



ChatGPT в мобильной разработке

Екатерина Батеева





Что делает разработчик изО дня в день?



«Красит кнопки»



Что делает разработчик изО дня в день?



«Красит кнопки»



«Парсит JSON»



Что делает разработчик изо дня в день?



«Красит кнопки»



«Парсит JSON»



Конвертирует код



Что делает разработчик изО дня в день?



«Красит кнопки»



«Парсит JSON»



Конвертирует код



Повышает
качество кода



Что делает разработчик изО дня в день?



«Красит кнопки»



«Парсит JSON»



Конвертирует код



Повышает
качество кода



Пишет скрипты



Что делает разработчик изО дня в день?



«Красит кнопки»



«Парсит JSON»



Конвертирует код



Повышает
качество кода



Пишет скрипты



Пишет Unit test'ы



Красим
кнопки





Could you generate UIButton with
system icon "minus.circle",
background Color white,
cornerRadius 20,
tint Color red



less

 Copy code

```
let button = UIButton(type: .system)
let image = UIImage(systemName: "minus.circle")?.withImageTintColor(.white)
button.setImage(image, for: .normal)
button.backgroundColor = .white
button.layer.cornerRadius = 20
```



```
let button = UIButton(type: .system)
let image = UIImage(systemName: «minus.circle")?
    .withRenderingMode(.alwaysTemplate)
button.setImage(image, for: .normal)
button.backgroundColor = .white
button.layer.cornerRadius = 20
```



Ожидание



Реальность



Попробуем что-то
посложнее





Could you, please create
UICollectionViewCell for
counting player's level?



```
import UIKit
```

```
class LevelCell: UICollectionViewCell {
```

```
    // MARK: Properties
```

```
    private let levelLabel = UILabel()
```

```
    // MARK: Initialization
```



```
override init(frame: CGRect) {  
    super.init(frame: frame)  
  
    //Configure the cell  
    backgroundColor = .white  
    layer.cornerRadius = 10  
    layer.borderWidth = 1  
    layer.borderColor = UIColor.gray.cgColor
```



```
//Configure the level label
levelLabel.textAlignment = .center
levelLabel.font = UIFont.systemFont(ofSize: 20, weight: .semibold)
levelLabel.translatesAutoresizingMaskIntoConstraints = false
contentView.addSubview(levelLabel)
NSLayoutConstraint.activate([
    levelLabel.centerXAnchor.constraint(equalTo: contentView.centerXAnchor),
    levelLabel.centerYAnchor.constraint(equalTo: contentView.centerYAnchor),
    levelLabel.widthAnchor.constraint(equalTo: contentView.widthAnchor)
])
}
```



```
required init?(coder: NSCoder) {  
    fatalError("init(coder:) has not been implemented")  
}  
  
func configure(with level: Int) {  
    levelLabel.text = "Level \ \(level)"  
}
```



swift

Copy code

```
collectionView.register(LevelCell.self, forCellWithReuseIdentifier: "LevelCell")

func collectionView(_ collectionView: UICollectionView, cellForItemAt indexPath: IndexPath)
    let cell = collectionView.dequeueReusableCell(withReuseIdentifier: "LevelCell", for: indexPath)
    cell.configure(with: playerLevels[indexPath.item])
    return cell
}
```



Level 0



Could you, please create UICollectionViewCell with height = 150, width = 150, cornerRadius 10, backgroundColor #FFE2AA, borderColor black, borderWidth 2, label with player name, plus and minus buttons to increase level?



▼ 🔥 PlayerCell

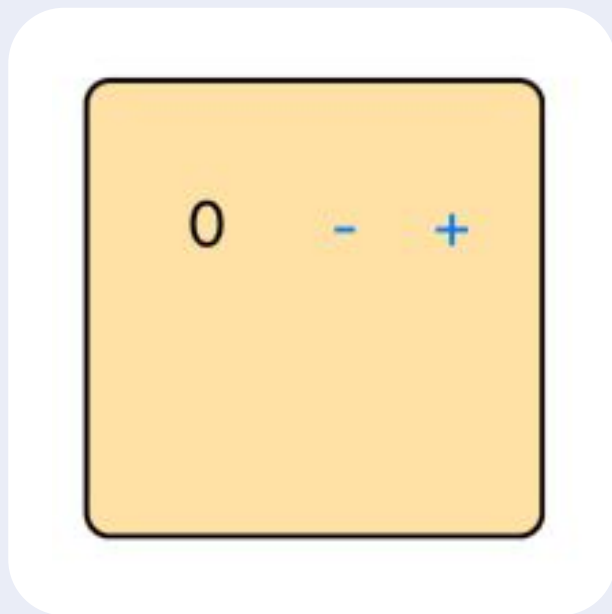
- Incorrect argument label in call (have 'hex:', expected 'named:')

▼ 🔥 ViewController

- ✗ Cannot assign value of type 'Int' to type 'String'
- ✗ Cannot convert value of type 'Int' to expected argument type 'Player'
- > ✗ Binary operator '==' cannot be applied to operands of type 'String' and 'Int'

