

ARC in Swift

Trình bày: BinhLV

Nhóm: VietHP

Yêu cầu

Đã biết

- ARC, Strong Reference (Retain Cycle)
- Weak Reference trong Objective C
- Optional
- Closure

Mục tiêu

- Phân biệt Weak Reference và Unowned Reference
- Xử lý Strong Reference Cycle giữa các class instance
- Xử lý Strong Reference Cycle khi dùng Closure

Weak Reference

Trong Objective C

```
@property (nonatomic, weak) Person *person;
```

person giữ một liên kết yếu đến instance mà nó tham chiếu đến

Cho nên, nó sẽ không tác động gì đến hoạt động của ARC

Weak Reference

Trong Swift

- Giữ một liên kết yếu đến instance mà nó tham chiếu đến
- Cho phép nil. Phải khai báo kiểu Optional

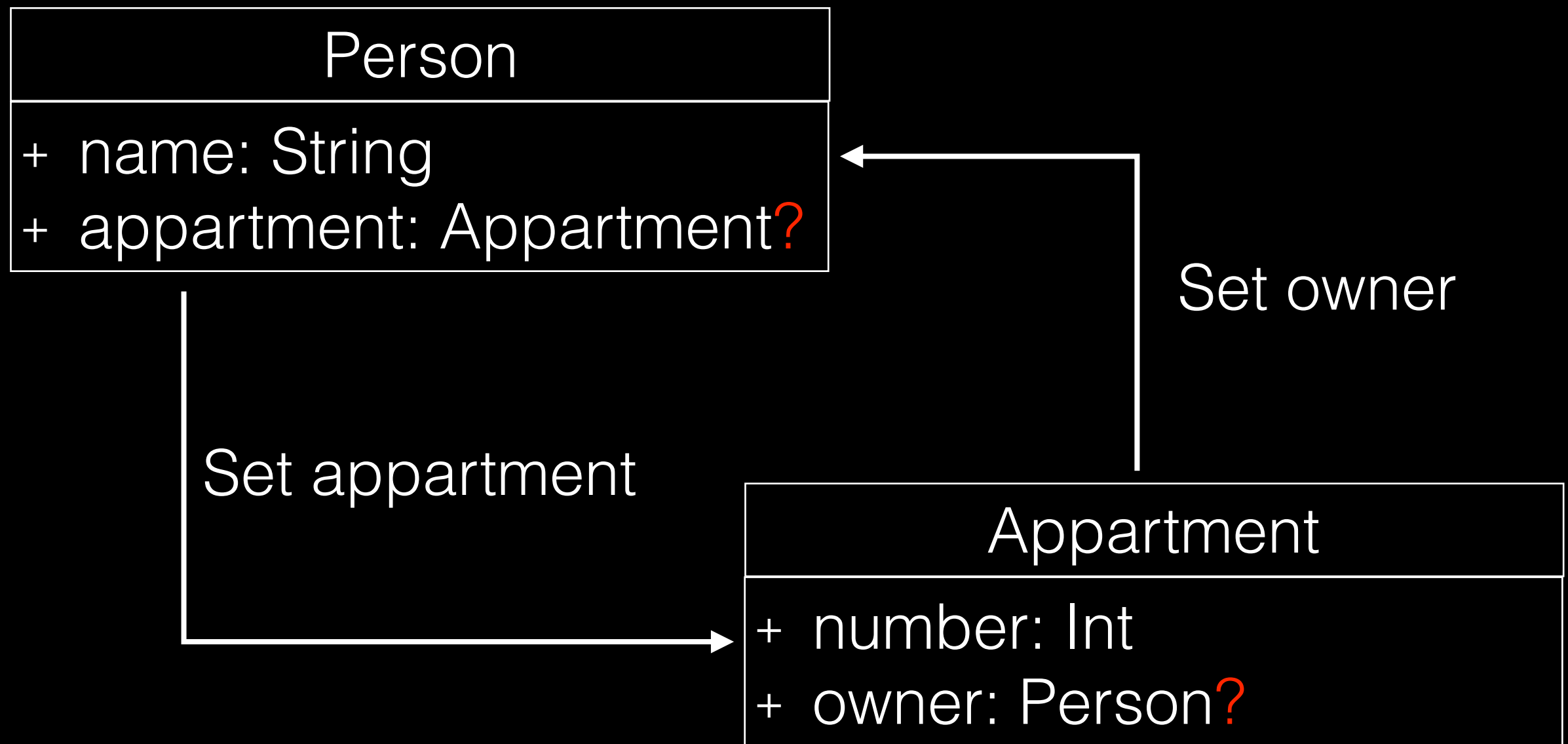
Unowned Reference

- Giữ một liên kết yếu đến instance mà nó tham chiếu đến
- Tuy nhiên, không cho phép nil

Ví dụ Strong Reference Cycle và cách xử lý

- Weak Reference
- Unowned Reference
- Closure

Ví dụ xử lý Strong Reference Cycle bằng weak reference



Ví dụ xử lý Strong Reference Cycle bằng weak reference

Person

- name có kiểu String: tên của một người
- apartment có kiểu Optional: nơi ở của người đó
- Optional vì một người có thể có chỗ ở hoặc không có chỗ ở nào

