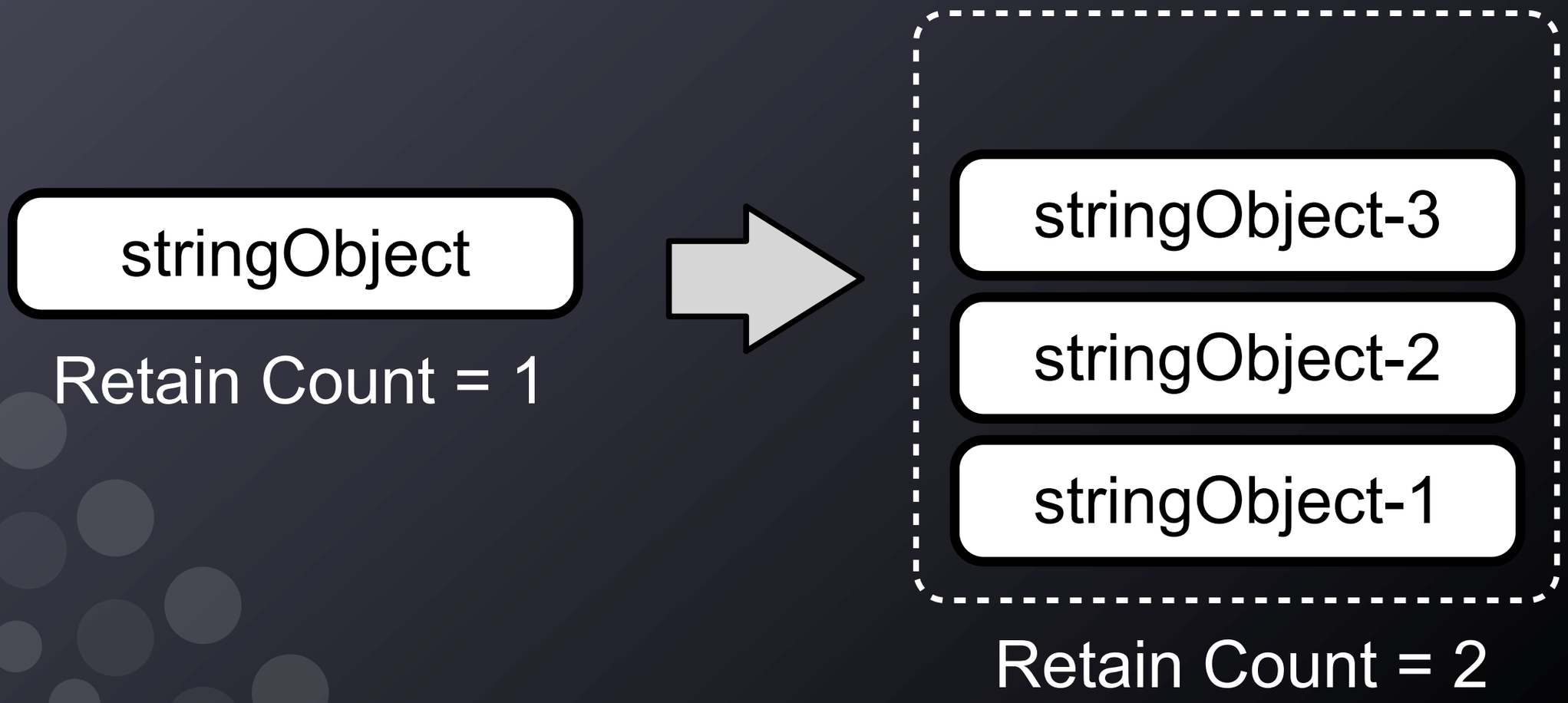
- **Class Method**
  + 가 붙어있는 **Method.**
  별도의 메모리 관리가 필요치 않음,
  해당 메소드 내에서만 수명이 보장 됨.

- **Instance Method**
  - 가 붙어 있는 **Method.**
  메모리 관리 필요.

GOYA

5.3 일반적인 Release 형태

```
NSString *stringObject = [[NSString alloc] initWithString:@”Yoon”];
[mutableArray addObject:stringObject];
[stringObject release];
```



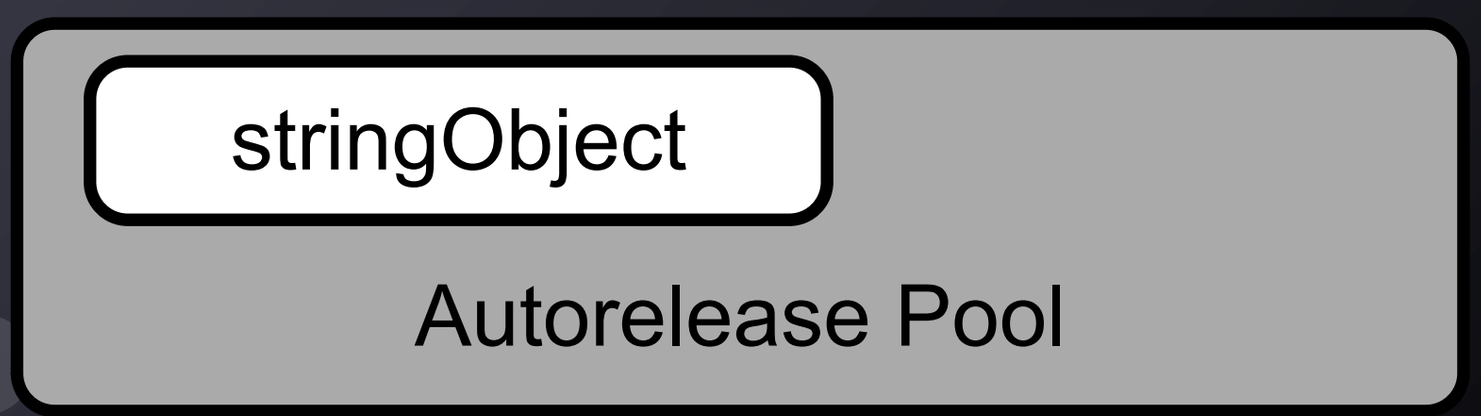
GOYA

5.4 Autorelease

```
NSString *stringObject;

// alloc, init, autorelease
stringObject = [[[NSString alloc] initWithString:@”Yoon”] autorelease];

// init with autorelease
stringObject = [NSString stringWithString:@”Yoon”];
```



