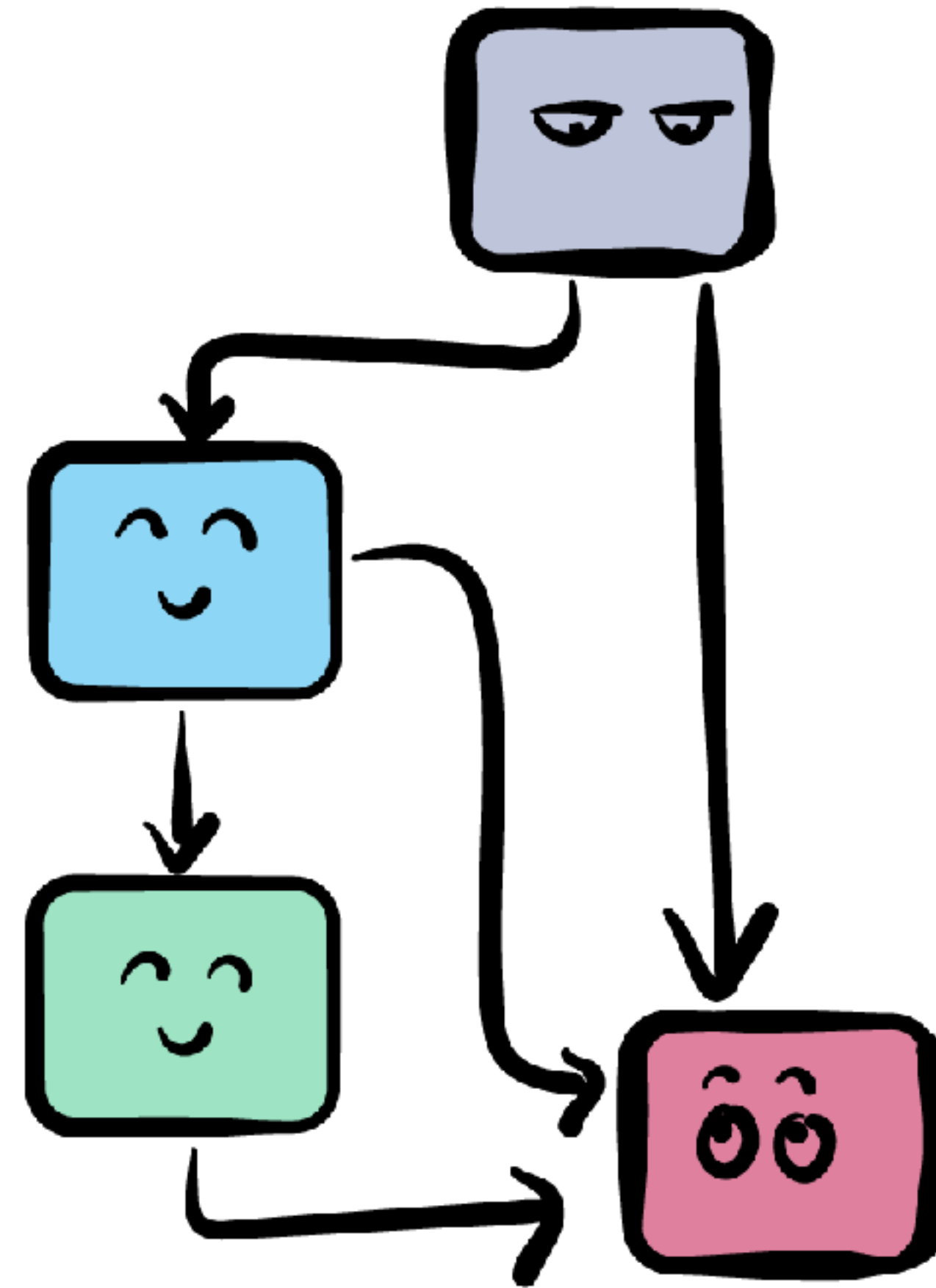


# >>> iOS <<< CONCURRENCY ..... WITH GCD & OPERATIONS



PART 5: OPERATION



# TILT-SHIFT OPERATION

```
30 let inputImage = UIImage(named: "dark_road_small.jpg")
```



31

```
51 tsOp.outputImage
```



52

# OPERATION

---

```
let operation1 = {  
    print("Operation 1 started")  
    print("Operation 1 finished")  
}
```

```
let operation2 = BlockOperation {  
    print("Operation 2 started")  
    print("Operation 2 finished")  
}
```

```
class ImageTransformOperation: Operation {  
    var inputImage: UIImage?  
    var outputImage: UIImage?  
  
    override func main() {  
        outputImage = transform(image: inputImage)  
    }  
}
```



