STRV

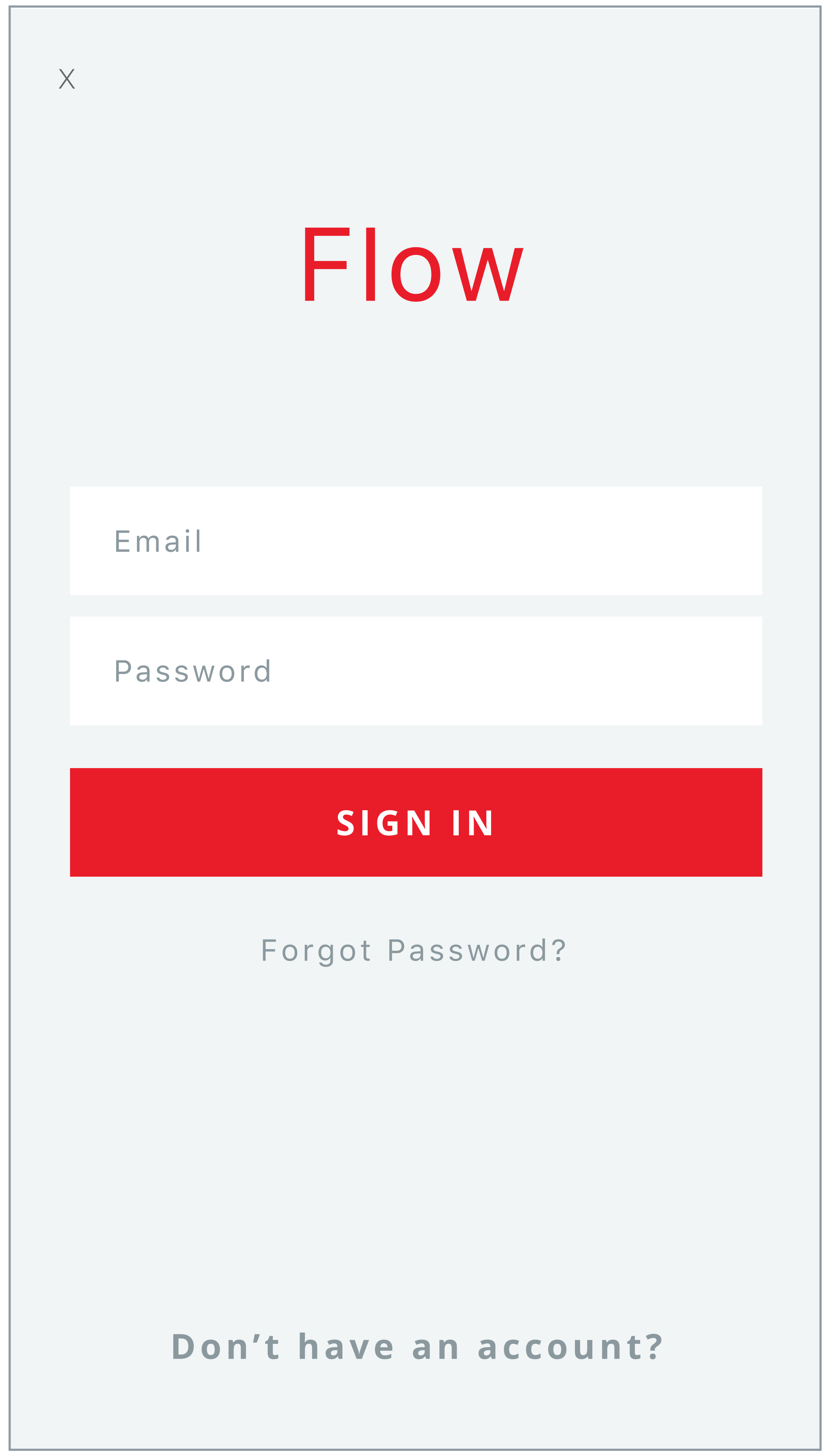
FLOW COORDINATOR

Alexandre Tavares, iOS Developer at STRV

STRV

THE PROBLEM

THE PROBLEM



```
class SignInViewController: UIViewController {
    let viewModel = SignInViewModel()

    func signInButtonTapped(_ sender: UIButton) {
        viewModel.signIn(email: emailTextField.text, password: passwordTextField.text) { (success) in
            if success {
                self.dismiss(animated: true, completion: nil)
            }
        }
    }

    func createAccountButtonTapped(_ sender: UIButton) {
        let viewController = CreateAccountViewController()
        viewController.email = emailTextField.text
        self.navigationController?.pushViewController(viewController, animated: true)
    }

    func forgotPasswordButtonTapped(_ sender: UIButton) {
        let viewController = ForgotPasswordViewController()
        viewController.email = emailTextField.text
        self.navigationController?.pushViewController(viewController, animated: true)
    }
}
```

```swift
class SignInViewController: UIViewController {
    let viewModel = SignInViewModel()

    func signInButtonTapped(_ sender: UIButton) {
        viewModel.signIn(email: emailTextField.text, password: passwordTextField.text) { (success) in
            if success {
                self.dismiss(animated: true, completion: nil)
            }
        }
    }

    func createAccountButtonTapped(_ sender: UIButton) {
        let viewController = CreateAccountViewController()
        viewController.email = emailTextField.text
        self.navigationController?.pushViewController(viewController, animated: true)
    }

    func forgotPasswordButtonTapped(_ sender: UIButton) {
        let viewController = ForgotPasswordViewController()
        viewController.email = emailTextField.text
        self.navigationController?.pushViewController(viewController, animated: true)
    }
}
```

WHAT IS A COORDINATOR?

