

Ch13 lab

13장의 App을 만드는 과정에서 사용된 Property Observers, NotificationCenter, Runtime autolayout Constraint 제어에 대한 추가적인 실습을 한다.

1. Property Observers

프로퍼티 옵저버란 변수의 값에 변경이 생기는 경우 변화를 관찰할 수 있다

- willSet: 값이 바뀌기 직전에 호출되는 함수로 새로운 값은 newValue(또는 매개변수)에 저장되어 있다
- didSet: 값이 바뀐 후에 호출되는 함수로 이전의 값은 oldValue(또는 매개변수)에 저장되어 있다.

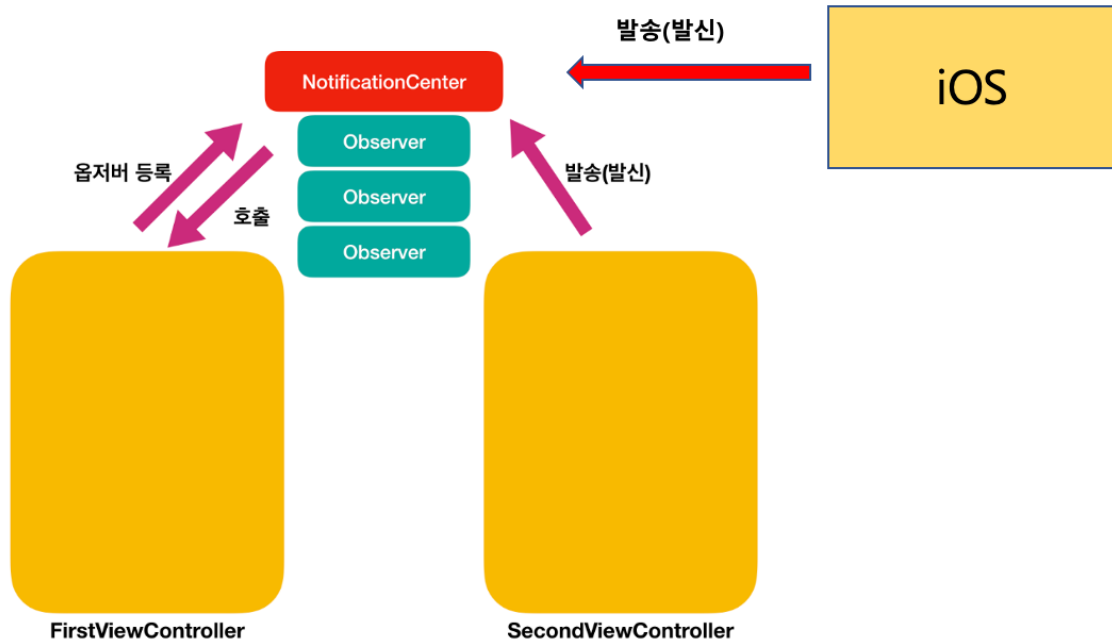
```
var x: Int = 10{
    willSet{    // 매개변수가 없는 경우 새로운 값은 newValue에 저장
        print("new value = ₩(newValue), current value=₩(x)")
    }
    didSet{      // 매개변수가 없는 경우 이전의 값은 oldValue에 저장
        print("old value = ₩(oldValue), current value=₩(x)")
    }
}
x = 20

var y: Int? {
    willSet(a){ // 매개변수가 있는 경우 newValue대신에 매개변수에 새로운 값이 저장
        print("new value = ₩(a), current value=₩(y)")
    }
    didSet(b){  // 매개변수가 있는 경우 oldValue대신에 매개변수에 이전의 값이 저장
        print("old value = ₩(b), current value=₩(y)")
    }
}

y = 30
```

2. Notification

- Notification은 하나의 App내에서 오브젝트(ViewController, View 등)간 통신 기능을 제공한다.
- MQTT의 publisher와 subscriber과 동일한다.



- 주요함수
 - addObserver: Notification을 요청한다.
 - removeObserver: Notification 요청을 취소(withdraw)한다
 - post: Notification 발송을 요청한다.

3. Runtime Autolayout

- Runtime에서 Autolayout Constraint를 마음대로 변경할 수 있다.

