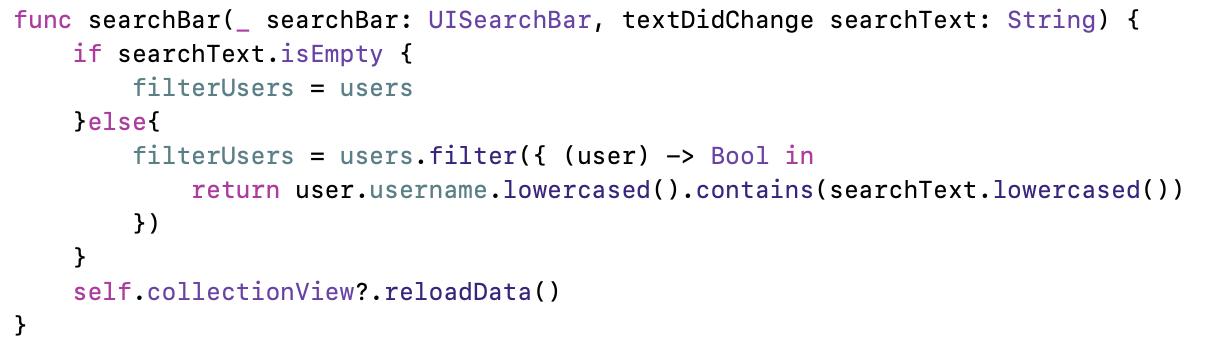
Libraries

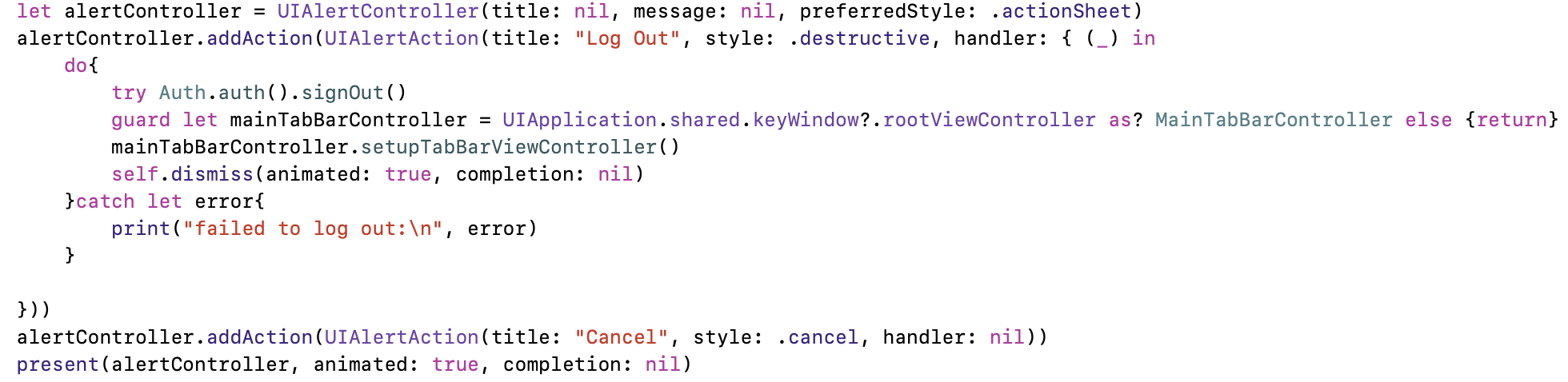
1. UILabel (l)
   1. l.text = “XXX”
   2. l.textAlignment = .center
   3. l.font = UIFont.systemFont(ofSize: 16)
   4. l.numberOfLines = 0//autoLayout
2. UIImageView (iv)
   1. Iv.contentMode = .scaleAspectFill/ .scaleToFill/ .scaleAspectFit
   2. Iv.clipsToBounds = true
3. UITextField (tf)
   1. tf.placeholder = “”
   2. tf.font = UIFont.systemFont(ofSize: 14)
   3. tf.isSecureTextEntry = true
   4. tf.borderStyle = .roundedRect
   5. tf.addTarget(self, action: #selector(handleInput), for: .editingChanged)
4. UIButton (b)
   1. let b = UIButton(type: .system)
   2. b.setTitle(“XX”, for: .normal)
   3. b.setTitleColor(.white, for: .normal)
   4. b.setImage(XX.withRenderingMode(.alwaysOriginal), for: .normal)
   5. b.tintColor = .black
   6. b.layer.cornerRadius = 5
   7. b.setAttributedTitle(attributedTitle, for: .normal)
      1. attributedTitle = NSMutableAttributedString(string: "XXX ", attributes: [NSAttributedStringKey.foregroundColor: UIColor.lightGray])
   8. attributedTitle.append(NSAttributedString(string: "XXX", attributes: [NSAttributedStringKey.foregroundColor: UIColor(r: 17, g: 154, b: 237)]))
5. UIStackView
   1. let stackView = UIStackView(arrangedSubviews: [XXX, XXX, XXX])
      1. stackView.axis = .horizontal
      2. stackView.distribution = .fillEqually
      3. stackView.spacing = 10
6. UISearchBar (sb)
   1. --------variable---
   2. sb.placeholder = “XXX”
   3. UITextField.appearance(whenContainedInInstancesOf: [UISearchBar.self]).backgroundColor = UIColor(r: 240, g: 240, b: 240)
   4. sb.delegate = self
   5. -----------method------
   6. textDidChange



