

#09. Category

전미정

Category

- 이미 존재하는 클래스에 메소드를 추가 할 수 있는 방법으로, 추가된 메소드는 이후 subclass에 상속되며, runtime 구동시 클래스에 원래 작성되어 있는 메소드와 구별되지 않는다.
- 카테고리를 이용해 특정 기능을 사용하는 메소드를 추가할 수 있고, 추후 메소드를 기능에 맞게 선별적으로 사용할 수 있다. 이는 상속으로 인한 코드의 거대화를 방지해주며, 유지보수를 편리하게 해준다.
- 카테고리는 기존 클래스에 메소드를 추가할 수 있지만, 인스턴스 변수를 추가할 수는 없다. 추가로 선언되는 변수 및 accessor 메소드 생성을 컴파일러가 실행하지 않기 때문에 이미 선언되어있는 변수를 사용해야 한다. 인스턴스 변수를 추가생성하기 위해서는 Class Extension를 사용해야한다.
- 기존 클래스에 이미 선언되어있는 메소드를 재선언하게 되면 카테고리의 메소드가 우선되어 기존의 메소드는 호출되지 않으니 주의해야한다.
- 중복되는 이름의 메소드를 생성하지 않도록 유의해야한다.

문법

```
#import "XYZPerson.h"

@interface XYZPerson (XYZPersonNameDisplayAdditions)
- (NSString *)lastNameFirstNameString;
@end
```

.h file

```
#import "XYZPerson+XYZPersonNameDisplayAdditions.h"

@implementation XYZPerson (XYZPersonNameDisplayAdditions)
- (NSString *)lastNameFirstNameString {
    return [NSString stringWithFormat:@"%@", self.lastName, self.firstName];
}
@end
```

.m file

```
#import "XYZPerson+XYZPersonNameDisplayAdditions.h"
@implementation SomeObject
- (void)someMethod {
    XYZPerson *person = [[XYZPerson alloc] initWithFirstName:@"John"
                                                             lastName:@"Doe"];

    XYZShoutingPerson *shoutingPerson =
        [[XYZShoutingPerson alloc] initWithFirstName:@"Monica"
                                                    lastName:@"Robinson"];

    NSLog(@"The two people are %@ and %@",
          [person lastNameFirstNameString], [shoutingPerson lastNameFirstNameString]);
}
@end
```

.m file of class file

- 카테고리 파일의 이름은 카테고리가 포함되는 클래스 이름 + 카테고리 이름으로 만들어진다.
- Private method를 만들기 위해서 .m파일의 @interface에서 선언, @implementation 부분에서 메소드를 정의할 수 있다.

Extension

- 카테고리과 함께, 이미 존재하는 class에 기능을 확장해 커스텀하게 변경할 수 있는 방식으로, 카테고리와 다르게 메소드와 인스턴스 변수 모두 추가할 수

있다.

- 카테고리과 다르게 별도의 파일을 생성하지 않기때문에 framework class는 확장 할 수 없다.
- 소스파일의 @interface에 ()를 이용해 선언하기 때문에 무명의 카테고리라고도 불린다.

<https://goo.gl/6w97CO> (apple reference)