

03

객체 지향 프로그래밍 소개

고준일

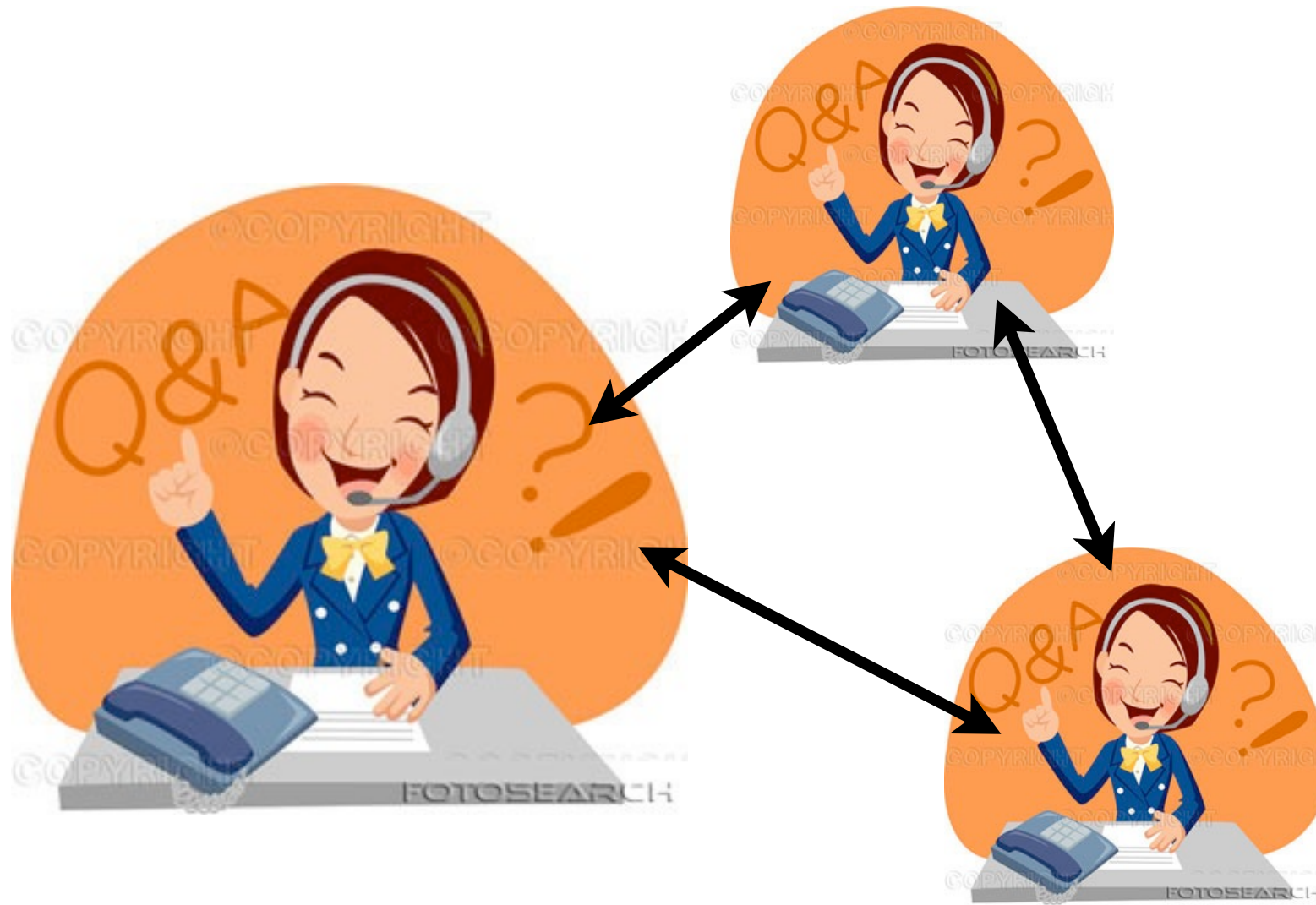
goya@vinia.net

Indirection



- 피자를 먹기 위해서는 피자집 전화번호를 외워야 하는가?