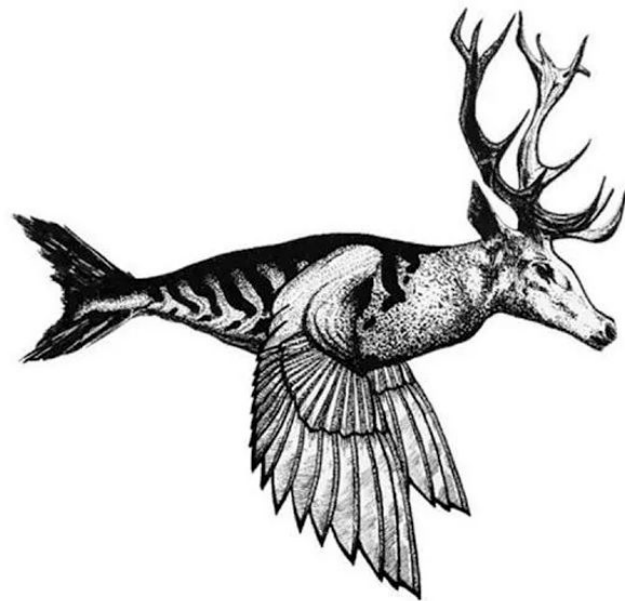


Исправим
ошибки...





Без внятного ТЗ —
результат хз





Минусы

01

**Нужно максимально точное
и подробное описание**



Минусы

01

**Нужно максимально точное
и подробное описание**

02

На простых объектах много ошибок



Минусы

01

**Нужно максимально точное
и подробное описание**

02

На простых объектах много ошибок

03

Многое придется переделать/дописать



Минусы

01

**Нужно максимально точное
и подробное описание**

02

На простых объектах много ошибок

03

Многое придется переделать/дописать

04

Код выглядит топорно, проблемы с DRY



Плюсы

01

Сгенерил много базового кода



Плюсы

01

Сгенерил много базового кода

02

Стабильная верстка с autolayout



Плюсы

01

Сгенерил много базового кода

02

Стабильная верстка с autolayout

03

Код легко читается, есть комментарии



Парсим JSON



```
{ "widget": {
  "debug": "on",
  "window": {
    "title": "Sample Konfabulator Widget",
    "name": "main_window",
    "width": 500,
    "height": 500
  },
  "image": {
    "src": "Images/Sun.png",
    "name": "sun1",
    "hOffset": 250,
    "vOffset": 250,
    "alignment": "center"
  },
  "text": {
    "data": "Click Here",
    "size": 36,
    "style": "bold",
    "name": "text1",
    "hOffset": 250,
    "vOffset": 100,
    "alignment": "center",
    "onMouseUp": "sun1.opacity = (sun1.opacity / 100) * 90;"
  }
}
```



```
struct Widget: Codable {  
    let debug: String  
    let window: Window  
    let image: Image  
    let text: Text  
}
```



```
struct Window: Codable {  
    let title: String  
    let name: String  
    let width: Int  
    let height: Int  
}
```

```
struct Image: Codable {  
    let src: String  
    let name: String  
    let hOffset: Int  
    let vOffset: Int  
    let alignment: String  
}
```

```
struct Text: Codable {  
    let data: String  
    let size: Int  
    let style: String  
    let name: String  
    let hOffset: Int  
    let vOffset: Int  
    let alignment: String  
    let onMouseUp: String  
}
```





```
{
  "menu": {
    "id": "file",
    "value": "File",
    "popup": {
      "menuitem": [
        {"value": "New", "onclick": "CreateNewDoc()"},
        {"value": "Open", "onclick": "OpenDoc()"},
        {"value": "Close", "onclick": "CloseDoc()"}
      ]
    }
  }
}
```




```
struct Menu: Codable {  
  let id: String  
  let value: String  
  let popup: Popup  
}
```

```
struct Popup: Codable {  
  let menuItem: [MenuItem]  
}
```

```
struct MenuItem: Codable {  
  let value: String  
  let onclick: String  
}
```



```
let jsonData = jsonString.data(using: .utf8)!  
let decoder = JSONDecoder()  
let menu = try decoder.decode(Menu.self, from:  
jsonData)
```

