



# G-CAMP WORKSHOP

BUILD THE RIGHT THING  
BUILD THING RIGHT  
SHIP FAST

*Lecturer* Kim Myoung-Ho  
*Nickname* 블스  
[blogstudy@naver.com](mailto:blogstudy@naver.com)

## 18강\_델리게이트와 이벤트

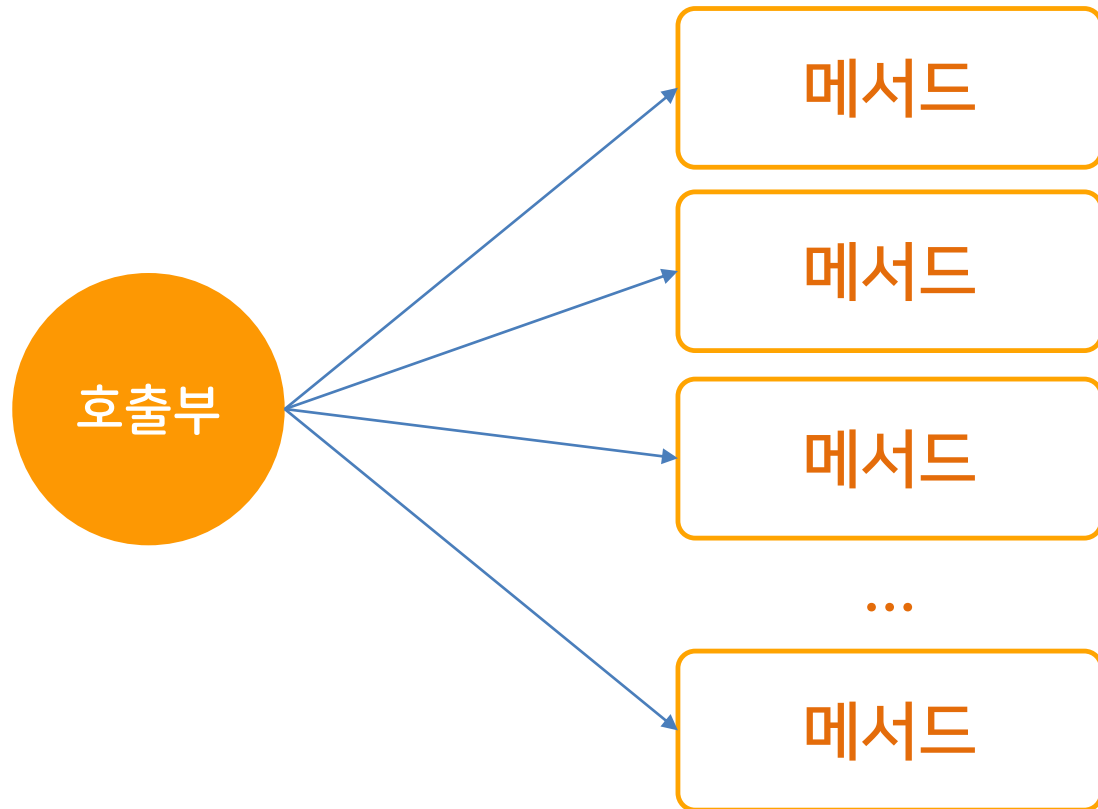
- Delegate란?
- Delegate 사용하기
- Delegate Chain
  - Event란?
  - Event 사용하기