4. 실습1

- 주어진 프로그램에서 이미지를 클릭하면 AlbumDetailViewController로 전이하나 이미지가 보이지 않는다. 이유는?
- 해결방법 1
 - Image에 대하여 property observer를 적용한다.

```
var image: UIImage? {  
    didSet {  
        if let imageView = imageView{  
            imageView.image = image  
        }  
    }  
}
```

```
}
```

- 방법2: Notification 적용

```
// prepare of AlbumViewController
```

```
PHCachingImageManager.default().requestImage(for: asset, targetSize: CGSize(),  
contentMode: .aspectFill, options: options, resultHandler: { image, _ in
```

```
    //albumDetailViewController.image = image
```

```
    NotificationCenter.default.post(name: Notification.Name("jmlee"), object: self, userInfo:  
    ["image": image])
```

```
}}
```

```
// AlbumDetailViewController
```

```
override func viewDidLoad() {
```

```
    super.viewDidLoad()
```

```
    imageView.image = image ?? nil
```

```
    NotificationCenter.default.addObserver(self, selector: #selector(receiveImage), name:  
    Notification.Name("jmlee"), object: nil)
```

```
}
```

```
@objc private func receiveImage(_ notification: Notification) {
```

```
    let image = (notification.userInfo as! [String: UIImage])["image"]
```

```
    imageView.image = image
```

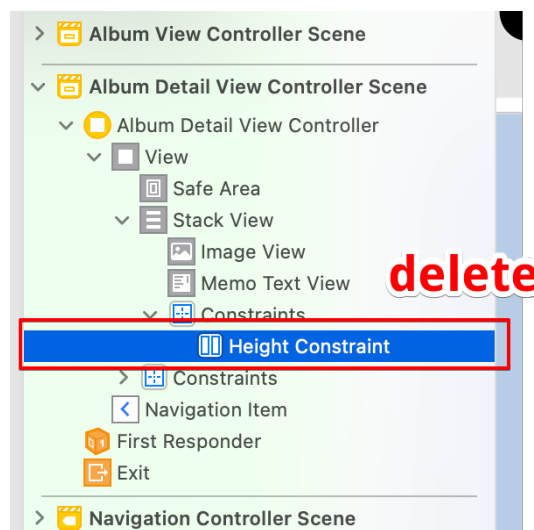
```
}
```

5. 실습2

- 화면의 orientation에 따라 아래 그림과 같이 동적 autolayout을 적용해보자



- AlbumDetailViewController에서 ImageView와 TextView의 height Constraint를 모두 삭제하라



- AlbumDetailViewController.swift에서 다음과 같이 코딩하라

- ◆ 4개의 변수를 선언하라

```
// AlbumDetailViewController.swift의 클래스 변수
var heightConstraintLandscape: NSLayoutConstraint?
var widthConstraintLandscape: NSLayoutConstraint?

var heightConstraintPortrait: NSLayoutConstraint?
var widthConstraintPortrait: NSLayoutConstraint?
```

- ◆ viewDidLoad에서 다음과 같이 코딩하라

```
super.viewDidLoad()
```

```
heightConstraintLandscape = imageView.heightAnchor.constraint(equalTo: memoTextView.heightAnchor, multiplier: 1)
widthConstraintLandscape = imageView.widthAnchor.constraint(equalTo: memoTextView.widthAnchor, multiplier: 1)
```

```
heightConstraintPortrait = imageView.heightAnchor.constraint(equalTo: memoTextView.heightAnchor, multiplier: 2)
widthConstraintPortrait = imageView.widthAnchor.constraint(equalTo: memoTextView.widthAnchor, multiplier: 1)
```

- ◆ viewDidLoad에서 화면 회전에 대한 Notification을 등록하라

```
NotificationCenter.default.addObserver(self, selector:
#selector(changeOrientation), name:
UIDevice.orientationDidChangeNotification, object: nil)
```

- ◆ changeOrientation함수를 다음과 같이 작성하라

```
@objc func changeOrientation(_ notification: Notification?) {

    if UIDevice.current.orientation.isLandscape {
        print("Landscape")
        stackView.axis = .horizontal
        heightConstraintPortrait?.isActive = false
        widthConstraintPortrait?.isActive = false

        heightConstraintLandscape?.isActive = true
        widthConstraintLandscape?.isActive = true
    }

    if UIDevice.current.orientation.isPortrait {
        print("Portrait")

        stackView.axis = .vertical
        heightConstraintLandscape?.isActive = false
        widthConstraintLandscape?.isActive = false

        heightConstraintPortrait?.isActive = true
        widthConstraintPortrait?.isActive = true
    }
}
```

- ◆ 잘되는가?

- 잘안된다. <= 현재의 화면 orientation을 알수 없다.

- 다음과 같이 viewDidLoad의 맨 마지막에 Notification을 posting해보자

```
NotificationCenter.default.post(name:
UIDevice.orientationDidChangeNotification, object: self, userInfo: nil)
```

- 잘안된다. 이유는

- UIApplication.shared.statusBarOrientation.isLandscape 사용

```
if UIApplication.shared.statusBarOrientation.isLandscape {
    // activate landscape changes
} else {
    // activate portrait changes
}
```

- 마무리

- ◆ AlbumDetailViewController.swift의 deinit를 다음과 같이 작성하라.

```
deinit {
    NotificationCenter.default.removeObserver(self, name:
    UIDevice.orientationDidChangeNotification, object: nil)
    NotificationCenter.default.removeObserver(self, name:
    Notification.Name("jmlee"), object: nil)
}
```