COORDINATORS

- Responsible for the Application Flow
- Create, show and dismiss ViewControllers and Child Coordinators
- Removes responsibility from the ViewController
- Architecture agnostic

COORDINATORS

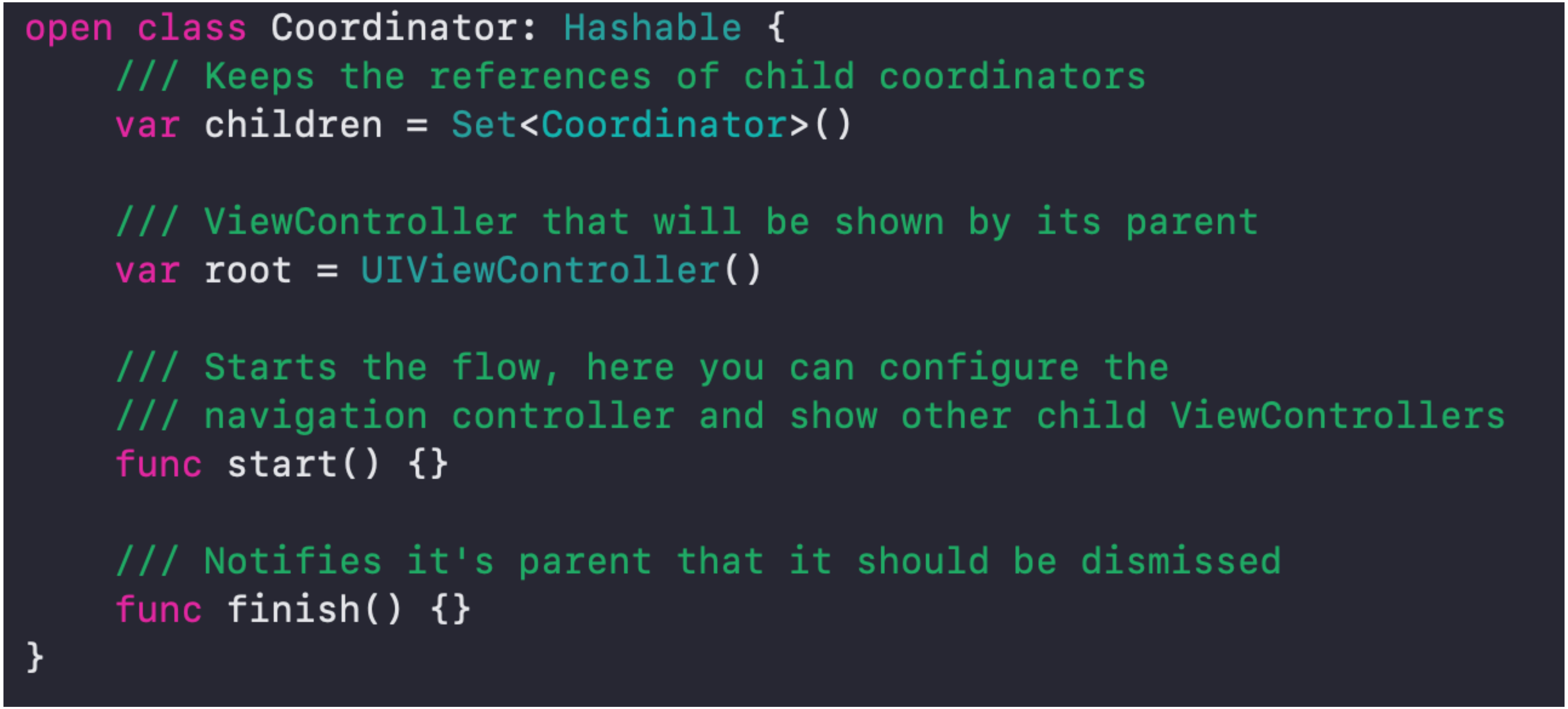
The coordinator should know how to

- ✅ Show ViewControllers that belong to it's flow
- ✅ Start new Flows
- ✅ Finish showing ViewControllers and other Flows