1. UIPageControl (pc)
   1. pc.currentPage = 0
   2. pc.numberOfPages = 3
   3. pc.currentPageIndicatorTintColor = .black
   4. pc.pageIndicatorTintColor = UIColor(r: 232, g: 236, b:241)
   5. ----method-----

UIAlertController

1. Logout Alert



Switch pages(UINavigation)

1. Create a new UIViewController page
   1. let xxController = XXController()
   2. let navXXController = UINavigationController(rootViewController: xxController)
   3. present(navXXController, animated: true, completion: nil)
   4. slef.dismiss(animated: true, completion: nil)
2. Create a new UICollectionViewController page
   1. let layout = UICollectionViewFlowLayout()
   2. let xxController = XXController(collectionViewLayout: layout)
   3. let navXXController = UINavigationController(rootViewController: xxController)
   4. present(navXXController, animated: true, completion: nil)
   5. slef.dismiss(animated: true, completion: nil)
3. Create a page based on previous page (subPage)
   1. ------in first page-----
   2. let xxController = XXController()
   3. navigationController?.pushViewController(xxController, animated: true)
   4. ------go back in second page----
   5. navigationController?.popViewController(animated: true)
4. setup Navigation
   1. navigationItem.title = “XXX”
   2. navigationItem.tileView = UIImageView(image: XXX)
   3. navigationItem.rightBarButtonItem = UIBarButtonItem(title/image: “XX”, style: .plain, target: self, action: #selector(handleXXX))

