Truths Universally Acknowledged:

Swift Design Patterns as Jane Austen Heroes

northofnormal

How to tell you are in a Jane Austen Novel

- Someone disagreeable is trying to persuade you to take a trip to Bath

How to tell you are in a Jane Austen Novel

- A woman who hates you is playing the pianoforte

Code Crush

Marketing made me do it.

Code Crush: Guard Statements FTW

By Anne Cahalan

When talking about clean code principles, one of my mentors told me that my code should read like sentences. Name things clearly and concisely, separate concerns logically, be terse but readable.

“So,” I said, “like *Hemingway* sentences? Not *Faulkner* sentences?”

How to tell you are in a Jane Austen Novel

- You once took a walk with a cad

Jane Austen

Delegates & Protocols

Mr Darcy

Pride and Prejudice

Pride And Prejudice

- Mr. Fitzwilliam Darcy arrives at Netherfield Hall, immediately is horrified by the neighbors and makes a terrible impression on Ms. Elizabeth Bennet
- Shortly thereafter, army officer Mr. Wickham arrives in town and makes a much better impression on the local ladies.
- A series of events, Darcy eventually warms to Lizzie, while she’s seeing the appeal of Mr. Wickham.
- So Darcy rushes to her side to tell her that, despite her appalling family and the fact that she is so terribly beneath him, he has, despite all his better judgment, fallen in love with her.

Pride And Prejudice

- Meanwhile, Wickham disappears with Lizzie’s younger sister—without marrying her.

- Everyone panics, Mr. Bennet goes looking for his lost daughter but returns without her—only to find that somehow, everything has worked out for the best.

- Wickham has married Lydia in return for an annual income…paid for by Darcy!

- Darcy returns, he and Lizzie take a walk in a garden and he proposes.

Protocols And Delegates

“A protocol defines a blueprint of methods, properties, and other requirements that suit a particular task or piece of functionality. The protocol can then be adopted by a class, structure, or enumeration to provide an actual implementation of those requirements. Any type that satisfies the requirements of a protocol is said to conform to that protocol.”

```swift
public protocol WooingDelegate {
    func wooingAttempt()
}

struct gentleman: WooingDelegate {
    func wooingAttempt() {
        print("You must allow me to tell you how ardently I
        admire and love you.")
    }
}

struct scoundrel: WooingDelegate {
    func wooingAttempt() {
        print("I know it's not funny, but your perfume smells
        like your daddy's got money.")
    }
}
```

```
class suitorTableViewCell: UITableViewCell {
   // some cell set up

    var wooingDelegate: WooingDelegate

    @IBAction func wooingButtonTapped(sender: UIButton) {
        wooingDelegate.wooingAttempt()
    }
}
```

```
extension SuitorListViewController: UITableViewDelegate, UITableViewDataSource {

    func numberOfSections(in tableView: UITableView) -> Int {
        return 2
    }

    func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
        if section == 0 {
            return gentlemen.count
        } else if section == 1 {
            return scoundrels.count
        }
    }

    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) ->
     UITableViewCell {
        let cell = tableView.dequeueReusableCell(withIdentifier: "SuitorListTableViewCell")
             as! SuitorListTableViewCell
        let suitor = suitorList[indexPath.row]
        cell.setupCell(with: suitor)

        return cell
    }

}
```

```
extension WooingDelegate {

    func sendLoveLetter() {
        print("You pierce my soul. I am half agony, half hope. Tell me
not that I am too late, that such precious feelings are gone for ever.
I offer myself to you again with a heart even more your own than when
you almost broke it, eight years and a half ago. Dare not say that man
forgets sooner than woman, that his love has an earlier death. I have
loved none but you. Unjust I may have been, weak and resentful I have
been, but never inconstant.")
    }

}
```

```swift
class suitorTableViewCell: UITableViewCell {
   // some cell set up

    var wooingDelegate: WooingDelegate?

    @IBAction func wooingButtonTapped(sender: UIButton) {
        if let wooingDelegate = wooingDelegate {
            wooingDelegate.wooingAttempt()
        }
    }

    @IBAction func sendLoveLetterButtonTapped() {
        wooingDelegate.sendLoveLetter()
    }
}
```

How are Protocols & Delegates like Mr. Darcy?

- They seem smug

How are Protocols & Delegates like Mr. Darcy?

- They seem smug
- They can be inscrutable

How are Protocols & Delegates like Mr. Darcy?

- They seem smug
- They can be inscrutable
- They will save the day, behind the scenes

How to tell you are in a Jane Austen Novel

- You develop a resentment at a public dance

Enums

Capt. Wentworth

Persuasion

Persuasion

- 7 years ago, Anne Elliot broke an engagement to a young naval officer, Frederick Wentworth.

- He was confident and ambitious, but not of a particularly noteworthy family—her father and sisters convinced her that she would be marrying beneath her station. The Elliot family is old, titled, landed.

- But time passes, and Freddie Wentworth becomes the decorated Capt. Wentworth, but as for Anne, “her bloom had vanished early…she was faded and thin.”