The coordinator should not know

- ❌ Where it belongs in the Application Flow
- ❌ How to dismiss itself

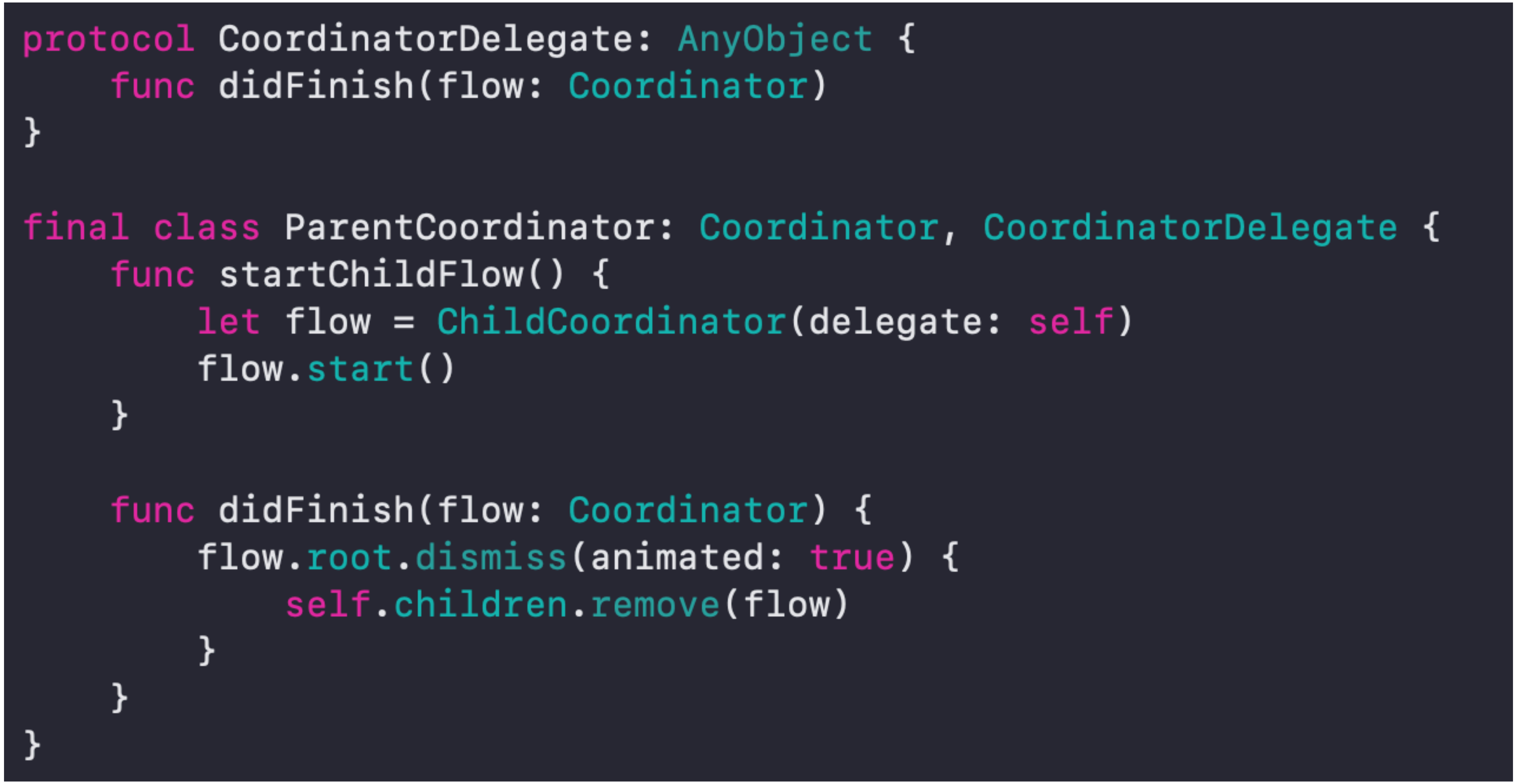
DEFAULT IMPLEMENTATION



```swift
open class Coordinator: Hashable {
    /// Keeps the references of child coordinators
    var children = Set<Coordinator>()

    /// ViewController that will be shown by its parent
    var root = UIViewController()

    /// Starts the flow, here you can configure the
    /// navigation controller and show other child ViewControllers
    func start() {}

    /// Notifies it's parent that it should be dismissed
    func finish() {}
}
```