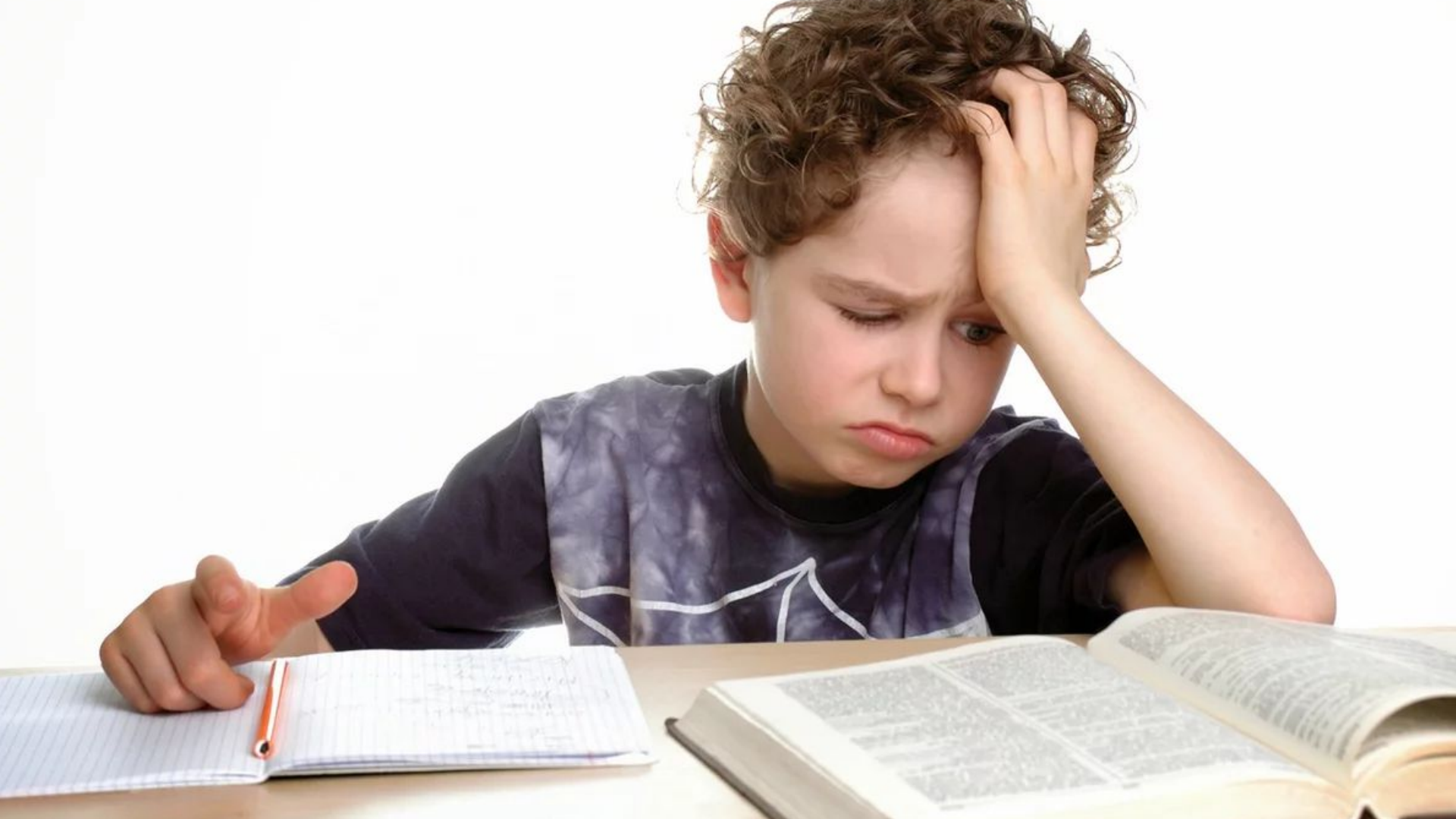




Конвертируем код



Не хочу учить Objective-C



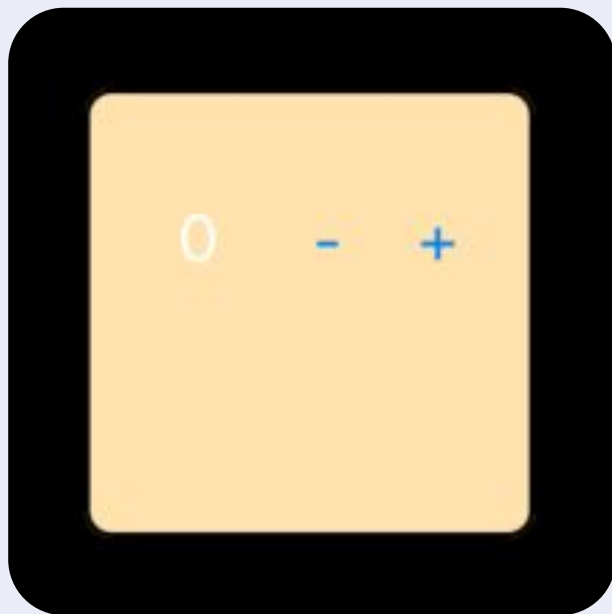


```
// Assumes input like "#00FF00" (#RRGGBB).
+ (UIColor *)colorFromHexString:(NSString *)hexString {
    unsigned rgbValue = 0;
    NSScanner *scanner = [NSScanner scannerWithString:hexString];
    [scanner setScanLocation:1]; // bypass '#' character
    [scanner scanHexInt:&rgbValue];
    return [UIColor colorWithRed:((rgbValue & 0xFF0000) >> 16)/255.0
    green:((rgbValue & 0xFF00) >> 8)/255.0
    blue:(rgbValue & 0xFF)/255.0 alpha:1.0];
}
```



OK

```
static func colorFromHexString(_ hexString: String) -> UIColor? {  
    var rgbValue: UInt32 = 0  
    let scanner = Scanner(string: hexString)  
    scanner.scanLocation = 1 // bypass '#' character  
    scanner.scanHexInt32(&rgbValue)  
  
    let red = CGFloat((rgbValue & 0xFF0000) >> 16) / 255.0  
    let green = CGFloat((rgbValue & 0x00FF00) >> 8) / 255.0  
    let blue = CGFloat(rgbValue & 0x0000FF) / 255.0  
  
    return UIColor(red: red, green: green, blue: blue, alpha: 1.0)  
}
```



```
scanner.scanLocation = 1 // bypass '#' character  
scanner.scanHexInt32(&rgbValue)
```