TabBar

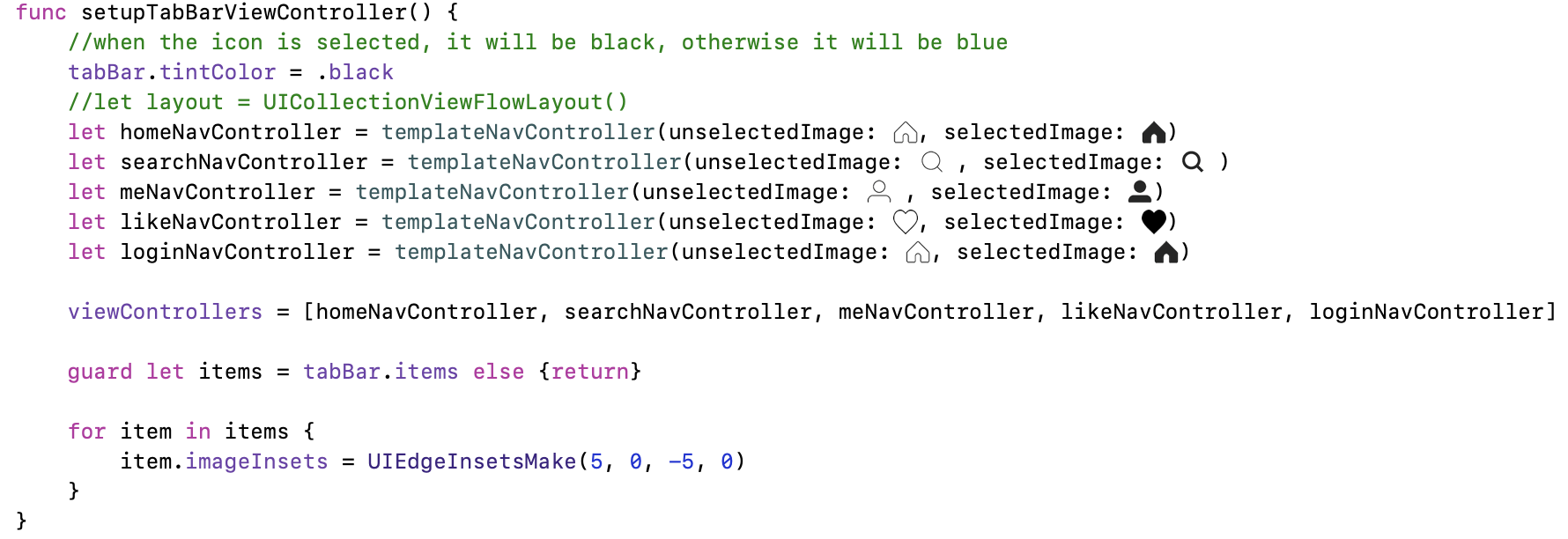
MainTabBarController

Subclass: UITabBarController, UITabBarControllerDelegate

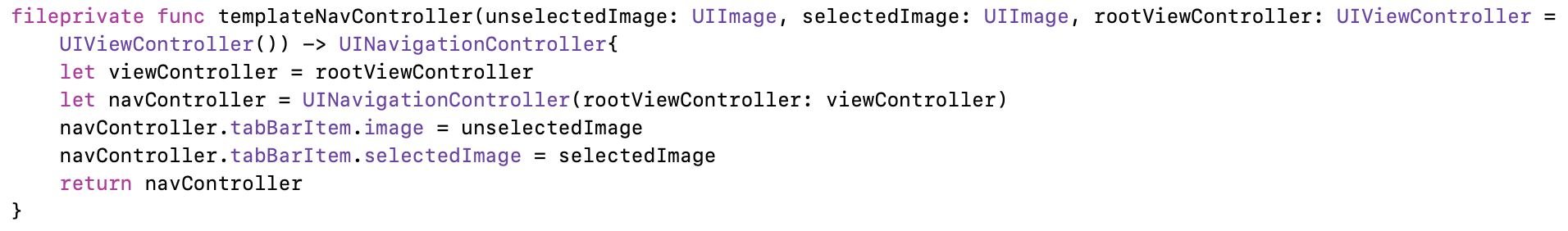
1. viewDidLoad //override func
   1. view.backgroundColor = .red
   2. delegate = self
   3. setupTabBarViewController()



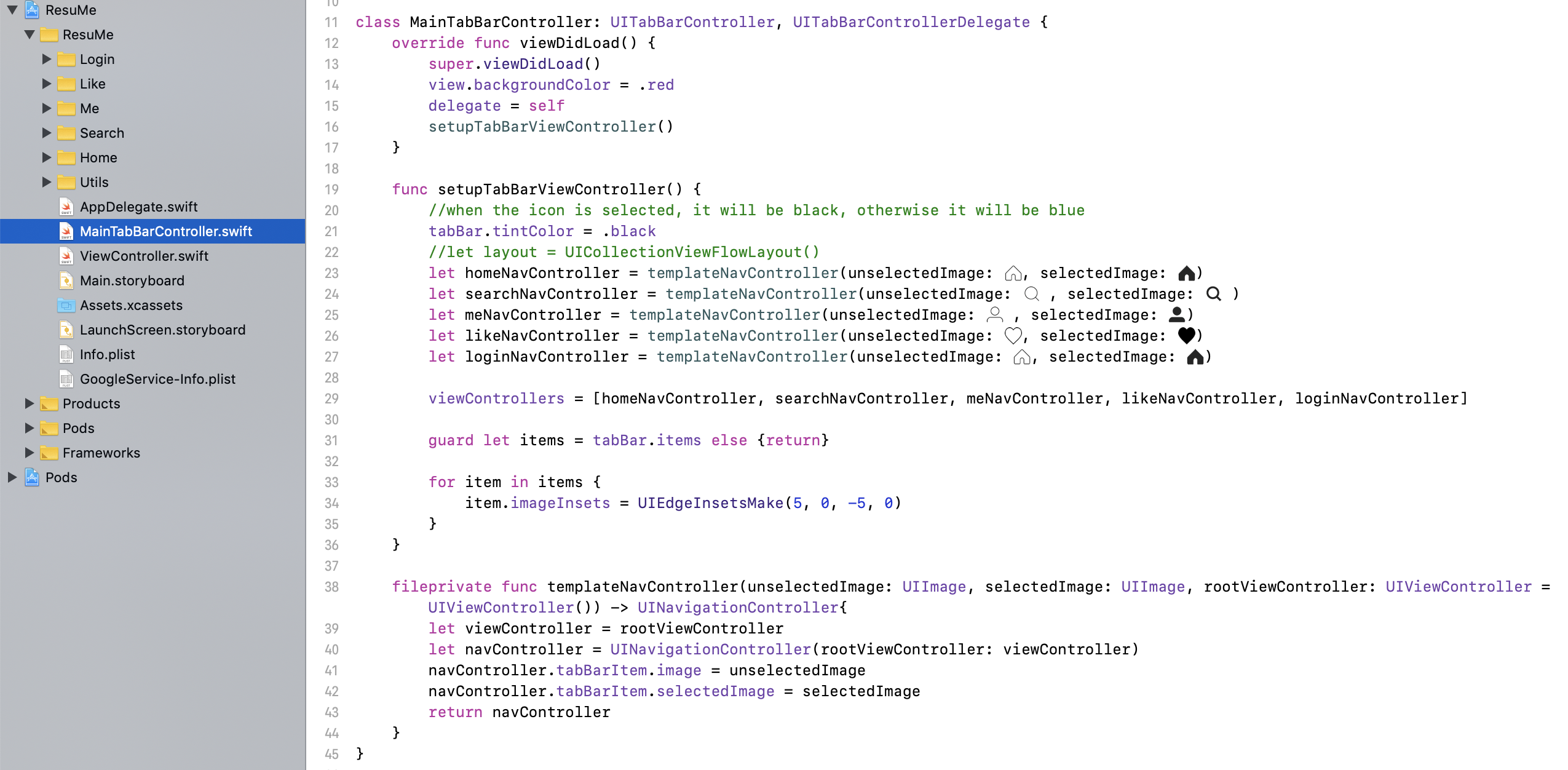
1. setupTabBarViewController //func
   1. tabBar.tintColor = .black
   2. let xxNavController = templateNavController() //custom func
   3. viewControllers = [xxNavController, …, xxNavController]
   4. adjust icons position