COORDINATOR FINISH

- By delegation

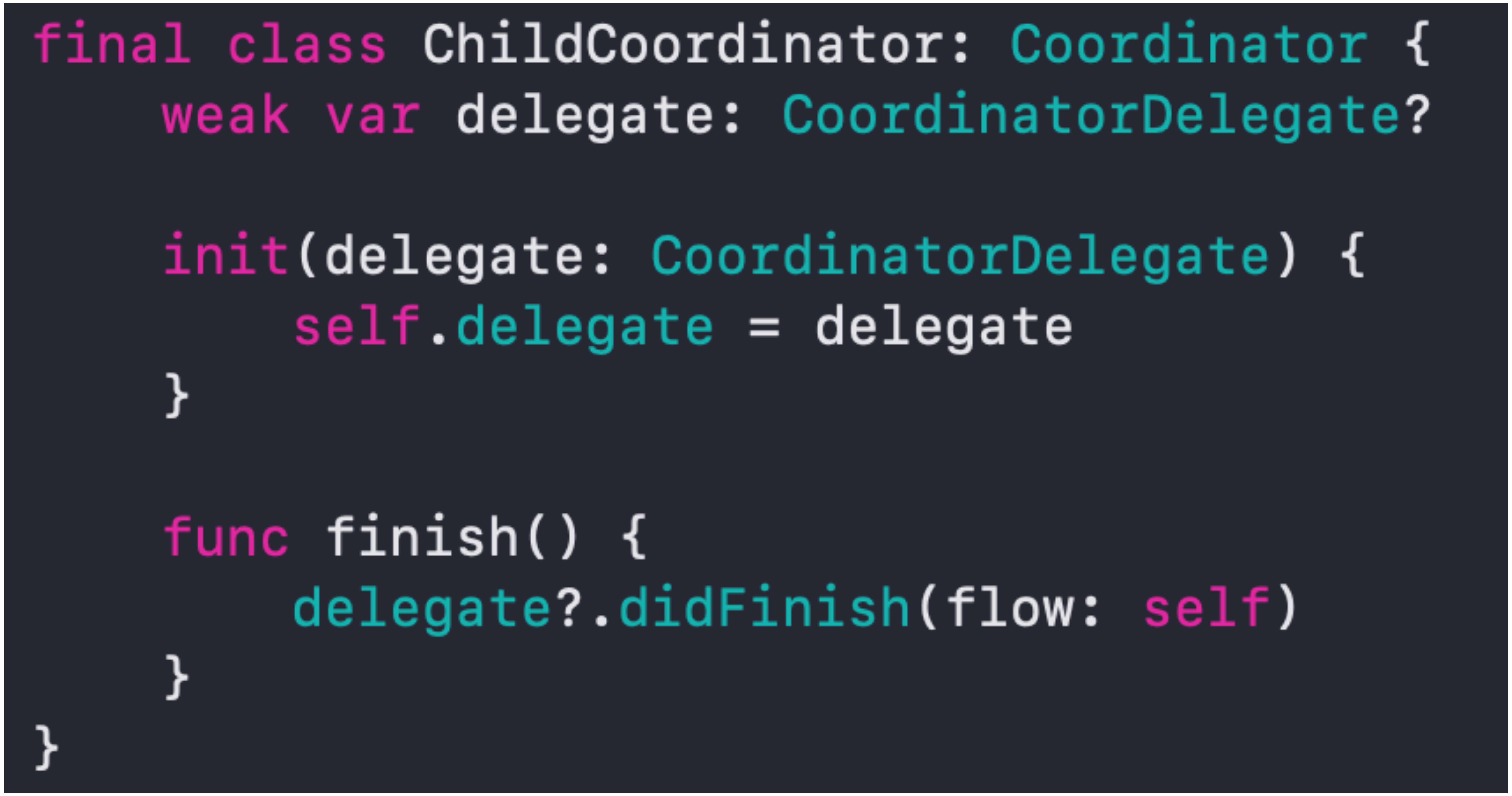


```swift
protocol CoordinatorDelegate: AnyObject {
    func didFinish(flow: Coordinator)
}

final class ParentCoordinator: Coordinator, CoordinatorDelegate {
    func startChildFlow() {
        let flow = ChildCoordinator(delegate: self)
        flow.start()
    }

    func didFinish(flow: Coordinator) {
        flow.root.dismiss(animated: true) {
            self.children.remove(flow)
        }
    }
}
```

STRV

COORDINATOR FINISH

- By delegation

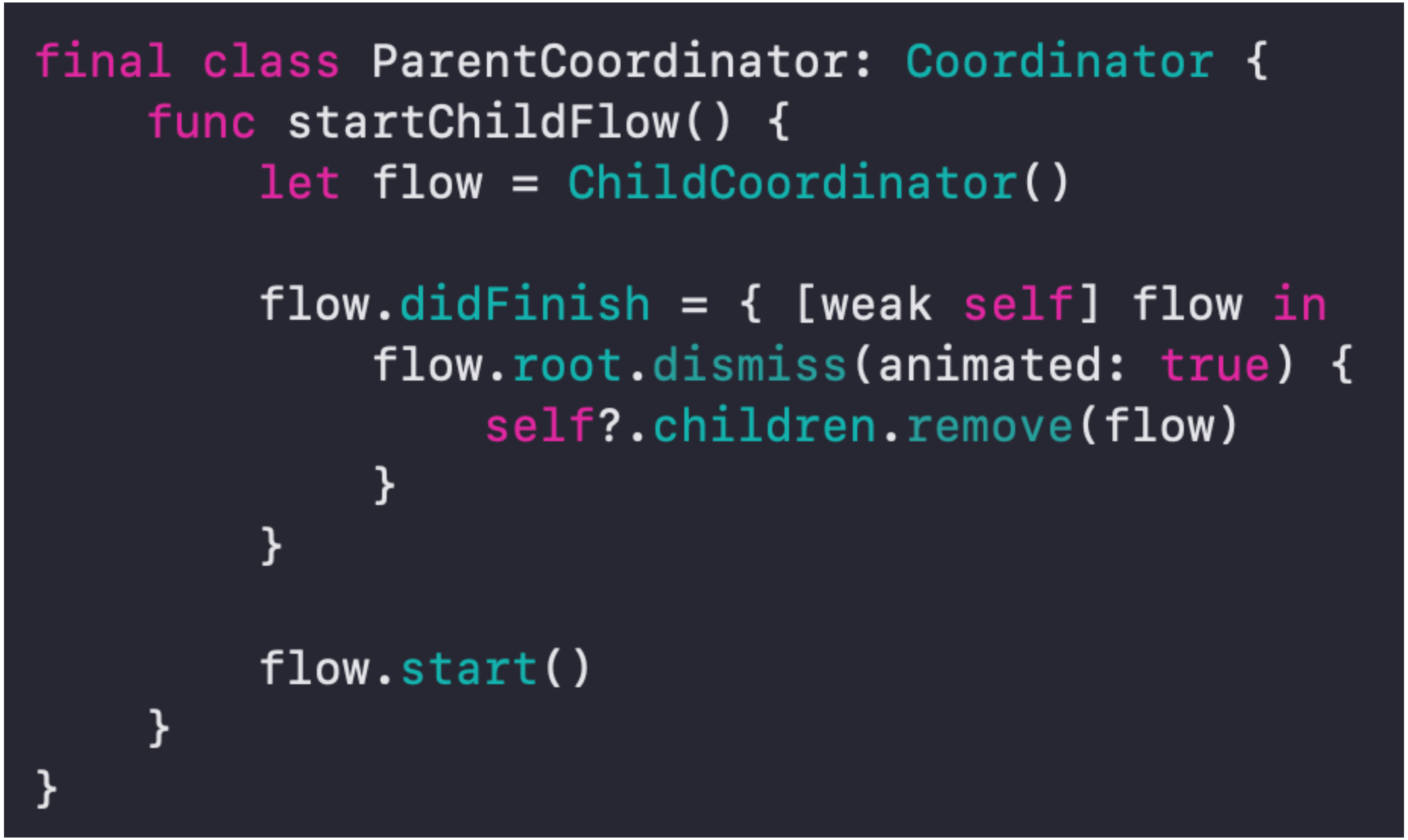


```
final class ChildCoordinator: Coordinator {
    weak var delegate: CoordinatorDelegate?

    init(delegate: CoordinatorDelegate) {
        self.delegate = delegate
    }

    func finish() {
        delegate?.didFinish(flow: self)
    }
}
```