# OPERATION STATES

---

```
class TiltShiftOperation : NSOperation {  
    var inputImage: UIImage?  
    var outputImage: UIImage?  
  
    override func main() {  
        guard let inputImage = inputImage else { return }  
        let mask = topAndBottomGradient(inputImage.size)  
        outputImage = inputImage.applyBlurWithRadius(4, maskImage: mask)  
    }  
}
```

Operation



# OPERATION STATES

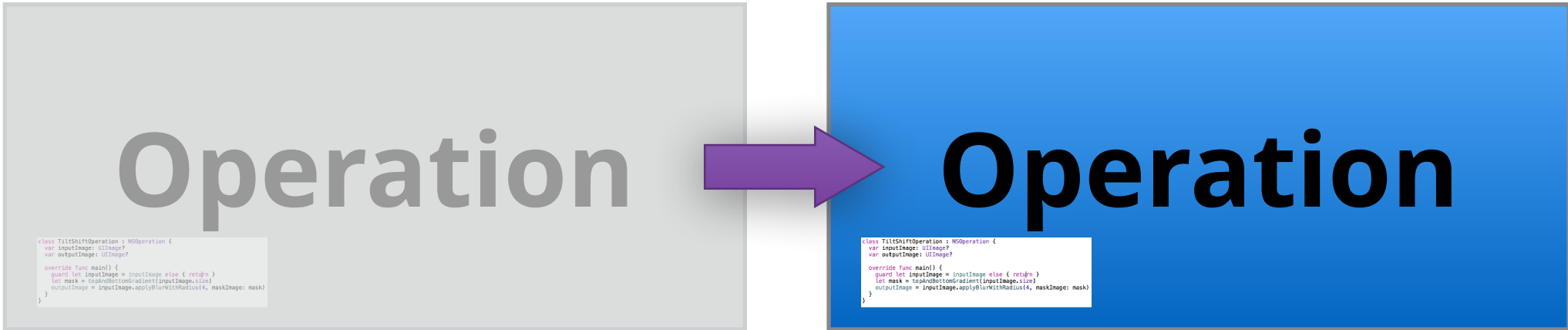
# Operation

```
class TiltShiftOperation : NGOperation {
    var inputImage: UIImage?
    var outputImage: UIImage?

    override fun make() {
        guard let inputImage = inputImage else { return }
        let mask = Image.fromColorGradient(inputImage.size)
        outputImage = inputImage.applyBlurWithRadius(4, maskImage: mask)
    }
}
```