Indirection



- 전화 상담원에게 전화를 걸면 받은 상담원이 모든 문제에 대한 상담을 할 수 있는가?

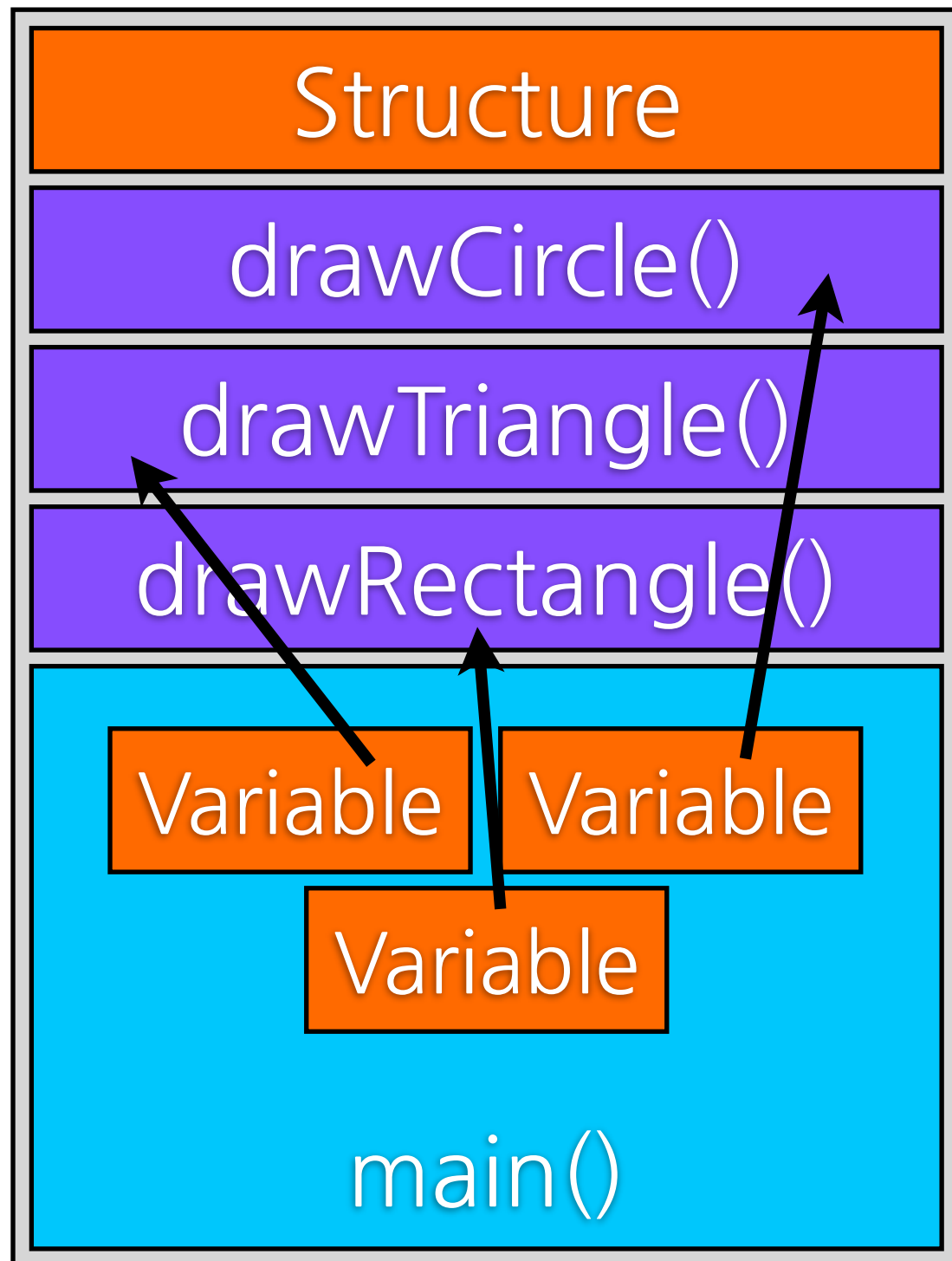
Indirection

p46 - p52 예제 DEMO

절차지향적 언어의 단점

1. 삼각형 하나를 추가하기 위해 여러 부분의 변경이 필요.
2. 전체 소스를 파악하기 이전에는 기능별 구분이 애매한점.