- And the proud Elliot family has fallen into debt and been reduced to renting out their manor to another naval officer

Persuasion

- One day, on a walk with Capt. Wentworth and some friends, one of the women falls and is terribly injured. Everyone freaks out, Anne is calm and level-headed

- Wentworth starts to reconsider his resentment of Anne, but he’s still pretty upset over the whole getting dumped thing

- In a drawing room, Anne has a conversation with a gentlemen over wether women or men fall out of love quicker

- Capt. Wentworth overhears and hides a letter to Anne declaring his love.

```swift
func setupCell(for setting: String, suitor: String) {
    let cell = SettingCell()

    if setting == "Netherfield Hall" {
        cell.backgroundColor = .blue
    } else if setting == "Cleveland Park" {
        cell.backgroundColor = .yellow
    } else if setting == "Kellynch Hall" {
        cell.backgroundColor = .green
    }

    if suitor == "Darcy" {
        suitorLabel.text = "Insufferable, but maybe not?"
    } else if suitor == "Wickham" {
        suitorLabel.text = "Immoral con-man"
    } else if suitor == "Willoughby" {
        //you know what, this is exhausting logic and I got tired
    }
}
```

```swift
func cellSetup(for setting: String, suitor: String) {
    let cell = SettingCell()

    switch setting {
    case "Netherfield Hall":
        cell.backgroundColor = .blue
    case "Cleveland Park":
        cell.backgroundColor = .yellow
    case "Kellynch Hall":
        cell.backgroundColor = .green
    }

    switch suitor {
    case "Darcy":
        suitorLabel.text = "Insufferable, but maybe not?"
    default:
        // you know what, I'm still tired
    }
}
```

What do we know?

- We know we have a limited number of settings
- We want to associate a background color with each setting
- Each setting has certain associated suitors

```swift
enum Setting {
    case netherfieldHall
    case clevelandPark
    case kellynchHall

    var backgroundColor: UIColor {
        switch self {
        case .netherfieldHall:
            return .blue
        case .clevelandPark:
            return .yellow
        case .kellynchHall:
            return .green
        }
    }
}
```

```
enum Setting {
    …
    var suitors: [Suitor] {
        switch self {
        case .netherfieldHall:
            return [.wickham, .darcy]
        case .clevelandPark:
            return [.willoughby, .brandon]
        case .kellynchHall:
            return [.wentworth]
        }
    }
}
```

```
enum Suitor {
    case wickham
    case darcy
    case willoughby
    case brandon
    case wentworth


    var name:String {
        switch self {
        case .wickham:
           //…etc
    }

    var description: String {
        switch self {
        case .wickham:
            // …etc
}
```

```
func doSomeSettingUIBusiness() {

    backgroundColor = setting.backgroundColor

    for suitor in setting.suitors {
        print("\(suitor.name): \(suitor.description)")
        // maybe do something useful in the UI, like set some
        labels
    }
}
```

How do we get from enums to dreamy sea captains?

- If blocks are expensive, spendthrift with your attention — like Anne’s father, blowing through the family fortune

How do we get from enums to dreamy sea captains?

- If blocks are expensive, spendthrift with your attention — like Anne’s father, blowing through the family fortune

- Enums & Switches in Swift have more opportunity, now — like Capt. Wentworth prospering from the Napoleonic Wars

How do we get from enums to dreamy sea captains?

- If blocks are expensive, spendthrift with your attention — like Anne’s father, blowing through the family fortune

- Enums & Switches in Swift have more opportunity, now — like Capt. Wentworth prospering from the Napoleonic Wars

- New times and new languages require new ways of thinking

How to tell you are in a Jane Austen Novel

- You have five hundred a year. From who? Five hundred what? No one knows. No one cares. You have it. It’s yours. All five hundred of it.

Guard Statements

Col. Brandon

Sense and Sensibility

Sense And Sensibility

- Marianne Dashwood meets the charming Mr. Willoughby when she sprains her ankle on a walk and he gallantly carries her home

- The two are inseparable and deeply in love, but suddenly Mr. Willoughby departs for London and stops answering Marianne’s letters

- Meanwhile, Col. Brandon shows up and is obviously taken with Marianne. She’s not interested.

Sense And Sensibility

- Marianne is invited to London, where she hopes to meet Mr. Willoughby again

- And she does, at dance where he is incredibly rude to her—and announces his engagement to a wealthy heiress

- Distraught, Marianne falls ill and Col. Brandon helps care for her, reading her sonnets

- And Col. Brandon and Marianne get married

Guard Statements

“A guard statement is used to transfer program control out of a scope if one or more conditions aren’t met”

```swift
func createSuitor() {
    if let name = caller.name{
        if let hatType = caller.hat() {
            if let incomeLevel = caller.income {
                if let horseName = caller.horse() {
                    sendToServer(name, hat: hatType, income:
                    incomeLevel, horse: horseName)
                } else {
                    print("No horse")
                }
            } else {
                print("No income")
            }
        } else {
            print("No hat")
        }
    } else {
        print("No name")
    }
}
```

```swift
func createSuitor() {
    guard let name = caller.name else {
        print("No name")
        return
    }

    guard let hasHat = caller.hasHat else {
        print("No hat")
        return
    }

    guard let income = caller.income else {
        print("No income")
        return
    }

    guard let hasHorse = caller.hasHorse else {
        print("No horse")
        return
    }

    sendToServer(name, hasHat: hasHat, income:income, hasHorse: hasHorse)
}
```

How is a guard statement like Col. Brandon?

- Was Marianne settling for the boring guy—or finding romance where she wasn’t looking for it?

How is a guard statement like Col. Brandon?

- Was Marianne settling for the boring guy—or finding romance where she wasn’t looking for it?

- Elegant in both the scientific and aesthetic sense

How is a guard statement like Col. Brandon?

- Was Marianne settling for the boring guy—or finding romance where she wasn’t looking for it?

- Elegant in both the scientific and aesthetic sense

- truth vs. Truth

If you can find capital-T-truth in your code…

…that means you can put capital-T-truth in your code.

What are the Truths in your code?

How to tell you are in a Jane Austen Novel

- You are in a garden, and you are astonished northofnormal