'scanLocation' was deprecated in iOS 13.0



'scanHexInt32' was deprecated in iOS 13.0



```
#import "UIColor+HexString.h"
```

```
@interface UIColor(HexString)
```

```
+ (UIColor *) colorWithHexString: (NSString *) hexString;
```

```
+ (CGFloat) colorComponentFrom: (NSString *) string start:  
(NSInteger) start length: (NSInteger) length;
```

```
@end
```



```
@implementation UIColor(HexString)

+ (UIColor *) colorWithHexString: (NSString *) hexString {
    NSString *colorString = [[hexString
    stringByReplacingOccurrencesOfString: @"#" withString: @""]
    uppercaseString];
    CGFloat alpha, red, blue, green;
    switch ([colorString length]) {
        case 3: ... // #RGB
        case 4: ... // #ARGB
        case 6: ... // #RRGGBB
        case 8: ... // #AARRGGBB
        default: ...
    }
}
```



```
case 3: // #RGB
    alpha = 1.0f;
    red = [self colorComponentFrom: colorString start: 0 length: 1];
    green = [self colorComponentFrom: colorString start: 1 length: 1];
    blue = [self colorComponentFrom: colorString start: 2 length: 1];
    break;
```

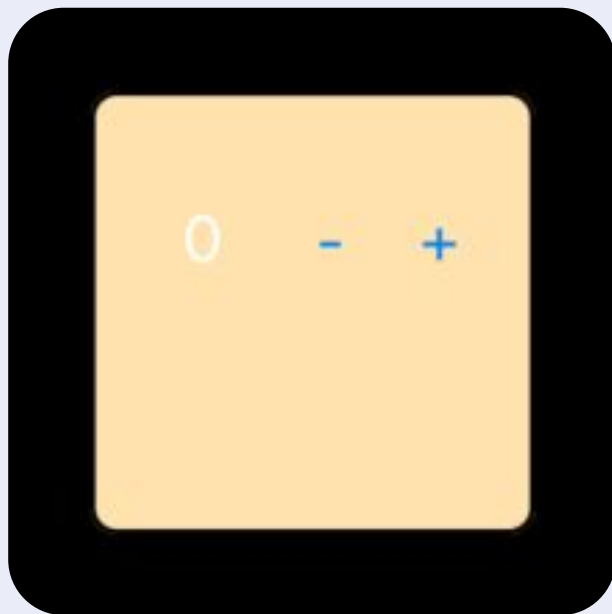


```
extension UIColor {
    static func color(fromHexString hexString: String) -> UIColor {
        var colorString =
            hexString.trimmingCharacters(in: .whitespacesAndNewlines)
                .uppercased()
        if colorString.hasPrefix("#") {
            colorString.remove(at: colorString.startIndex)
        }
        guard colorString.count == 6 || colorString.count == 8 else {
            fatalError("Invalid color value \(hexString). #RGB, #ARGB, #RRGGBB, or #AARRGGBB.")
        }
    }
}
```



```
var alpha: CGFloat = 1.0
var red: CGFloat = 0.0
var blue: CGFloat = 0.0
var green: CGFloat = 0.0
var startIndex = colorString.startIndex
if colorString.count == 8 {
    let alphaIndex = colorString.index(startIndex, offsetBy: 2)
    let alphaSubstring = colorString[startIndex..
```

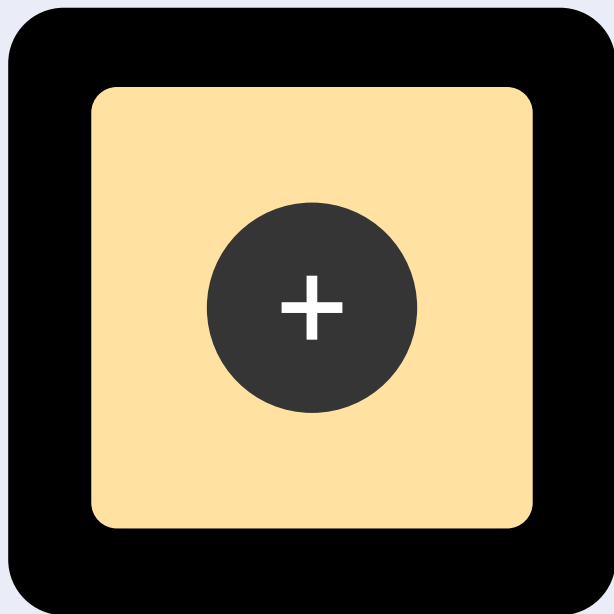
```
let redIndex = colorString.index(startIndex, offsetBy: 2)
let redSubstring = colorString[startIndex..
```