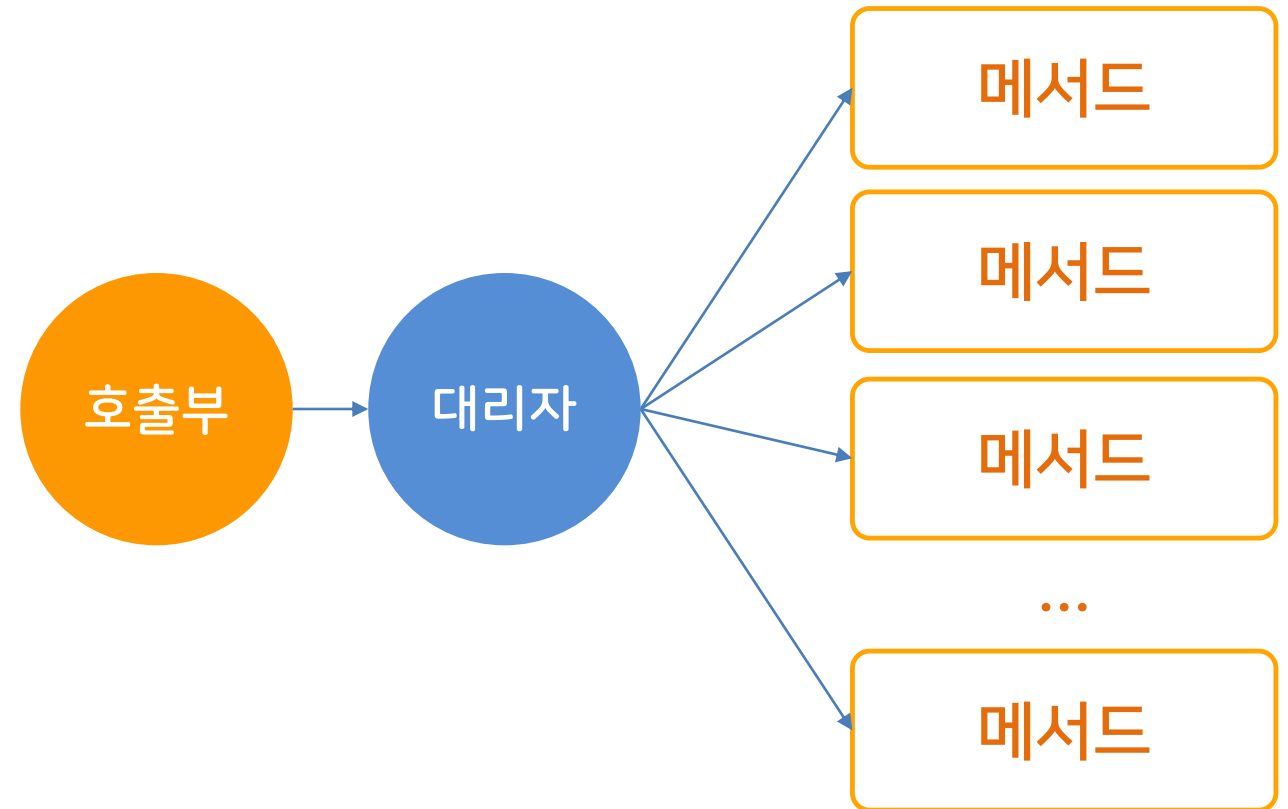
## 18-1. Delegate란?

- Delegate : '대리자'로 직접 호출하지 않고 대리자를 이용한다.

메서드를 직접 호출한다.

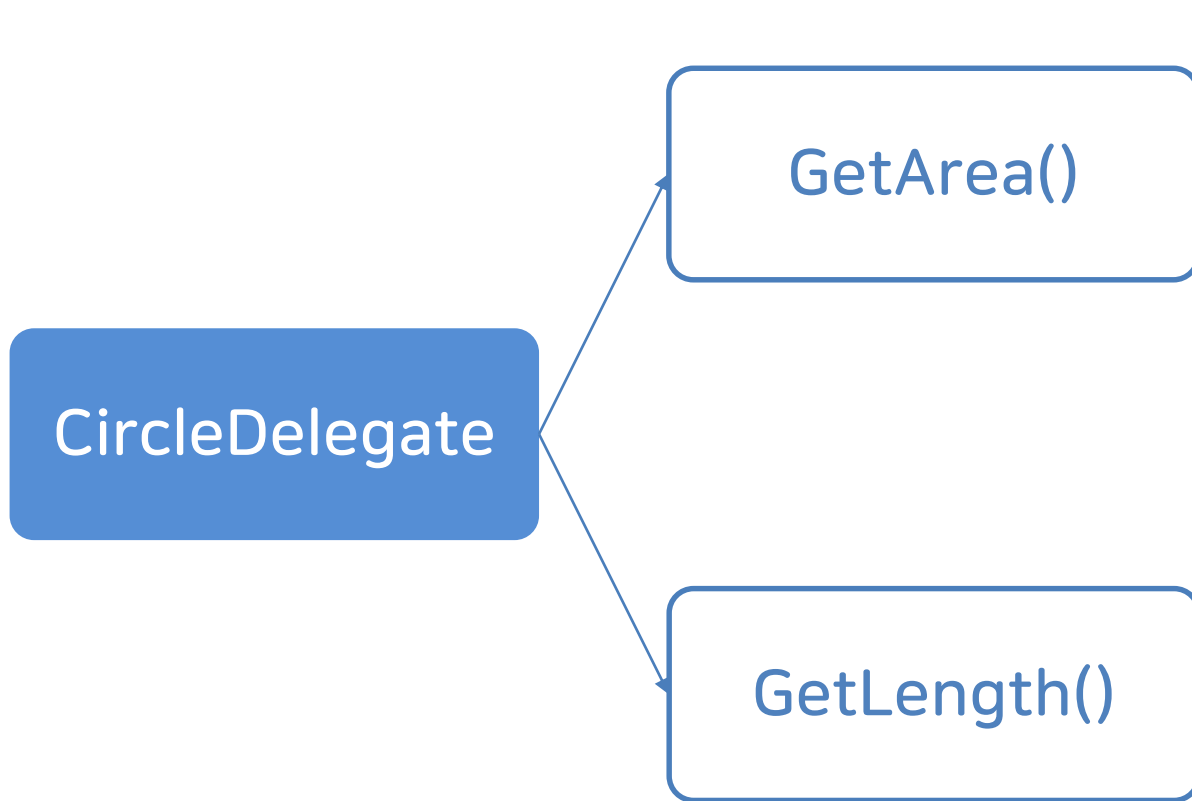


메서드를 대리자를 통해서 호출한다.