STRV

COORDINATOR FINISH

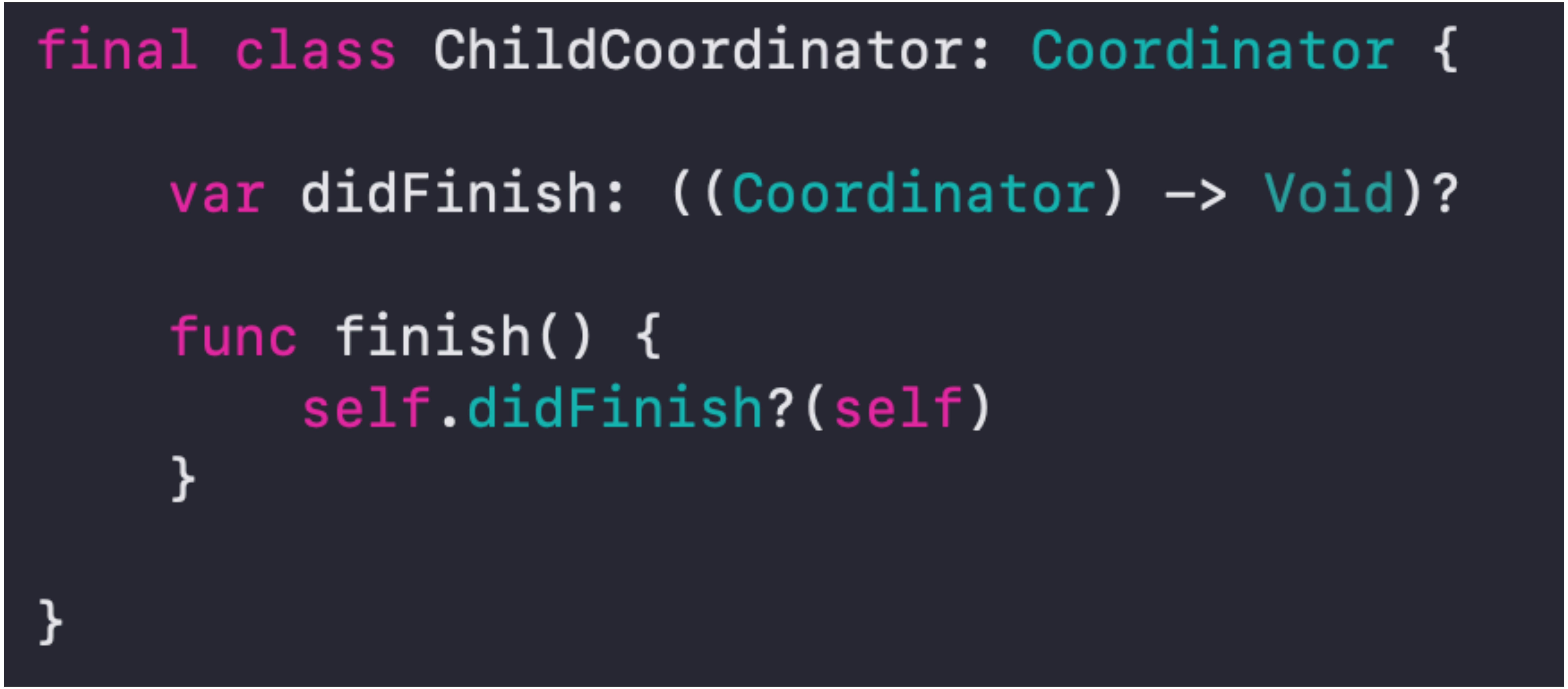
- Using closures



```swift
final class ParentCoordinator: Coordinator {
    func startChildFlow() {
        let flow = ChildCoordinator()

        flow.didFinish = { [weak self] flow in
            flow.root.dismiss(animated: true) {
                self?.children.remove(flow)
            }
        }

        flow.start()
    }
}
```

COORDINATOR FINISH

- Using closures



```swift
final class ChildCoordinator: Coordinator {

    var didFinish: ((Coordinator) -> Void)?

    func finish() {
        self.didFinish?(self)
    }

}
```