VS



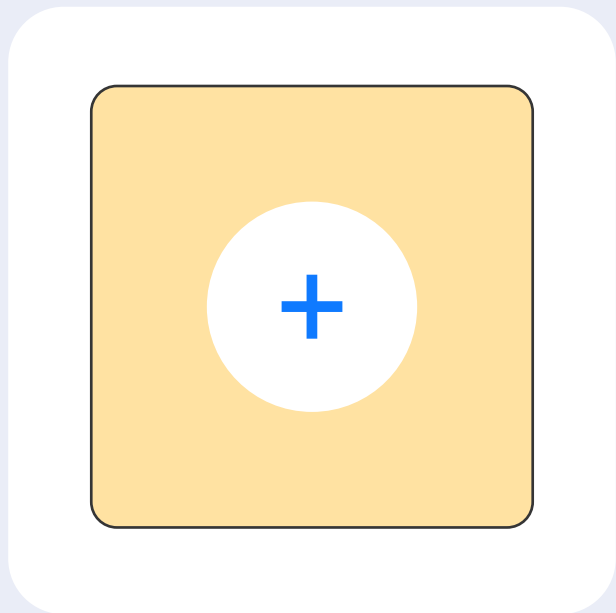




```
struct NewPlayerCell: View {
    weak var delegate: NewPlayerCellDelegate?

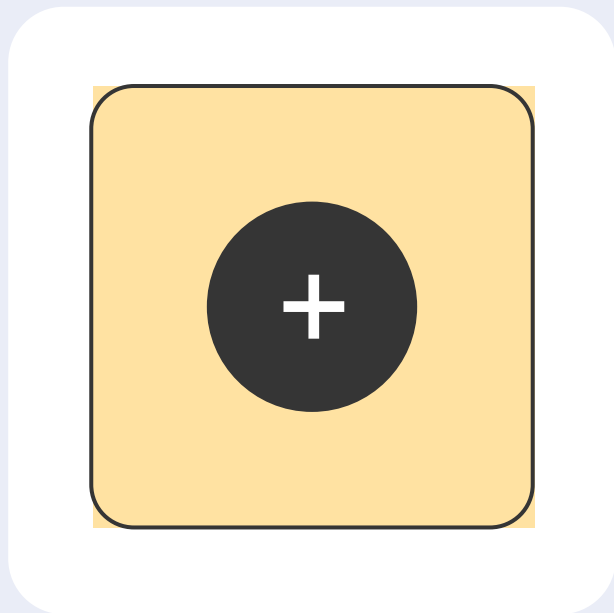
    var body: some View {
        ZStack {
            RoundedRectangle(cornerRadius: 25)
                .stroke(Color.mainLineColor, lineWidth: 2)
                .background(Color.mainBackgroundColor)
                .cornerRadius(25)

            Button(action: {
                delegate?.addPlayer()
            }, label: {
                Text("+")
                    .font(.largeTitle)
                    .frame(width: 60, height: 60)
                    .background(Color.white)
                    .cornerRadius(30)
            })
        }
        .padding(16)
    }
}
```





```
struct ActionButton: View {  
    var body: some View {  
        Button(action: {}, label: {  
            Text("+")  
                .font(.system(size: 50))  
                .foregroundColor(.white)  
                .frame(width: 60, height: 60)  
                .background(Color.mainLineColor)  
  
            .clipShape(RoundedRectangle(cornerRadius: 11))  
        })  
        .contentShape(Rectangle())  
        .padding(16)  
    }  
}
```







Storyboard

конвертируем сами



Повышаем
качество кода



Заменим deprecated



```
scanner.scanHexInt32(&rgbValue)
```



'scanHexInt32' was deprecated in iOS 13.0

```
scanner.scanHexInt64(&rgbValue)
```



Сколько не просила
переделать —
Появлялись warning'и



Оптимизирует верстку



The `layoutViews()` function seems to be laying out various views on the screen. I can offer some suggestions to optimize this function:



1. Reuse views wherever possible: If there are any views that are being reused across multiple parts of the screen, you could consider creating them once and reusing them instead of creating new instances every time.
2. Simplify constraints: If possible, simplify the constraints being applied to views. For example, you could consider using `leading` and `trailing` constraints instead of setting fixed widths for views.
3. Batch views for improved performance: If there are many views being laid out, consider grouping them together and then adding them to the parent view all at once, instead of adding them one by one.
4. Use `lazy` initialization: If any of the views being created are not always needed, you could consider using lazy initialization to defer their creation until they are actually needed.
5. Use a background thread: If the layout process is taking a long time and blocking the UI thread, you could consider performing the layout on a background thread to improve performance.

Overall, without more context about the specific use case and requirements of this function, it's difficult to provide more specific optimization suggestions.



Советы

01

Переиспользуй view

02

**Используй ширину вместо
leading и trailing**

03

Группируй view

04

Используй lazy инициализацию

05

Используй background потоки



Советы общие — не ищет
конкретные ошибки



Пишем скрипты



А если запустить тесты
и перезапустить упавшие?



```
lane :test do
  begin
    # run tests
    scan(scheme: "MyAppTests")
  rescue
    # get failed tests and run them again
    failed_tests =
      scan_last_result[:testsFailedCount].to_i
    if failed_tests > 0
      puts "Re-running #{failed_tests} failed
tests..."
      scan(scheme: "MyAppTests",
        only_testing: scan_last_result[:failedTests])
    else
      puts "All tests passed!"
    end
  end
end
```



Хочу собрать коммиты
за сегодня



```
#!/bin/bash
```

```
# Get the start and end of today as Unix timestamps
```

```
start_of_day=$(date -j -f "%Y-%m-%d" "$(date +%Y-%m-%d) 00:00:00" "+%s")
```

```
end_of_day=$(date -j -f "%Y-%m-%d" "$(date +%Y-%m-%d) 23:59:59" "+%s")
```

```
# Get the current branch name
```

```
branch=$(git rev-parse --abbrev-ref HEAD)
```

```
# Get all commits made to the current branch today
```

```
git log --pretty=format:"%h %s" --abbrev-commit --since="$start_of_day" --until="$end_of_day"  
"$branch"
```



Хочу собрать коммиты
с задачами текущей версии



```
lane :get_ios_commits do
  # The name of the Git remote and branch to fetch from
  remote_name = "origin"
  branch_name = "main"

  # The tag names for the range of commits to fetch
  from_tag = "9.71.1"
  to_tag = "HEAD"

  # Fetch the latest changes from the remote
  sh "git fetch #{remote_name}"

  # Get the commit hashes for the range of tags
  from_commit = sh("git rev-parse #{from_tag}", log: false).strip
  to_commit = sh("git rev-parse #{to_tag}", log: false).strip

  # Get the commit messages for the range of commits
  commits = sh("git log --pretty=%H%x09%s --grep='IOS-' #{from_commit}..#{to_commit}", log:
false).strip.split("\n")

  # Print the list of commits
  puts "Found #{commits.length} IOS-related commits between #{from_tag} and #{to_tag}:"
  commits.each do |commit|
    commit_hash, commit_message = commit.split("\t")
    puts "#{commit_hash}: #{commit_message}"
  end
end
```



А если запретить
коммиты по правилу?



