

# DI container를 이용하여 레거시와 모듈화를 동시에 잡기



안정민

# 소개

---

- 現 한국카카오은행 iOS 개발자 (16.10 ~ )
- 블로그 운영 : [minsone.github.io](http://minsone.github.io)

# 목차

---

- 의존성이란 무엇인가?
- DI란 무엇인가?
- 현실적인 DI 문제
- DI Container
- QnA

의존성이란 무엇인가?

# 의존성이란 무엇인가?

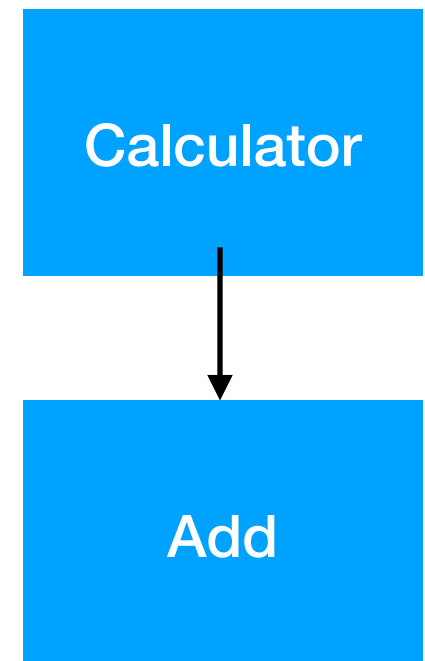
---

```
class Calculator {  
    func add(lhs: Int, rhs: Int) -> Int {  
        return lhs + rhs  
    }  
}
```

# 의존성이란 무엇인가?

---

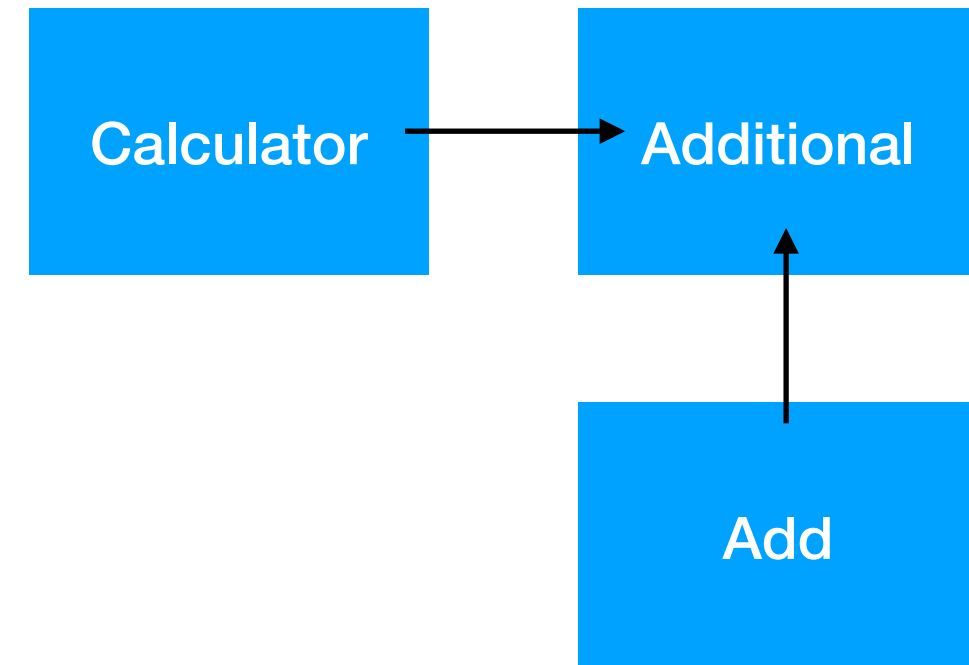
```
class Add {  
  func run(lhs: Int, rhs: Int) -> Int {  
    return lhs + rhs  
  }  
}  
  
class Calculator {  
  private let addCalc: Add  
  init(add: Add) {  
    self.addCalc = add  
  }  
}  
let calc = Calculator(add: Add())
```