1. templateNavController(unselectedImage: UIImage, selectedImage: UIImage, rootViewController: UIViewController = UIViewController()) -> UINavigationController



Overall



AppDelegate

1. window = UIWindow()
2. window?.rootViewController = MainTabBarController()



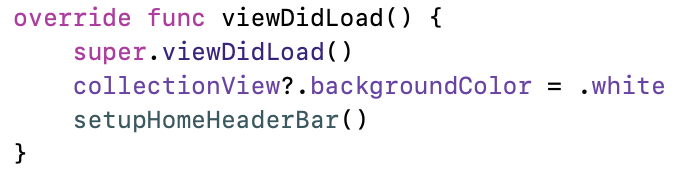


Collection View

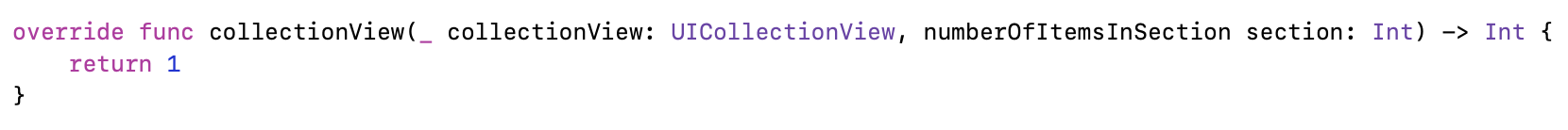
CollectionViewController

Subclass: UICollectionViewController, UICollectionViewDelegateFlowLayout

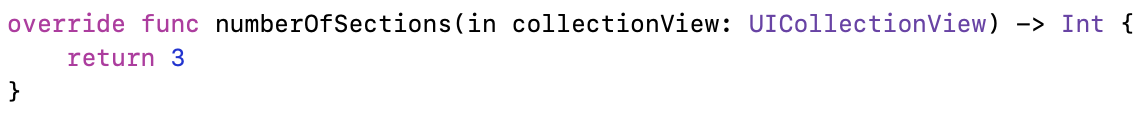
1. viewDidLoad
   1. collectionView?.backgroundColor = .white