Here's an example pre-commit hook script that prevents a commit if the commit message does not start with "IOS-":



bash

Copy code

```
#!/bin/sh

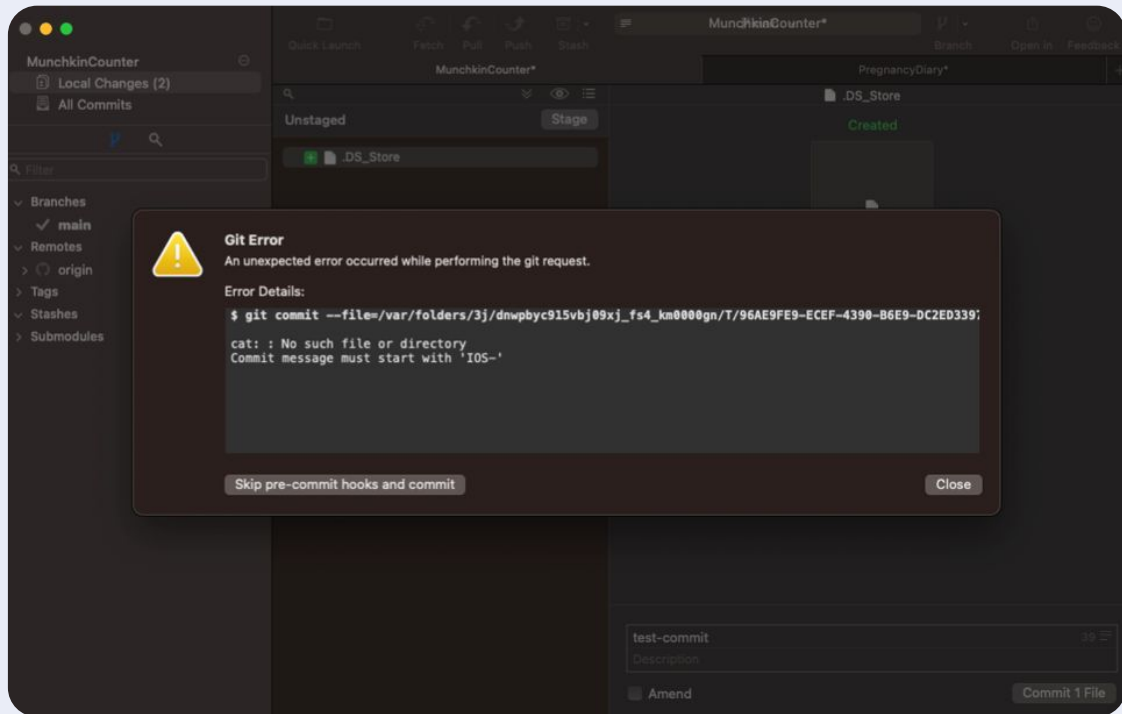
commit_msg_file=$1
commit_msg=$(cat "$commit_msg_file")

if ! echo "$commit_msg" | grep -q "^IOS-"; then
    echo "Commit message must start with 'IOS-'"
    exit 1
fi

exit 0
```

To use this script, create a file named "pre-commit" in the ".git/hooks" directory of your repository and paste the code above into it. Make sure to make the file executable by running `chmod +x .git/hooks/pre-commit`.

Now, when you try to commit with a message that does not start with "IOS-", you will receive an error message and the commit will not be allowed.







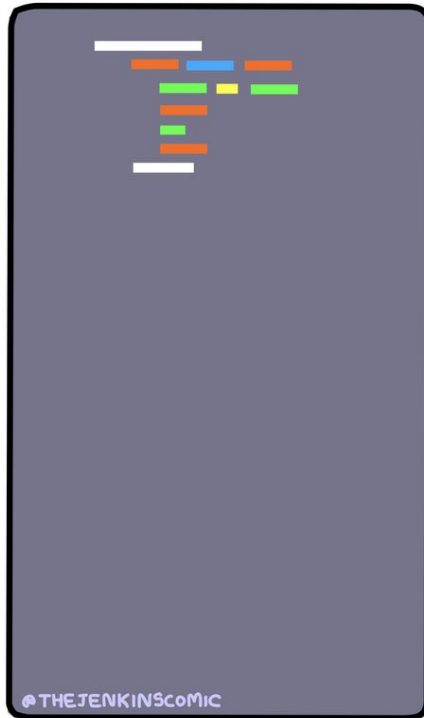
Пишем Unit-тесты



Код есть, а тестов нет



ACTUAL CODE



UNIT TESTS





```
class ViewController: UIViewController {  
    private let viewModel = ViewModel()  
  
    private lazy var collectionView = UICollectionView(...)  
    private lazy var layout: UICollectionViewFlowLayout = UICollectionViewFlowLayout.init()  
  
    private lazy var refreshButton: UIButton = {  
        let button = UIButton()  
        return button  
    }()  
}
```