TYPES OF FLOW

Vertical

- Is presented by the parent flow
- Doesn't share the parent's Navigation Controller

Horizontal

- Is pushed by the parent flow
- Shares the parent's Navigation Controller

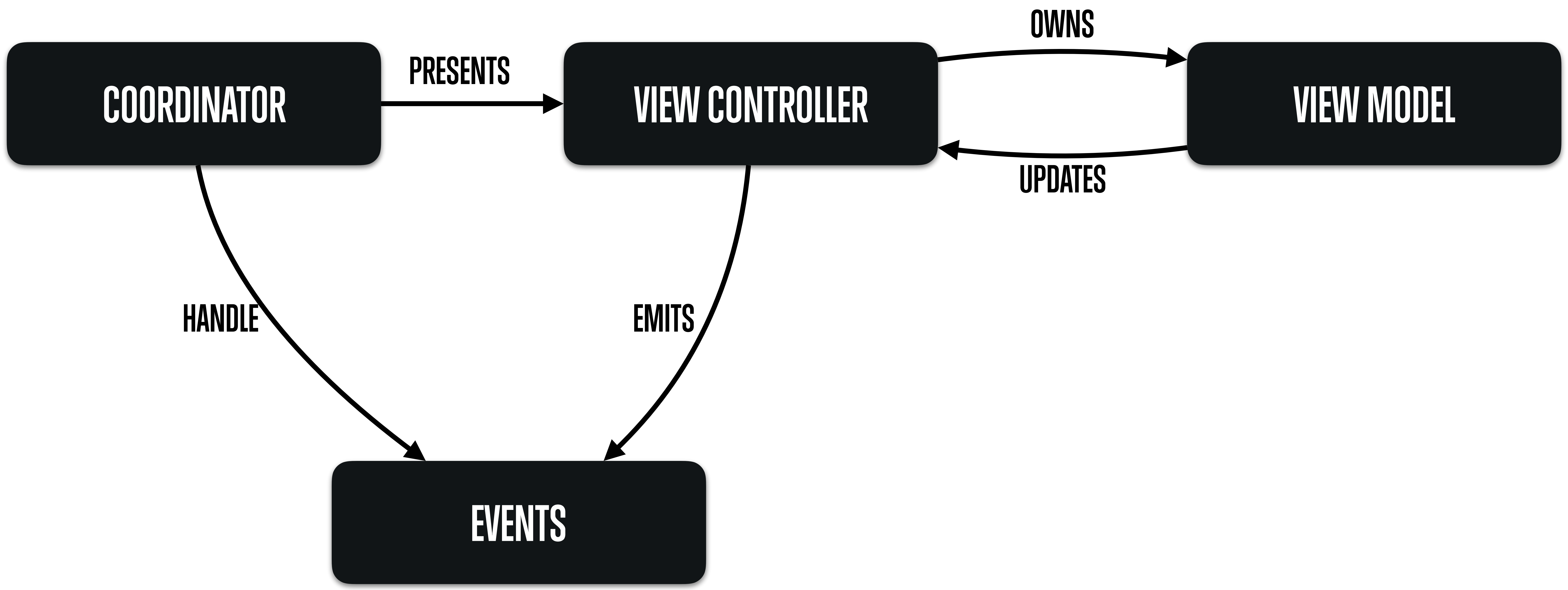
COORDINATOR

- Using RX

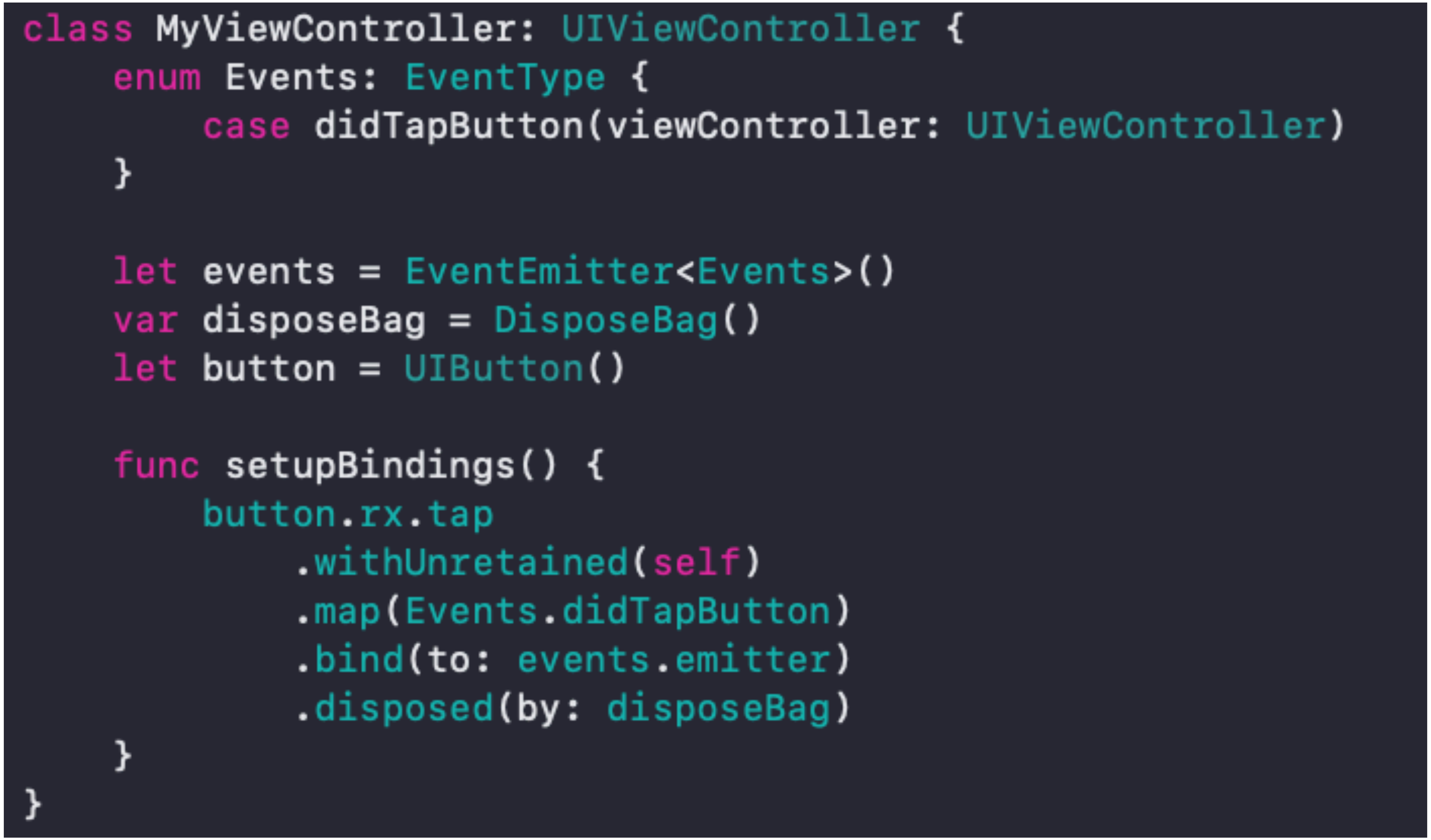
VIEW CONTROLLERS

VIEW CONTROLLERS

- Always created and shown by the Coordinator
- Should only know about it's View and ViewModel
- Receives data from the ViewModel and emits interaction events