Ví dụ xử lý Strong Reference Cycle bằng weak reference

Appartment

- number có kiểu Int: địa chỉ của căn hộ
- owner có kiểu Optional: người sở hữu căn hộ đó
- Optional vì một căn hộ có thể có người ở hoặc chưa có ai thuê

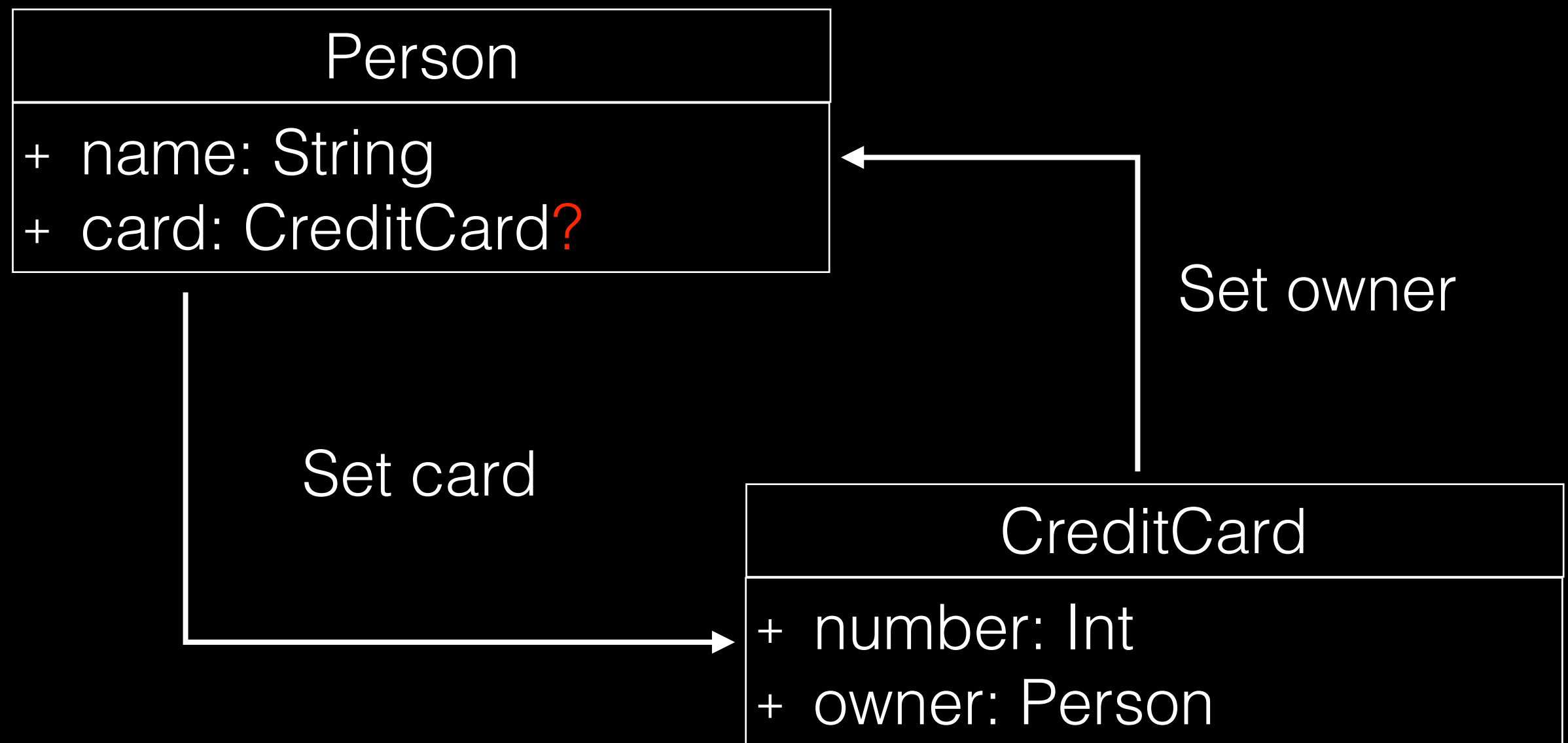
Ví dụ xử lý Strong Reference Cycle bằng weak reference

```
class Person {  
    let name: String  
    var apartment: Apartment?  
    init(name: String) {  
        self.name = name  
    }  
}
```

Ví dụ xử lý Strong Reference Cycle bằng weak reference

```
class Apartment {  
    let number: Int  
    weak var owner: Person?  
    init(number: Int) {  
        self.number = number  
    }  
}
```

Ví dụ xử lý Strong Reference Cycle bằng unowned reference



Ví dụ xử lý Strong Reference Cycle bằng unowned reference

Person

- name có kiểu String: tên của một người
- card có kiểu Optional: thẻ tín dụng của người đó
- Optional vì một người có thể có hoặc không có thẻ tín dụng nào

Ví dụ xử lý Strong Reference Cycle bằng unowned reference

CreditCard

- number có kiểu Int: số thẻ tín dụng
- owner có kiểu Person: người sở thẻ tín dụng đó
- Một thẻ tín dụng thì chắc chắn có người sở hữu