# OPERATION STATES

isReady



# OPERATION STATES

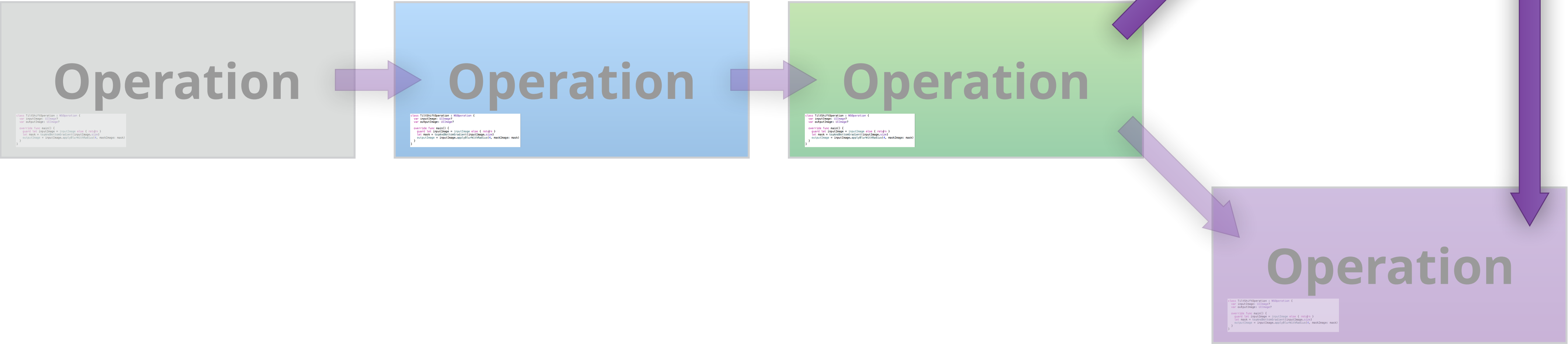


# OPERATION STATES

isCancelled

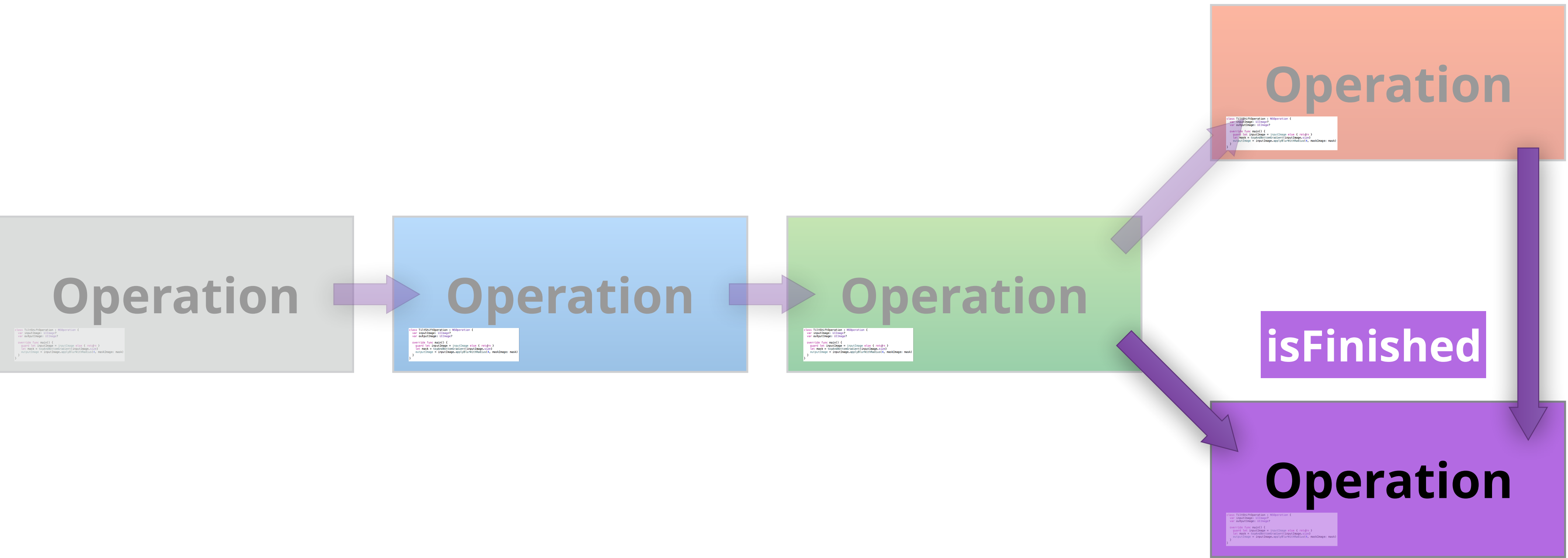
Operation

Operation





# OPERATION STATES



# OPERATION CLASS

---

```
open class Operation : NSObject {  
    open func start()  
    open func main()  
    open var isCancelled: Bool { get }  
    open func cancel()  
  
    open var isExecuting: Bool { get }  
    open var isFinished: Bool { get }  
    open var isAsynchronous: Bool { get }  
    open var isReady: Bool { get }  
  
    open var completionBlock: (() -> Swift.Void)?  
    open var qualityOfService: QualityOfService  
    open var name: String?  
    ...  
}
```

# BLOCKOPERATION CLASS

---

```
open class BlockOperation : Operation {  
    public convenience init(block: @escaping () -> Swift.Void)  
    open func addExecutionBlock(_ block: @escaping () -> Swift.Void)  
    open var executionBlocks: [() -> Swift.Void] { get }  
}
```



# CHALLENGE TIME!

---

