



# Автоматичний аналіз рівня енергоспоживання мобільних застосунків

Слайди до доповіді

Виконала:

студентка БП-4 «Інженерія програмного забезпечення»

Грисюк А.О.

Науковий керівник:

старший викладач

Франків О.О.

# Актуальність дослідження



## Об'єкт дослідження

Енергоспоживання  
мобільних застосунків  
на операційній системі iOS

## Предмет дослідження

Методи статичного аналізу  
коду мобільних застосунків  
на Swift

## Результат кваліфікаційної роботи

EnergyUsageAnalyzer – інструмент для аналізу  
енергоспоживання мобільних застосунків

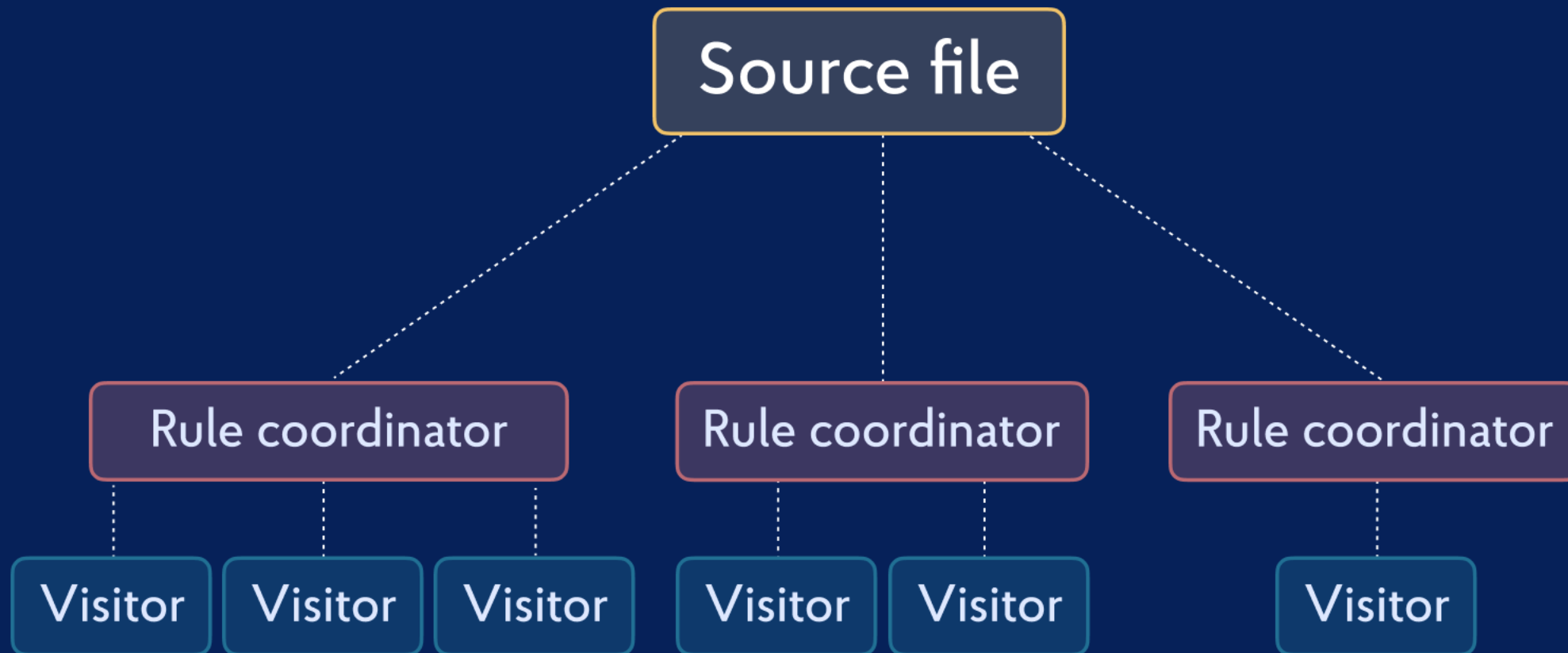


SwiftSyntax

Swift Package Plugins

ArgumentParser

# Принцип роботи аналізатора



## Високоточне оновлення локації

Конфігурація двох параметрів

- Точність розташування
- Мінімальна відстань, на яку пристрій повинен переміститися, перш ніж локація оновиться

## Відсутність зупинки локації, коли потреби в ній немає

Початок оновлення розташування  
+  
Відсутність зупинки оновлення  
розташування

Допустиме відхилення  
часу спрацювання таймера  
Максимальний час, через який  
таймер може спрацювати після  
запланованої дати

Відсутність зупинки таймера  
Постійне спрацювання таймера  
через заданий інтервал



Постійний пошук пристроїв  
початок сканування  
+  
відсутність завершення сканування

Напівпрозорість

Розмиття

Тінь

## Member ignoring method

метод, що не використовує жодної властивості чи функції власного класу; метод, який можна зробити статичним

## Draw Allocation

створення об'єктів у методі, призначеному для відмальовування інтерфейсу

## Parameter Object

функція, що приймає більше, ніж два параметри, котрі можна замінити одним об'єктом