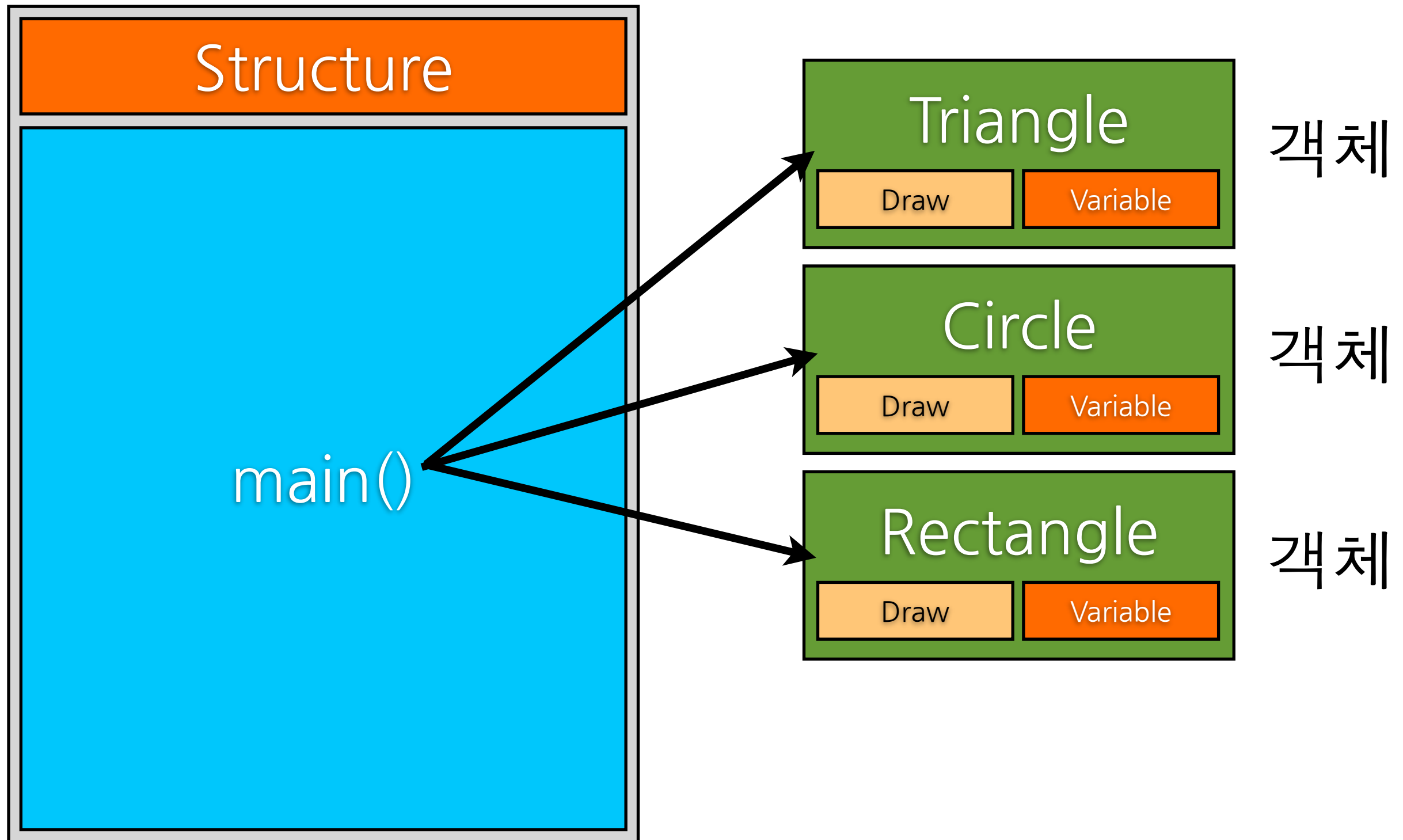
절차지향의 형태



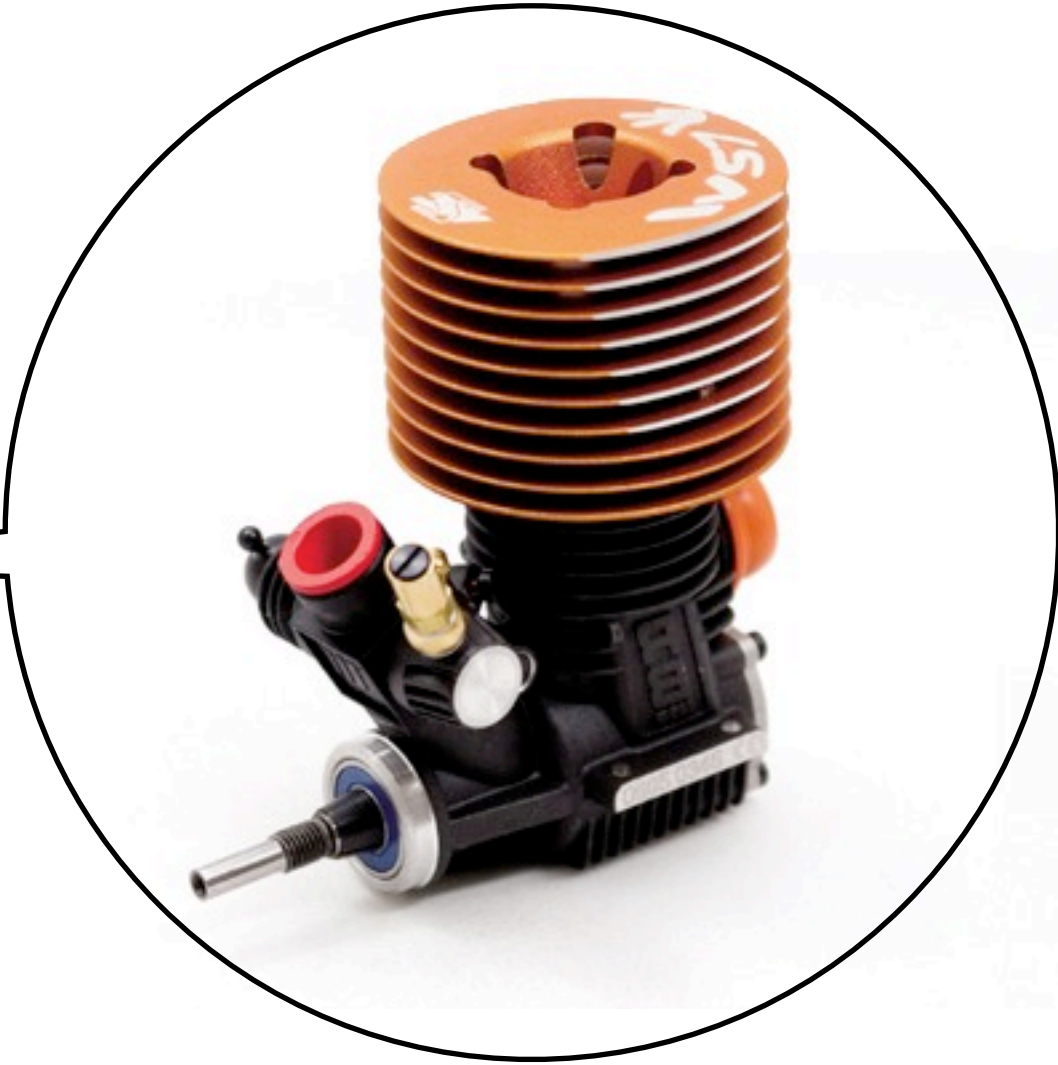
- 기능 변경시 많은 부분을 고려해야 한다.
- 기능 확장이 용이하지 못하다.
- 유지보수에 어려움이 많다.