OK

```
import XCTest
@testable import Munchkin

class MunchkinTests: XCTestCase {

    var viewController: ViewController!

    override func setUp() {
        super.setUp()
        viewController = ViewController()
        let _ = viewController.view // Load view hierarchy
    }

    func testCollectionView() {
        XCTAssertNotNil(viewController.collectionView)
        XCTAssertTrue(viewController.collectionView.delegate === viewController)
        XCTAssertTrue(viewController.collectionView.dataSource === viewController)
        XCTAssertTrue(viewController.collectionView.keyboardDismissMode == .onDrag)
        XCTAssertTrue(viewController.collectionView.collectionViewLayout is UICollectionViewFlowLayout)

    }

    func testRefreshButton() {
        XCTAssertNotNil(viewController.refreshButton)
        XCTAssertTrue(viewController.refreshButton.backgroundColor == .mainBackgroundColor)
        XCTAssertTrue(viewController.refreshButton.layer.borderColor == UIColor.label.cgColor)
        XCTAssertTrue(viewController.refreshButton.layer.borderWidth == 2)
        XCTAssertTrue(viewController.refreshButton.layer.cornerRadius == 10)
        XCTAssertTrue(viewController.refreshButton.titleLabel?.textColor == .mainLineColor)
        XCTAssertTrue(viewController.refreshButton.titleLabel?.text == NSLocalizedString("restartGame", comment: ""))
        XCTAssertTrue(viewController.refreshButton.actions(forTarget: viewController, forControlEvents:
            .touchUpInside)?.contains("reloadGame") ?? false)
    }
}
```




```
struct ContentView: View {  
    @ObservedObject private var model = Coordinator()  
    var body: some View {  
  
        NavigationStack(path: $model.path) {  
            VStack {  
                ButtonContent("First View") { model.path = ["first"] }  
                ...  
            }  
            .navigationDestination(for: String.self) { destination in  
                model.resolve(pathItem: destination)  
            }  
        }.environmentObject(model)  
    }  
}
```



```
import XCTest
@testable import YourApp // replace with your app's name
class CoordinatorTests: XCTestCase {

    func testResolveFirstPathItem() {
        let coordinator = Coordinator()
        coordinator.resolve(pathItem: "first")

        XCTAssertEqual(coordinator.path, ["first"])
        // Add more assertions here as needed
    }
}
```



```
func testResolveMultiplePathItems() {  
    let coordinator = Coordinator()  
    coordinator.resolve(pathItem: "first")  
    coordinator.resolve(pathItem: "second")  
    coordinator.resolve(pathItem: "third")  
    coordinator.resolve(pathItem: "fourth")  
  
    XCTAssertEqual(coordinator.path, ["first", "second", "third", "fourth"])  
    // Add more assertions here as needed  
}
```



А если на сервис?



```
enum TranslateError: Error {}
```

```
class ServiceSearch: SearchServiceProtocol {  
    public var baseUrl = "https://url.ru/api/public/v1/"  
    let method = "words/search"  
    var session = URLSession.shared
```

```
    func search(word: String,  
                then handler: @escaping Handler) {  
        guard let url = URL(string: "\(baseUrl)\(method)?search=\(word)") else {  
            handler(.failure(.urlError))  
            return  
        }  
    }  
}
```



```
let task = session.dataTask(with: url) { result in
    switch result {
    case .success(let data):
        do {
            let translate = try JSONDecoder().decode([Translate].self, from: data)
            handler(.success(translate))
        } catch {
            handler(.failure(.invalidData))
        }
    case .failure(let error):
        handler(.failure(.networkFailure(error)))
    }
}
task.resume()
}
```



```
class ServiceSearchTests: XCTestCase {  
  
    var searchService: ServiceSearch!  
  
    override func setUpWithError() throws {  
        try super.setUpWithError()  
        searchService = ServiceSearch()  
    }  
  
    override func tearDownWithError() throws {  
        searchService = nil  
        try super.tearDownWithError()  
    }  
}
```



```
func testSearch_WithValidWord_ReturnsResult() {
    let expectation = XCTestExpectation(description: "Search returns result")
    searchService.search(word: "apple") { result in
        switch result {
        case .success(let translate):
            XCTAssertGreaterThan(translate.count, 0)
        case .failure(let error):
            XCTFail("Expected successful search, but got error: \(error)")
        }
        expectation.fulfill()
    }
    wait(for: [expectation], timeout: 5.0)
}
```




```
func testSearch_WithInvalidWord_ReturnsError() {  
    let expectation = XCTestExpectation(description: "Search returns error")  
    searchService.search(word: "xyzzz") { result in  
        switch result {  
        case .success:  
            XCTFail("Expected failed search, but got success")  
        case .failure(let error):  
            XCTAssertEqual(error, TranslateError.invalidData)  
        }  
        expectation.fulfill()  
    }  
    wait(for: [expectation], timeout: 5.0)  
}
```



Проблемы ChatGPT



Проблемы взаимодействия

01

ChatGPT учится
на «всяком» коде



Проблемы взаимодействия

01

ChatGPT учится
на «всяком» коде

02

Демотивирует изучать
документацию



Проблемы взаимодействия

01

ChatGPT учится
на «всяком» коде

02

Демотивирует изучать
документацию

03

Если задача отходит от шаблона —
заставляет попотеть



Проблемы взаимодействия

01

ChatGPT учится
на «всяком» коде

02

Демотивирует изучать
документацию

03

Если задача отходит от шаблона —
заставляет попотеть

04

Качество использование с подпиской
и без может различаться



Проблемы взаимодействия

01

ChatGPT учится
на «всяком» коде

02

Демотивирует изучать
документацию

03

Если задача отходит от шаблона —
заставляет попотеть

04

Качество использование с подпиской
и без может различаться

05

Только текст и есть ограничение на ввод



Security

01

VPN + Зарубежный номер



Security

01

VPN + Зарубежный номер

02

Кто имеет доступ к вашему
коду со стороны ChatGPT?



