

클래스와 객체

헤더파일추가

- #import 지시어로 사용
- Framework의 경우는 <Framework>로 사용된다.
- 다른 클래스의 경우는 “classfile.h로 헤더파일을 추가 가능하다.(.m파일은 참조 불가)
- 헤더 파일이 import되어야 다른 클래스의 객체 생성 및 사용이 가능하다.

@class 사용

- @class name;
- 해당 클래스의 존재를 알려준다.
- 실제 사용하기 위해서는 @implementation에 #import 시켜야한다.

Objective-C class특징

- interface / 헤더(정의)
- implementation (구현)

선언부 - interface

- 클래스를 선언한다. (상속 관계와 프로토콜 선언)
- 객체와 통신할 메서드를 선언한다.
- property 변수를 선언한다.
- .h 파일 안에 작성된다.

구현부 - implementation

- 클래스의 메서드를 구현하는 실제 코드가 담겨 있다.
- .m 파일에 작성된다.

메소드

- 컴퓨터가 수행할 명령어
- 인스턴스(객체)에 적용되거나 , 클래스에 적용되는 행동으로 나눌수 있다.
- 객체는 보통 객체 자신만 접근 가능한 정보를 포함하는데, 메서드는 이 데이터에 접근하고 수정하는 방법을 제공한다.

메소드 타입(-,+)

- (-) : 인스턴스(객체) 메서드
- (+) : 클래스 메서드
- 메서드 내용은 중괄호{}로 함수의 영역을 표시한다.
- ‘return’ 키워드는 반환 값을 나타낸다.

메서드 특징 - 매개변수(parameter)

- 매개변수가 있을때는 쌍따옴표로 (:) 입력 부분을 구분
- 매개변수는 타입과 변수명을 나눠 쓴다.
- 타입은 괄호() 안에 작성된다.
- 매개변수는 해당 함수 안에서만 사용 가능하다.

메서드 특징 - 반환값(return value)

- 반환값이 있는 경우 괄호() 안에 반환 타입을 작성 해준다.
- 결과값 반환은 return 키워드를 사용하며 반환타입과 같은 타입만 반환가능
- return키워드 사용시 그 이후에 작성된 코드는 실행되지 않으며 메서드를 종료한다.
- 한개의 값만 반환 할수있다.
- 반환값이 없는 경우는 반환 타입에는 void를 작성한다, return키워드 또한 사용할 필요가 없다.

산술연산자

- “+” 더하기, “-“ 빼기, “ * “ 곱하기, “ / “ 나누기, “%” 나머지(integer일때만)

메세징

- 메서드를 불러오고 싶다면 object를 메세징해야 합니다.(함수의 호출)
- 메세지는 대괄호[]에 의해 표현됩니다.
- 왼쪽에는 메세지를 받을 오브젝트, 오른쪽은 보낼 메세지 입니다.

초기화 메서드

- 인스턴스 생성시 객체의 초기화값을 정해줄 수 있다.
- 기본 초기화 메서드(init)은 NSObject에 정의 되어 있다.
- init 메서드를 override 할수있다.
- 사용자 임의대로 custom init 메서드를 만들 수 있다.
- 초기화 함수 return값으로instancetype을 반환 해주어야 한다.

클래스 메서드 vs 인스턴스 메서드

- 클래스 메서드
 - 인스턴스 속성에 상관없이 클래스 자체에 필요한 행위를 정의 할때 사용
 - 메서드 타입에 (+) 기호 사용

클래스 메서드 선언

```
//클래스 메서드
//숫자 한개를 받아서 그 수가 짝수인지 확인하는 메소드
+ (BOOL)isEvenNumber:(NSInteger)number;

//성적 평균을 입력시 해당 평균의 성적 등급을 매긴다.
+ (NSInteger)matchingGrade:(CGFloat)averige;
```

클래스 메서드 사용

```
//짝수일경우는 예스, 홀수일경우는 노
NSLog(@"%d", [Jjacsu isEvenNumber:12]);
//100~90은 1등급, 80~90은 2등급
NSLog(@"%ld", [Jjacsu matchingGrade:89]);
```

인스턴스 메서드

- 인스턴스에서 사용되는 행위로 대부분 메서드가 해당된다.
- 메서드 타입에 (-) 기호 사용

프로퍼티 작성

```
//triangle 구하기
- (CGFloat)triangleArea:(NSInteger)bottom high:(NSInteger)high; //삼각형의 넓이
```

인스턴스 메서드 선언

```
//삼각형 넓이 구하기
- (CGFloat)triangleArea:(NSInteger)bottom high:(NSInteger)high
{
    return (CGFloat)0.5 * bottom * high;
}
```

클래스 메서드 사용

```
//삼각형
NSLog(@"삼각형의 넓이는 %lf이다", [dimOne triangleArea:2 high:2]);
```

조건문 개론

- 함수 내부에서 실행되는 선택문
- 특정 조건에 따라 선택적으로 코드를 실행시킨다.
- 대표적인 조건문으로 if-else문과 switch-case문이 있다.

if-else문

- 조건이 참일경우 if 문 대괄호 안의 코드가 실행된다.
- 만약 조건이 거짓인 경우 else문 대괄호 안의 코드가 실행된다.
- 조건값은 참, 거짓의 나타나는 bool값으로 표현

```
//어떤값을 넣었을때 짝수이면 1, 짝수가 아니면 0
+ (BOOL)isEvenNumber:(NSInteger)number
{
    //처음엔 0이니깐 노
    BOOL isEven = NO;
    //2의 배수
    if (number%2 == 0) {
        //짝수
        isEven = YES;
    }else
    {
        //홀수
        isEven = NO;
    }

    return isEven;
}
```

else if문

- 추가 조건 방법으로 반복해서 추가 할수있다.
- 조건2는 조건1이 거짓일때 실행된다.

```
//100보다 작고 90보다 크면 1, 90보다 작고 80보다 크면 2
+ (NSInteger)matchingGrade:(CGFloat)averige
{
    NSInteger gradeNum = 0;
    if (averige <= 100 && averige >= 90)
    {
        gradeNum = 1;
    }else if(averige <= 90 && averige >= 80)
    {
        gradeNum = 2;
    }
    return gradeNum;
}
```

비교연산자

- “==“ A와B가 같다
- “>=“ A가 B보다 크거나 같다
- “<=“ A가 B보다 작거나 같다
- “ > “ A가 B보다 크다
- “ < “ A가 B작다, “ ! “ 같지않다

논리연산자

- “&&“ A조건이 참이고, B조건이 참이면 참이다, 둘다 참이면 참이다, AND
- “||“ A조건이나, B조건 둘중에 하나가 참이면 참이다, 둘중 하나라도 참이면 참이다, OR
- “! “ 조건의 반대(참 -> 거짓, 거짓 -> 참), NOT

SWITCH문

- 상태 값에 따라 실행될 코드를 선택하는 조건문

```
//1이면 전액장학금, 2면 오십프로, 3이면 이십오프로 나머지는 없음
+ (void)scholarlarchipWithGrade:(NSInteger)grade
{
    //메서드 안에서 로그를 출력하기 때문에 메인에서 로그찍을필요없다
    switch (grade) {
        case 1:
            NSLog(@"전액장학금");
            break;

        case 2:
            NSLog(@"50%");
            break;

        case 3:
            NSLog(@"25%");
            break;
        //마지막엔 디폴트
        default:
            NSLog(@"없음");
            break;
    }
}
```