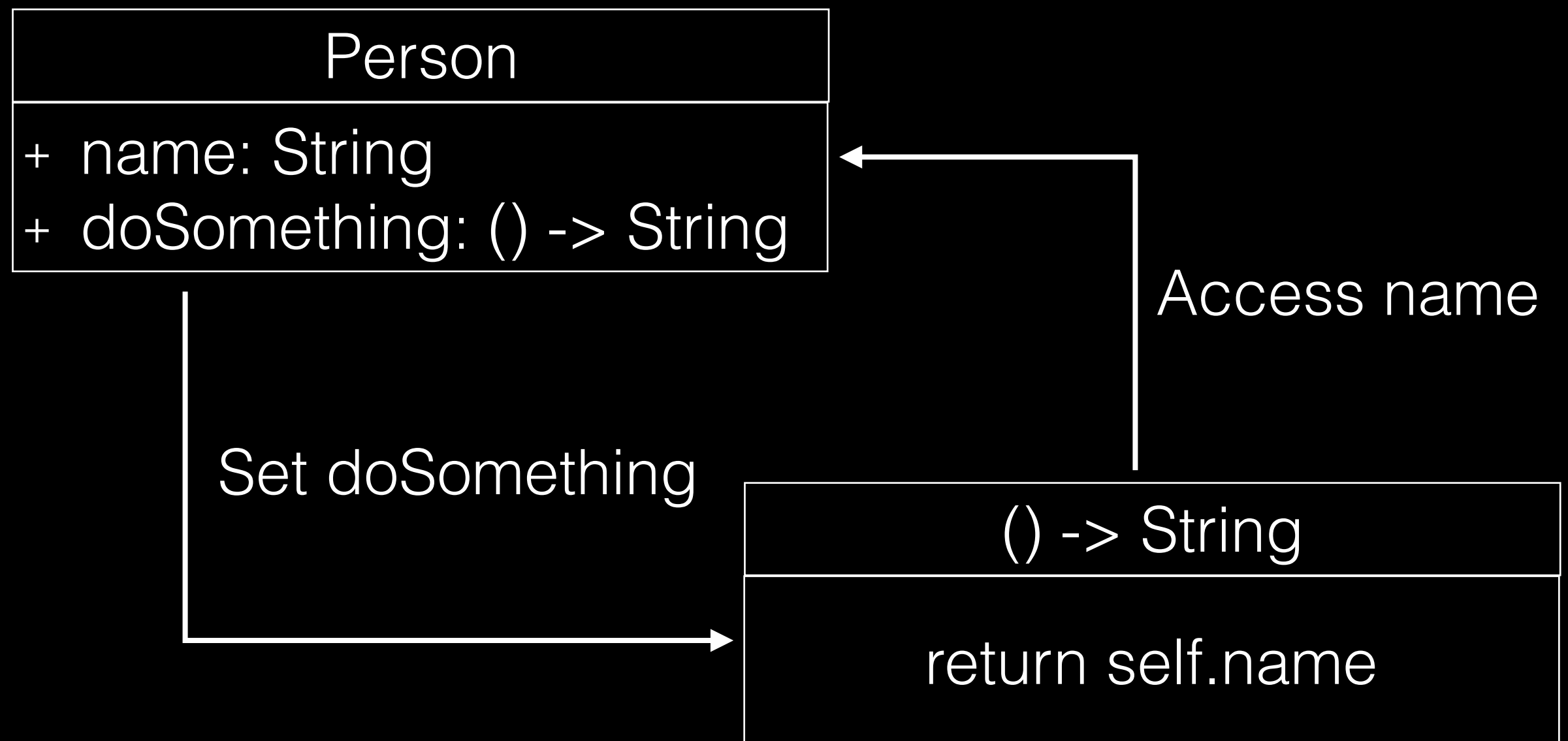
Ví dụ xử lý Strong Reference Cycle bằng unowned reference

```
class Person {  
    let name: String  
    var card: CreditCard?  
    init(name: String) {  
        self.name = name  
    }  
}
```


Ví dụ xử lý Strong Reference Cycle bằng unowned reference

```
class CreditCard {  
    let number: Int  
    unowned let owner: Person  
    init(number: Int, owner: Person){  
        self.number = number  
        self.owner = owner  
    }  
}
```

Ví dụ Strong Reference Cycle trong Closure



Ví dụ Strong Reference Cycle trong Closure

```
class Person {  
    var name: String!  
    lazy var doSomething: () -> String = {  
        [unowned self] in  
        return self.name  
    }  
  
    init(name: String) {  
        self.name = name  
    }  
}
```

QA

Khi nào sử dụng **weak reference**, khi nào sử dụng **unowned reference**

QA

Khai báo weak referenc là một constant được không

```
class Person {
```

```
    weak let name: String?
```

```
}
```

QA

Đoạn code dưới đây có gây ra strong reference cycle không?

```
NSOperationQueue.mainQueue.addOperationWithBlock  
{ () -> Void in
```

```
    self.tableView.reloadData()
```

```
}
```