Security

01

VPN + Зарубежный номер

02

Кто имеет доступ к вашему
коду со стороны ChatGPT?

03

В коде могут быть уязвимости



Выводы



Что смогли делегировать?



«Красит кнопки»



«Парсит JSON»



Конвертирует код



Повышает
качество кода



Пишет скрипты



Пишет Unit test'ы



Что смогли делегировать?



«Красит кнопки»



«Парсит JSON»



Конвертирует код



Повышает
качество кода



Пишет скрипты



Пишет Unit test'ы



Правила пользования

01

Не передавай пароли/
ключи/сертификаты



Правила пользования

01

Не передавай пароли/
ключи/сертификаты

02

Даже если код хороший —
не используй точь-в-точь



Правила пользования

01

Не передавай пароли/
ключи/сертификаты

02

Даже если код хороший —
не используй точь-в-точь

03

Используй персональную учетку
и желательно персональный номер



Правила пользования

01

Не передавай пароли/
ключи/сертификаты

02

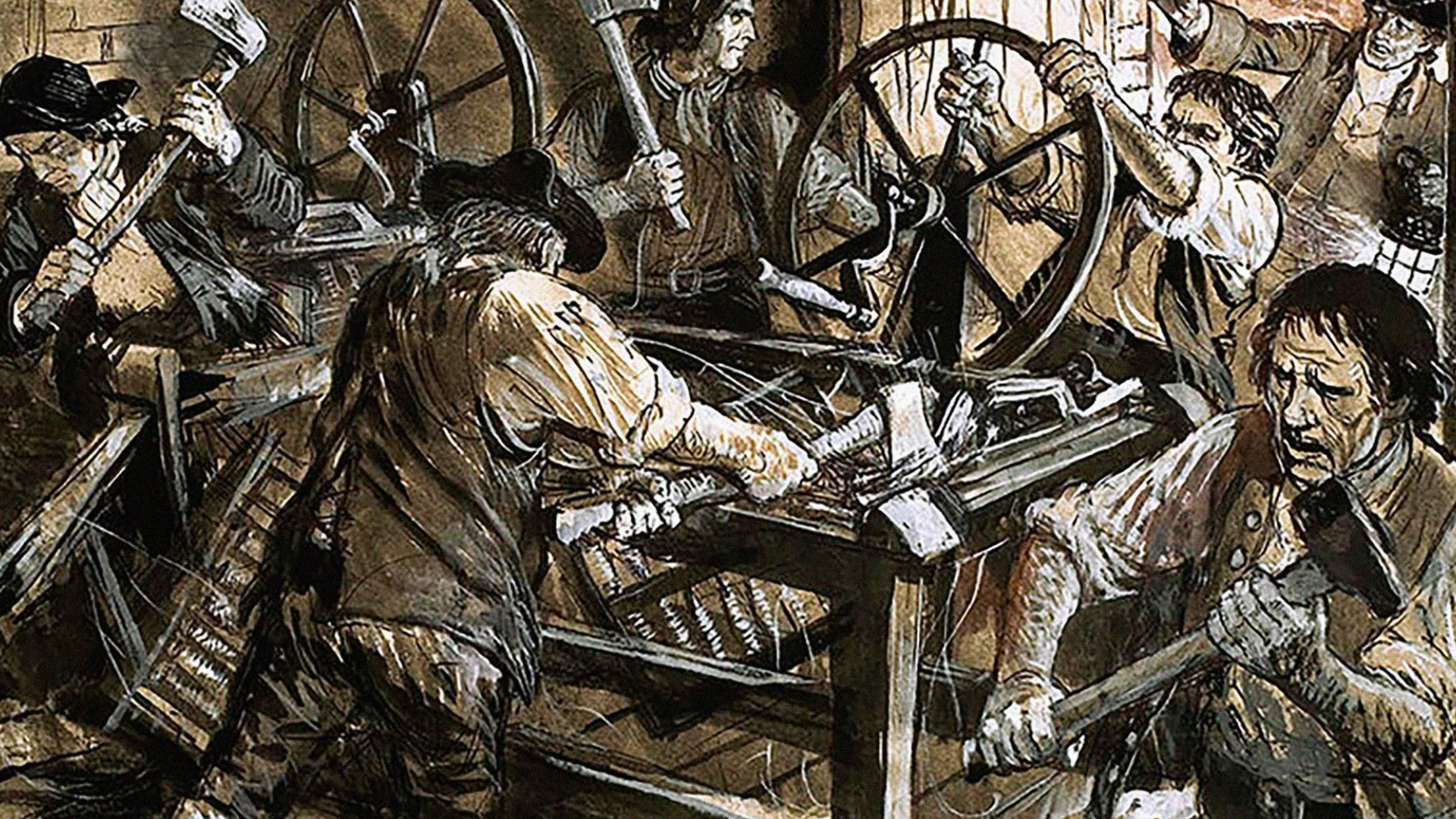
Даже если код хороший —
не используй точь-в-точь

03

Используй персональную учетку
и желательно персональный номер

04

Проверяй код на качество и дорабатывай





Не можешь победить — возглавь

Н. Макиавелли



Вопросы



Перейти в блог ChatGPT*

* <https://openai.com/blog/chatgpt>