GOYA

5.4 Autorelease

```
- (NSString *)fullName
{
    NSString *composedString = [[NSString alloc] initWithFormat:
                                @"%@ %@", firstName, lastName];
    [composedString release];

    return composedString;
}
```

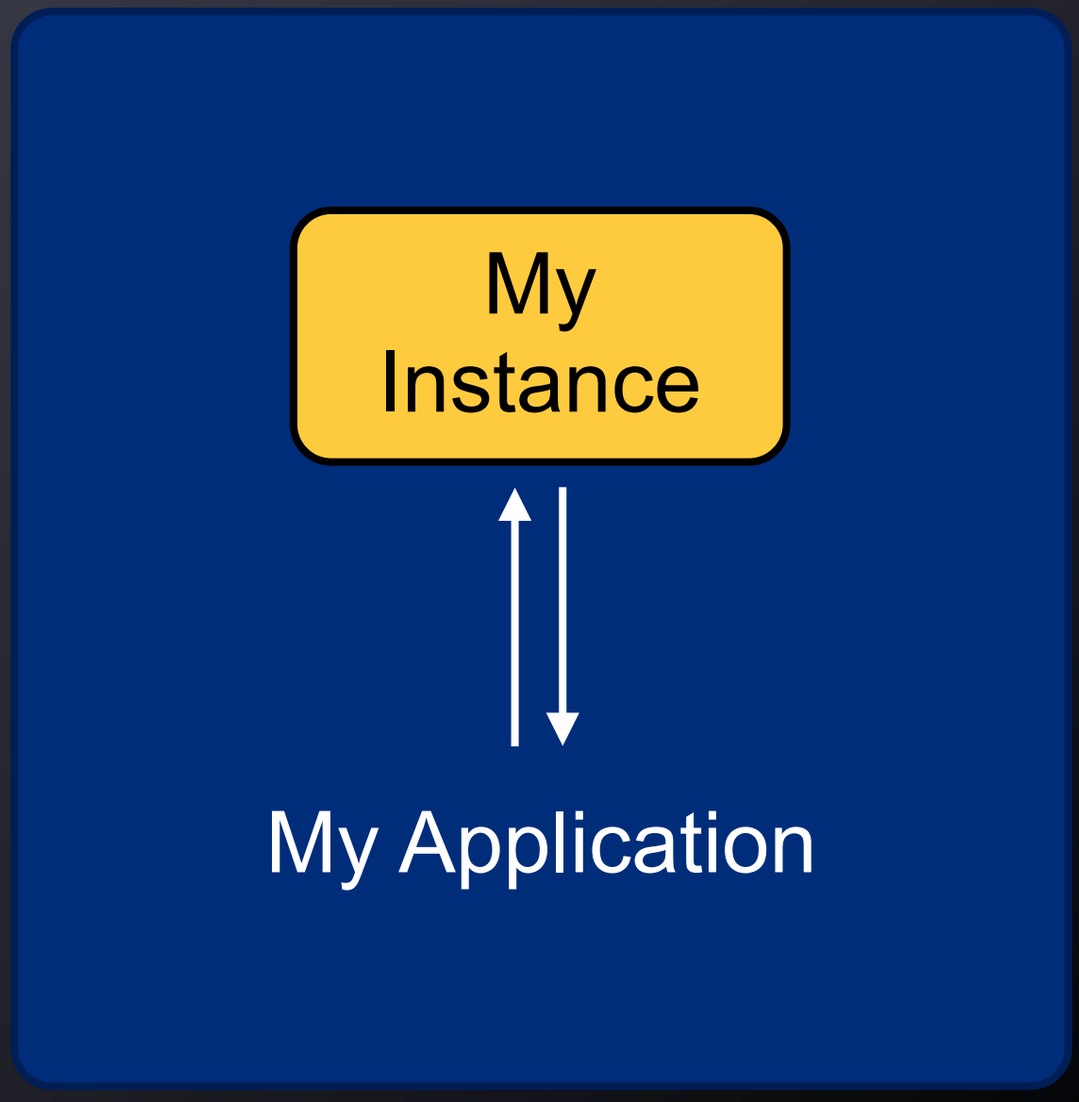
```
- (NSString *)fullName
{
    NSString *composedString = [[NSString alloc] initWithFormat:
                                @"%@ %@", firstName, lastName];
    [composedString autorelease];

    return composedString;
}
```

GOYA

6. Delegate

기존 Class에 새로운 기능을 부여하고 싶을때



Class Instance의 동작을 즉각 알아채고 싶을때

GOYA

6-1. Delegate 구현

```
- (void)createB {
    B *b = [[B alloc] init];
    b.delegate = self;
}

- (void)changedColor {
    ...
}
```

```
@protocol
- (void)changedColor
@end


[self.delegate
changedColor];
```

GOYA