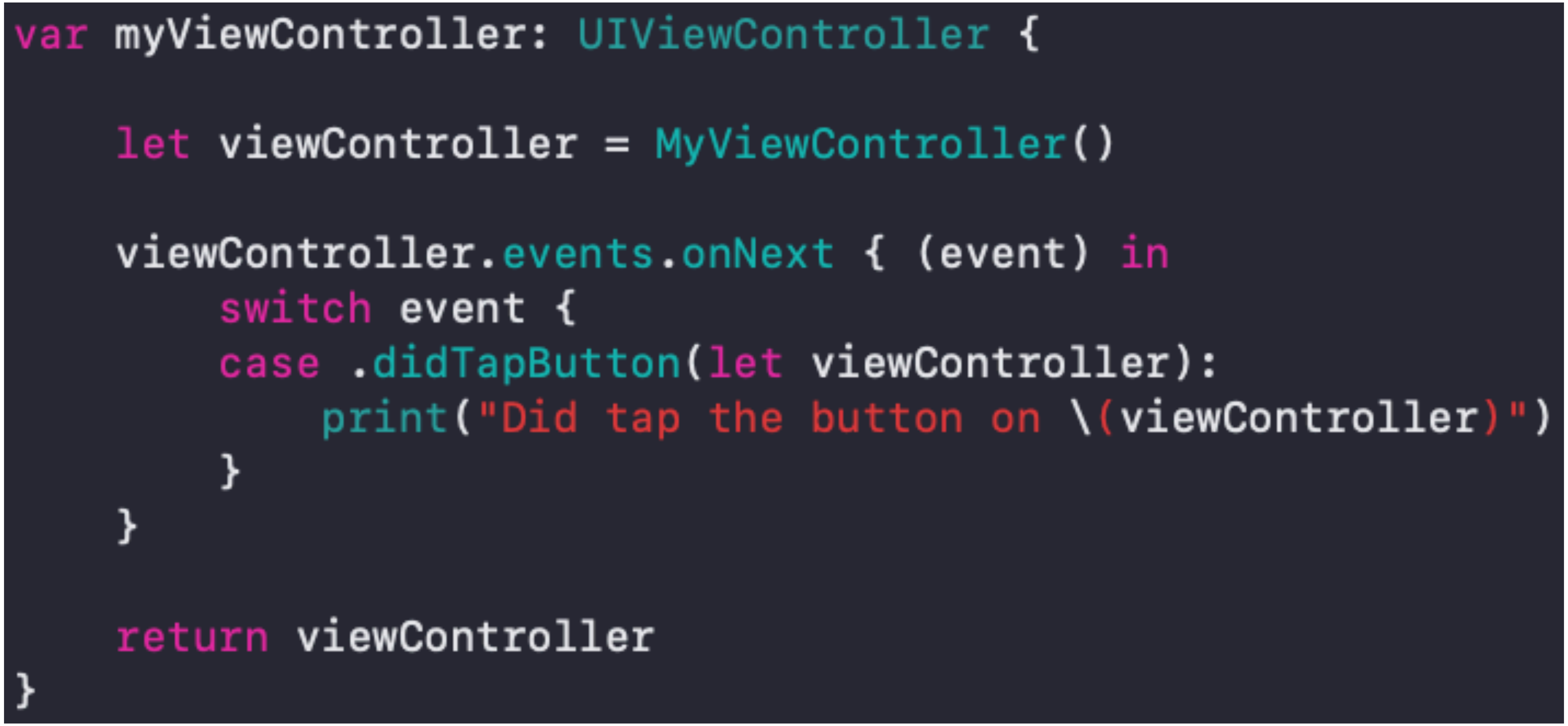


EVENTS



```
class MyViewController: UIViewController {
    enum Events: EventType {
        case didTapButton(viewController: UIViewController)
    }

    let events = EventEmitter<Events>()
    var disposeBag = DisposeBag()
    let button = UIButton()

    func setupBindings() {
        button.rx.tap
            .withUnretained(self)
            .map(Events.didTapButton)
            .bind(to: events.emitter)
            .disposed(by: disposeBag)
    }
}
```

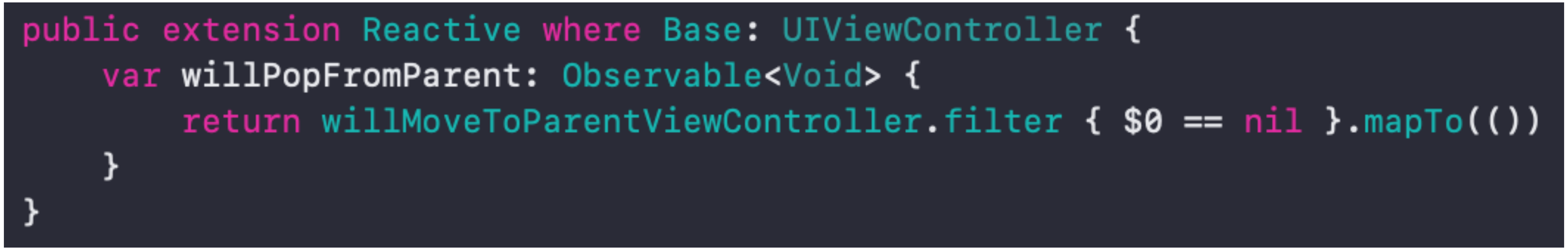
EVENTS



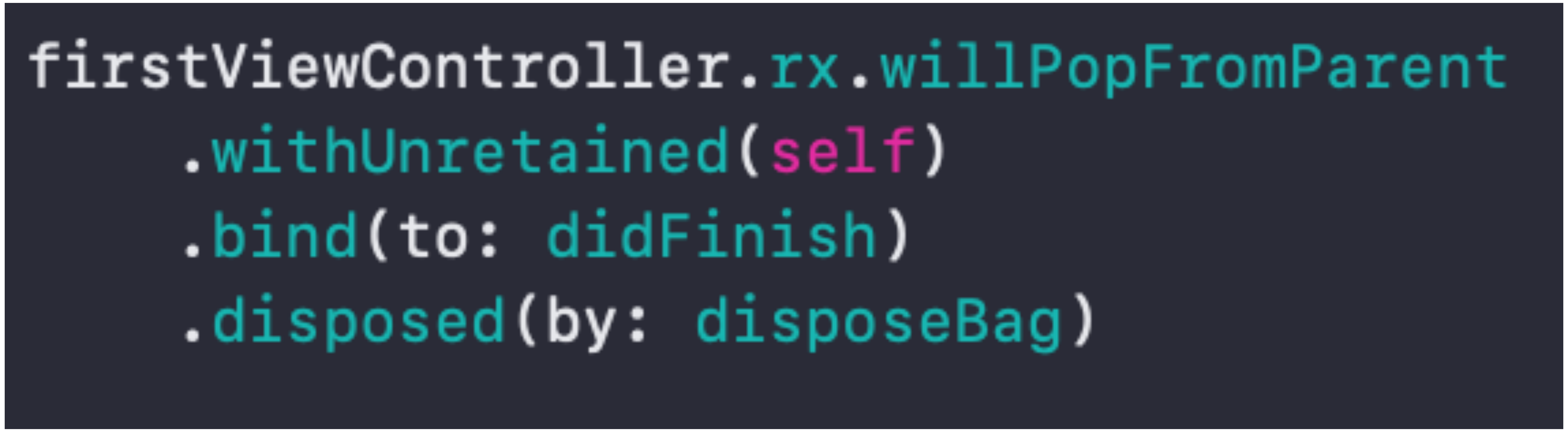
```swift
var myViewController: UIViewController {

    let viewController = MyViewController()

    viewController.events.onNext { (event) in
        switch event {
        case .didTapButton(let viewController):
            print("Did tap the button on \(viewController)")
        }
    }

    return viewController
}
```

HANDLING BACK BUTTON ON HORIZONTAL FLOW

- If not handled, the flow will never emit a finishing event
- Easy to handle with RxSwift



```swift
public extension Reactive where Base: UIViewController {
    var willPopFromParent: Observable<Void> {
        return willMoveToParentViewController.filter { $0 == nil }.mapTo(())
    }
}
```



```swift
firstViewController.rx.willPopFromParent
    .withUnretained(self)
    .bind(to: didFinish)
    .disposed(by: disposeBag)
```

THAT'S IT

Alexandre Tavares

alexandre.tavares@strv.com

STRV

QUESTIONS?

Alexandre Tavares

alexandre.tavares@strv.com

compiled.social/AlTavares

STRV

STRV