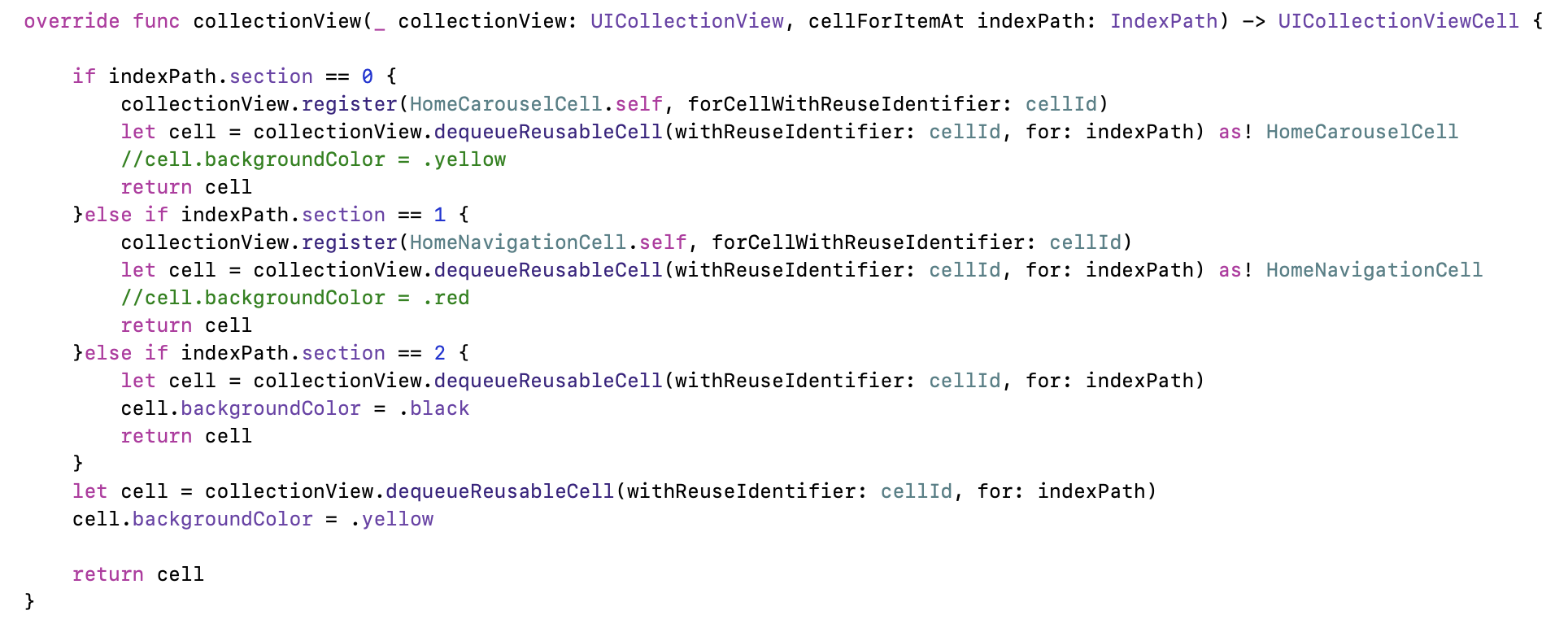
1. numberOfItem



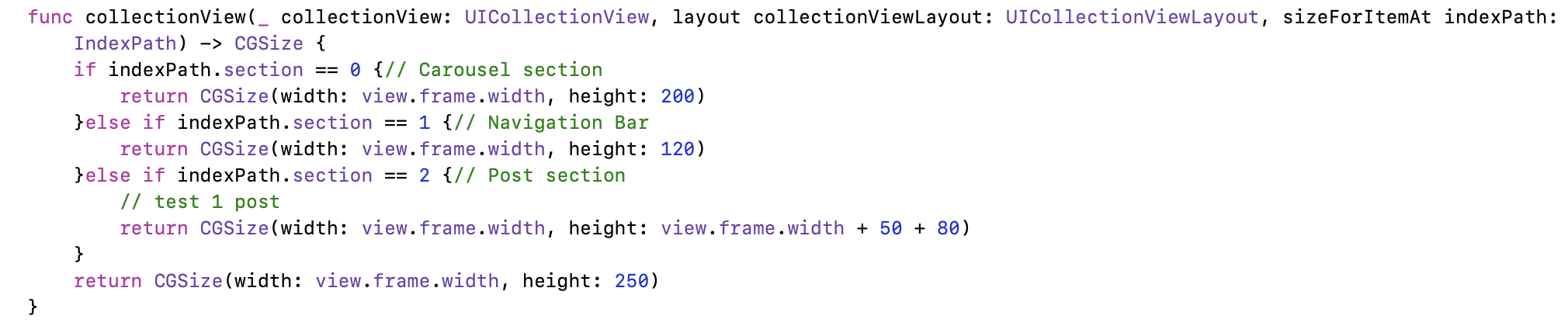
1. numberOfSections // One section only has one header and one footer