예) 함수나 변수가 변경되면 코드 전반적인 수정이 불가피 하다.

객체지향은 무엇인가?



객체란 무엇인가?



- 객체지향 언어는 자동차의 부품들과 같다.

객체지향 언어는 자동차의 부품들 처럼 각각의 객체들이 고유의 기능을 가지고 하나의 자동차를 구성하는 형태와 같다.

객체란 무엇인가?



- 객체지향 언어에서 객체는 카세트와 같다.

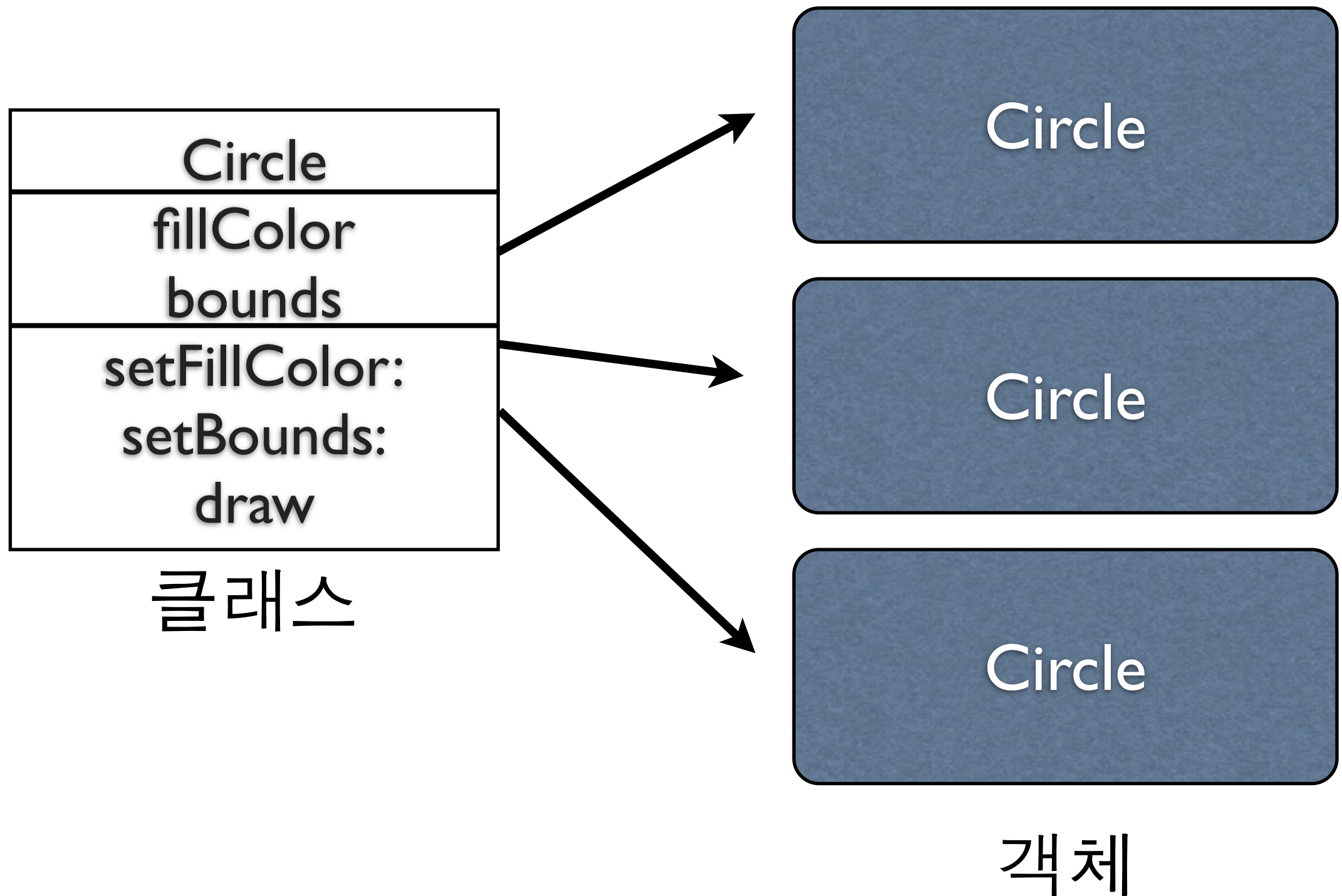
우리는 카세트가 내부에서 어떻게 동작하는지 전혀 관심이 없다.
하지만 어떻게 하면 음악이 재생이 되고, 또 어떻게 하면 정지 되는지는 알고 있다.
세상에는 엄청난 종류의 카세트가 존재하지만 대부분 메뉴얼을 읽어야 할 필요성을 느끼는 사람들은 많지 않다.