# 의존성이란 무엇인가?

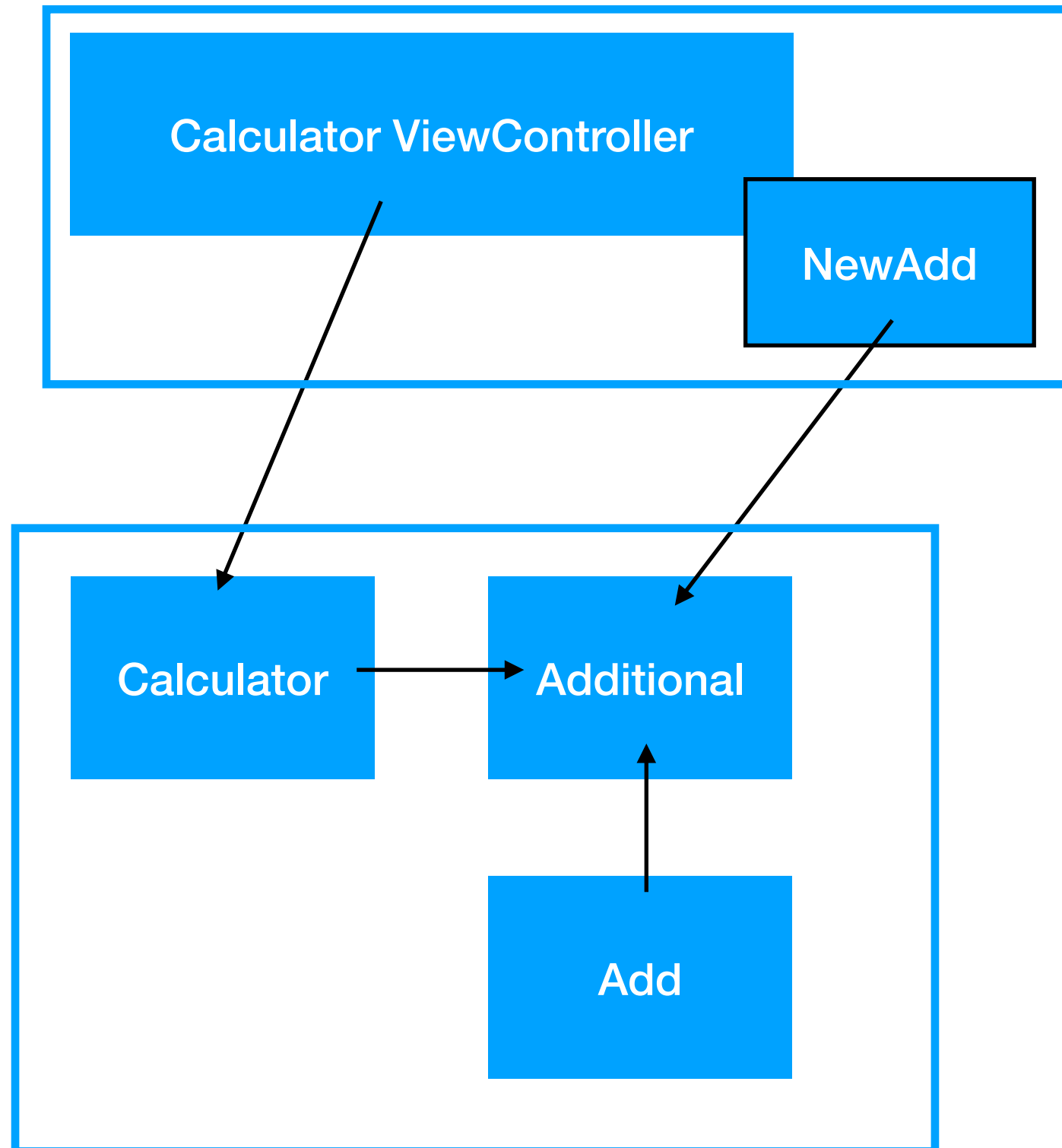
---

```
protocol Additional {  
    func run(lhs: Int, rhs: Int) -> Int  
}  
  
class Add: Additional {  
    func run(lhs: Int, rhs: Int) -> Int {  
        return lhs + rhs  
    }  
}  
  
class Calculator {  
    private let addCalc: Additional  
    func add(lhs: Int, rhs: Int) -> Int {  
        return addCalc.run(lhs: lhs, rhs: rhs)  
    }  
    init(add: Additional) {  
        self.addCalc = add  
    }  
}  
  
let calc = Calculator(add: Add())
```



# 의존성이란 무엇인가?

---





# 현실적인 DI 문제

# 현실적인 DI 문제

---

- 리팩토링의 부담감
- 새로운 기능을 모듈로 만들지만 기존 기능이 필요한 경우
- 모듈간의 의존성이 생기는 경우
- 등등

# 현실적인 DI 문제

---

- 리팩토링의 부담감
  - Adapter를 만들어 기존 코드를 안건드리고 작업.
  - 여력이 생길 때, 하위 기능 & 모듈부터 작게 하나씩 작업.

# 현실적인 DI 문제

---

```
protocol CalculatorAdaptable {
    func add(lhs: Int, rhs: Int) -> Int
}

class CalculatorAdapter: CalculatorAdaptable {
    private let calculator: Calculator

    init(calculator: Calculator) {
        self.calculator = calculator
    }

    func add(lhs: Int, rhs: Int) -> Int {
        return calculator.add(lhs: lhs, rhs: rhs)
    }
}

class ViewController: UIViewController {
    let calc: CalculatorAdaptable
    init(calc: CalculatorAdaptable) {
        self.calc = calc
    }
}
```

# 현실적인 DI 문제

---

- 새로운 기능을 모듈로 만들지만 기존 기능이 필요한 경우
- 비즈니스 로직 서비스를 모듈에 등록 또는 모듈 기능 호출시 주입

# 현실적인 DI 문제

---

- 모듈간의 의존성이 생기는 경우
  - DI Container를 이용

# DI Container