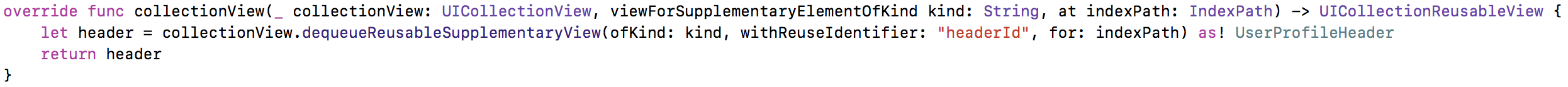
1. cellForItemAt
   1. // test version
      1. collectionView.register( [UICollectionViewCell.self], …)
      2. cell.backgroundColor = .yellow
   2. // implement version
      1. collectionView.register( [xxx.self], …)
   3. let cell = collectionView.dequeueReuseableCell(…)



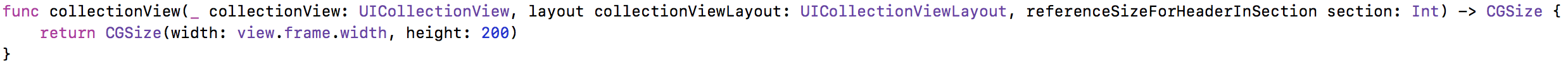
1. sizeForItemAt



1. header/footer
   1. viewForSupplementryElementOfKind
      1. let header = collectionView.dequeueReusableSupplementary