- CircleDelegate는 GetArea (), GetLength()를 참조한다.



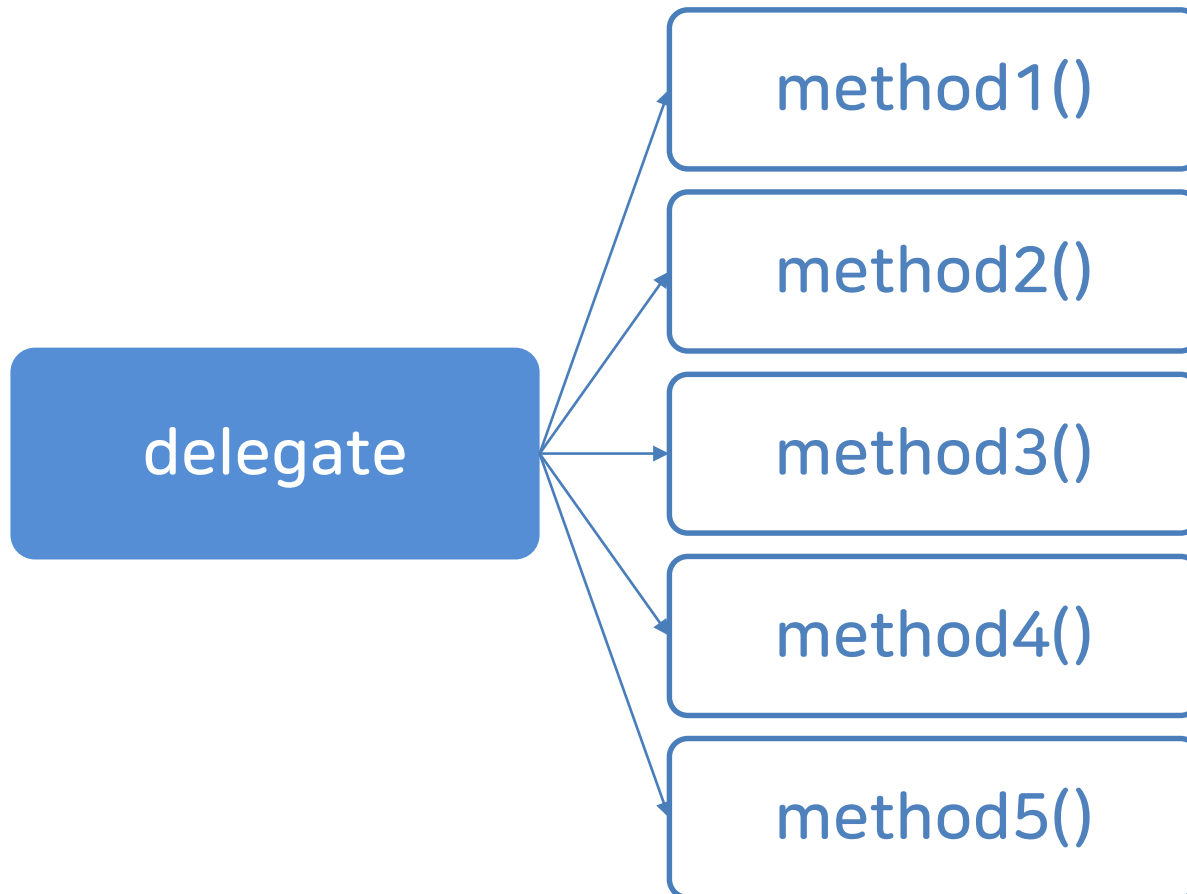
### Delegate의 특징

- 메서드를 참조한다.
- 참조하는 메서드와 반환형, 매개변수가 같다.
- 데이터 타입이다.
- new를 이용해서 생성(인스턴스)한다.
- 연속해서 메서드를 참조할 수 있다.
- 외부에서 호출할 수 있다.



## 18-3. Delegate Chain

- Delegate가 참조하는 메서드를 연속으로 호출한다.



delegate()



method1()실행  
method2()실행  
method3()실행  
method4()실행  
method5()실행



## 18-4. Event란?

ex) lec18/delegatePjt

- Event는 Delegate와 같지만, 외부에서 접근할 수 없다.





## 18-5. Event 사용하기

```
delegate bool PostEventMessage(string msg);
```

```
class PassNoticeEventListener  
{
```

```
    public event PostEventMessage postEventMessage;  
    private string Message;
```

```
    ...
```

```
    public PassNoticeEventListener(string Message)  
    {  
        this.Message = Message;  
    }
```

```
    ...
```

```
    public void SendMessage(string msg)  
    {  
        postEventMessage(msg);  
    }
```

```
}
```

외부에서 접근할 수 없다.

SendMessage()를 통해서  
PostEventMessage 인스턴스에 접근한다.