용어정리

Circle
fillColor bounds
setFillColor: setBounds: draw

- 클래스
객체의 타입을 나타내는 구조체.
객체를 만들어내기 위한 설계도와 같다.
값을 저장할 공간(멤버변수)과 어떤 일을 위한 동작(메소드)가 정의되어 있다.
- 객체(인스턴스)
클래스를 통해 만들어진 실재 부품.
클래스에서 정의된 값을 저장할 공간이 실재 메모리에 할당되고,
클래스에서 정의된 동작의 포인터를 갖는다.
- 메시지
객체에 정의된 동작을 실행시키는 것을 말한다.
(메소드를 호출한다)

클래스와 객체



이제 객체지향으로 코드를 변경해 봅시다.

그 전에 Objective-C의 문법을 몇가지
간단히 살펴보자면...

Objective-C 문법

Interface
(*h)

Implementation
(*m)

Objective-C 문법

@interface

@end

@implementation

@end

Objective-C 문법

```
@interface Circle:NSObject
```

```
@end
```

```
@implementation
```

```
@end
```

Objective-C 문법

```
@interface Circle:NSObject  
{  
    ShapeColor fillColor;  
    ShapeRect bounds;  
}
```

@end

```
@implementation
```

@end

Objective-C 문법

```
@interface Circle:NSObject
```

```
{
```

```
    ShapeColor fillColor;
```

```
    ShapeRect bounds;
```

```
}
```

```
- (void)setFillColor:(ShapeColor)fillColor;
```

```
- (void)setBounds:(ShapeRect)bounds;
```

```
- (void)draw;
```

```
@end
```

```
@implementation
```

```
@end
```

Objective-C 문법

@interface Circle:NSObject

{

ShapeColor

ShapeRect

}

- (void)setFillColor:(ShapeColor)fillColor;

- (void)setBounds:(CGRect)bounds;

- (void)draw;

@end

@implementation

인수가 없는 메소드

- (void)draw;

인수가 하나인 메소드

- (void)setFillColor:(ShapeColor)fillColor;

인수가 여러개인 메소드 (중위 표기법)

- (void)setFillColor:(ShapeColor)fillColor
withRect:(CGRect)rect;

@end

Objective-C 문법

```
@interface Circle:NSObject
{
    ShapeColor fillColor;
    ShapeRect bounds;
}
- (void)setFillColor:(ShapeColor)fillColor;
- (void)setBounds:(ShapeRect)bounds;
- (void)draw;
```

@end

@implementation

```
- (void)setFillColor:(ShapeColor)fillColor {
    ...
}

- (void)setBounds:(ShapeRect)bounds {
    ...
}

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

@end