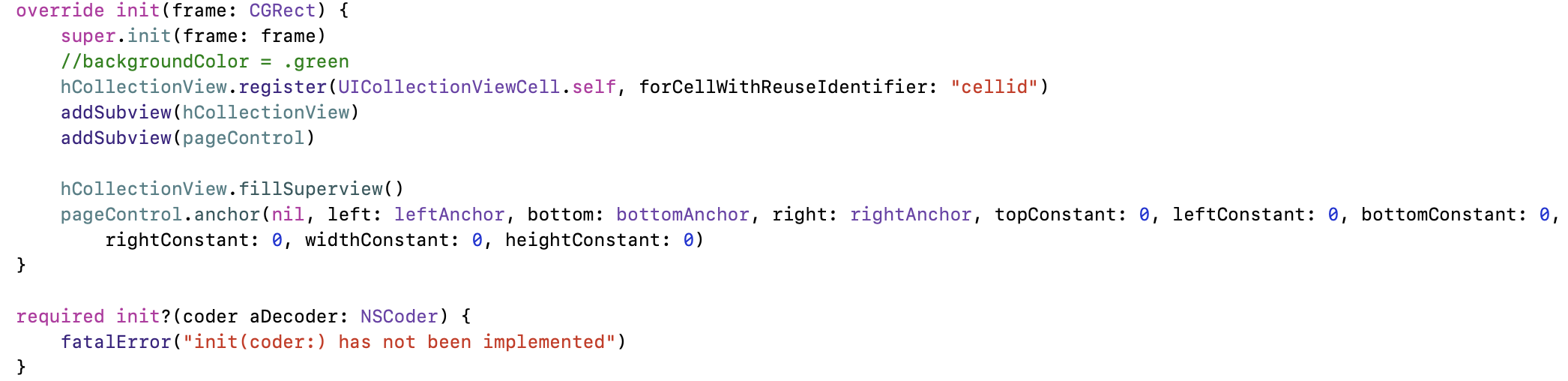
* 1. refsizeheader or refsizefooter



CollectionViewCell

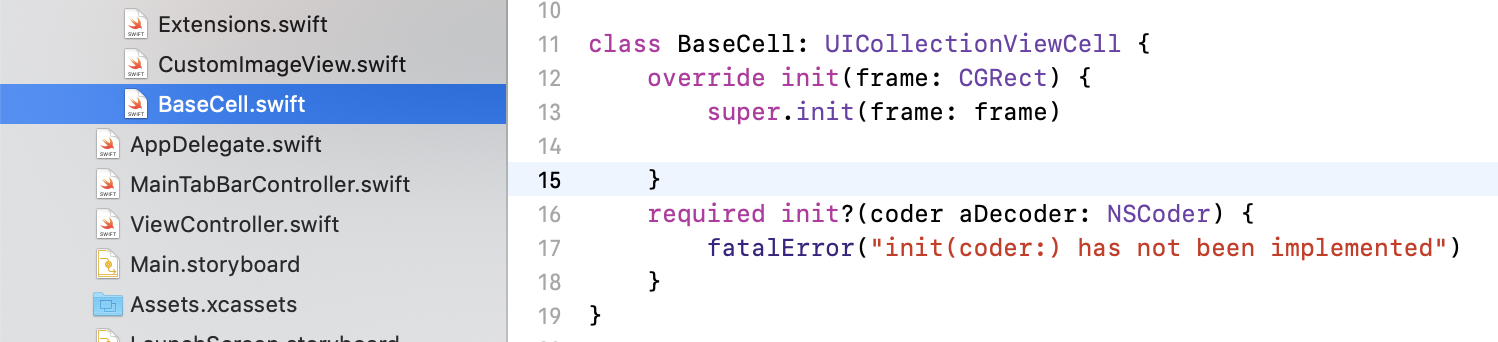
Subclass: UICollectionViewCell

1. init
   1. addSubview
   2. adjust postion methods (Extention of UIView)
      1. xx.anchor
      2. xx.fillSuperview
      3. xx.anchorCenterSuperview
      4. xx.anchorCenterXToSuperview
      5. xx.anchorCenterYToSuperview
      6. anchorWithReturnAnchors

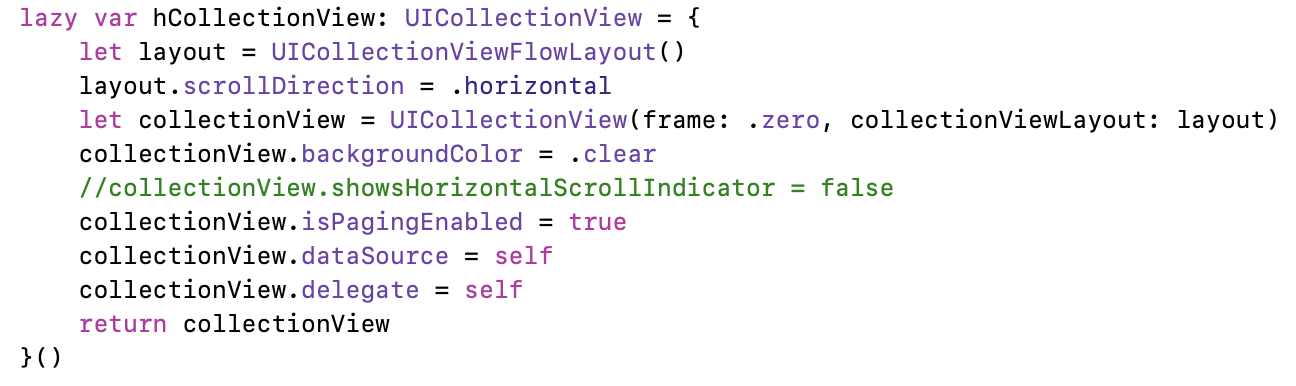


1. customCell
   1. A CollectionView in a CollectionViewCell
      1. Subclass: BaseCell, UICollectionViewDelegateFlowLayout, UICollectionViewDataSource, UICollectionViewDelegate

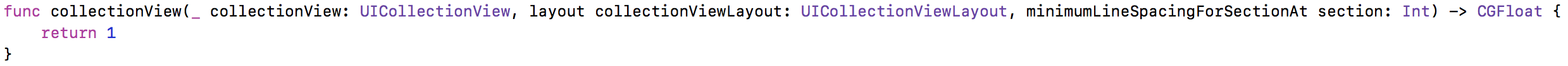




* + 1. Create UICollectionView Object



* + 1. numberOfItems
    2. cellForItemAt
    3. sizeForItemAt
  1. minimize gaps
     1. miniLineSpacing



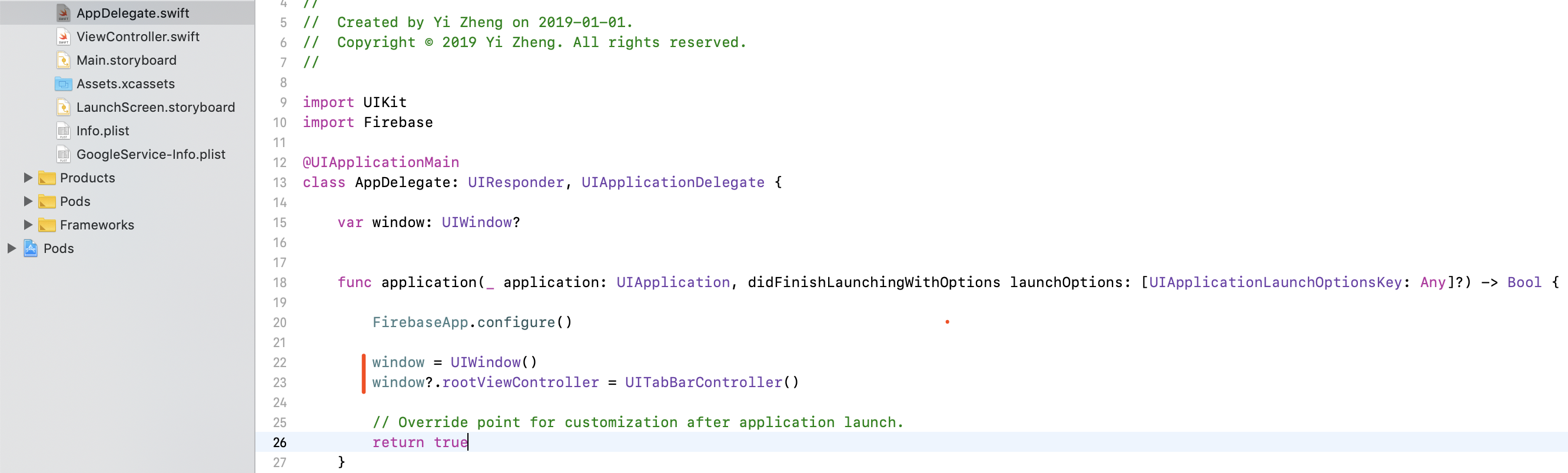
* + 1. miniIterItem

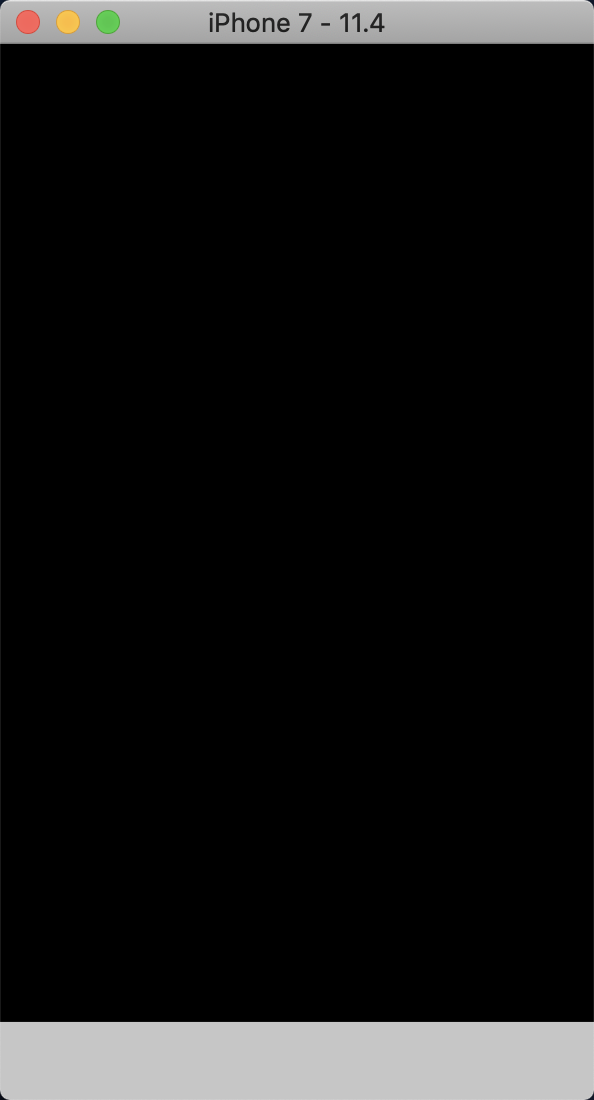


* + 1. insetForSection

Short cuts

1. Create a file -> command + n
2. Run app -> command + r
3. Build app -> command + b
4. Fold method/class -> option + command + <-
5. Go to AppDelegate.swift file





1. Create MainTabBarController.swift file

subclass: UITabBarController, UITabBarControllerDelegate

* 1. Import UIKit
  2. Add viewDidLoad
  3. In AppDelegate.swift:
     1. Add “window = UIWindow()/ window?.rootViewController = MainTableBarController()”