## Вбудовування методу

```
func outer() {  
    inter()  
}
```

# Можливість деактивувати правила



```
energy-analyzer
EnergyUsageAnalyzer > energy-analyzer > No Selection
1 disabled_rules:
2   - shadow_rule
3   - draw_allocation_rule
4
```

energy-analyzer.yml

# Запуск аналізатора з командного рядка



```
cd EnergyUsageAnalyzer
```

```
swift run EnergyUsageAnalyzer -i test.swift
```

```
EnergyUsageAnalyzer — -zsh — 98x10
Building for debugging...
[4/4] Compiling EnergyUsageAnalyzer LocationAccuracyAnalyzer.swift
Build of product 'EnergyUsageAnalyzer' complete! (0.73s)
test.swift:60:9: warning: Found opacity usage, consider avoiding it (opacity_rule)
test.swift:60:9: warning: Found blur, consider avoiding it (blur_rule)
test.swift:60:9: warning: Found shadow usage, consider avoiding it (shadow_rule)
test.swift:15:9: warning: Found inefficient location configs: consider using desiredAccuracy = Kilometer and distanceFilter = 2^8 meters (location_accuracy_rule)
test.swift:39:9: warning: Found Timer named cancellable that doesn't stop (publish_timer_rule)
test.swift:39:23: warning: Found Timer that has no tolerance (tolerance_rule)
```

# Інтеграція плагіна в XCode



Project → Package Dependencies → «+»



Targets → Build Phases → Run Build Tool Plug-ins → «+»



Build

⌘ B



# Приклад роботи в XCode // GPU



The screenshot shows the Xcode IDE interface. At the top, the project name is "EnergyUsageAnaly...", the target is "iPhone 16 Plus (1)", and the build status is "Build Succeeded | Today at 6:05 PM". The editor displays a Swift file named "File3" with the following code:

```
1 import SwiftUI
2
3 struct ContentView: View {
4     var body: some View {
5         Rectangle()
6             .fill(Color.blue)
7             .opacity(0.5)
8             .frame(width: 100, height: 100)
9             .blur(radius: 5)
10            .shadow(color: |.gray, radius: 10, x: 5, y: 5)
11    }
12 }
13
14
```

On the right side of the editor, there are three yellow warning icons with the following messages:

- Found blur, consider avoiding it (blur\_rule)
- Found opacity usage, consider avoiding it (opacity\_rule)
- Found shadow usage, consider avoiding it (shadow\_rule)

At the bottom right of the editor, the status bar shows "Line: 10 Col: 28".

# Приклад роботи в XCode // CPU



```
EnergyUsageAnaly... iPhone 16 F Build Succeeded | 24.05.2025 at 1:44 PM 10
main
File File2
EnergyUsageAnalyzerTest EnergyUsageAnalyzerTest File2 M generateReport(title:author:date:)
1 class ReportGenerator { Found functions that can be static (instance_identifier_rule)
2
3 // Member Ignoring Method
4 func formatTitle(_ title: String) -> String {
5     return title.uppercased()
6 }
7
8 // Inline Method
9 func getCurrentDate() -> String { Found inline method, consider refactoring (inline_method_rule)
10     getDate()
11 }
12
13 func getDate() -> String {
14     return "2025-05-24"
15 }
16
17 // No Parameter Object
18 func generateReport(title: String, author: String, date: String) { Found 3 arguments, consider introducing a paramete...
19     print("Report Title: \(formatTitle(title))")
20     print("Author: \(author)")
21     print("Date: \(date)")
22 }
23 }
```

# Приклад роботи в XCode // Локація



```
1  import UIKit
2  import CoreLocation
3
4  class LocationSettingsViewController: UIViewController, CLLocationManagerDelegate {
5      let locationManager = CLLocationManager()
6
7      override func viewDidLoad() {
8          super.viewDidLoad()
9
10         locationManager.delegate = self
11         locationManager.requestWhenInUseAuthorization()
12
13         locationManager.desiredAccuracy = kCLLocationAccuracyBest
14         locationManager.distanceFilter = 256
15
16         locationManager.startUpdatingLocation()
17     }
18 }
```

Found inefficient location configs: consider using desiredAccuracy = Kilometer and dista...

Found location start call without a corresponding stop. Make sure to stop it when it!...

Line: 19 Col: 1

# Приклад роботи в XCode // Bluetooth



```
EnergyUsageAnaly... iPhone 16 Plus (1 Build Succeeded | Today at 6:10 PM 15
Bluetooth
EnergyUsageAnalyzerTest > EnergyUsageAnalyzerTest > Bluetooth > M centralManagerDidUpdateState(_:)
1 import CoreBluetooth
2
3 class BluetoothScanner: NSObject, CBCentralManagerDelegate {
4     var centralManager: CBCentralManager!
5
6     override init() {
7         super.init()
8         centralManager = CBCentralManager(delegate: self, queue: nil)
9     }
10
11     func centralManagerDidUpdateState(_ central: CBCentralManager) {
12         switch central.state {
13             case .poweredOn:
14                 print("Bluetooth is on. Starting scan...")
15                 centralManager.scanForPeripherals(withServices: nil, options: nil)
16
17                 DispatchQueue.main.async {
18                     print("Scan stopped.")
19                 }
20
21             default:
22                 print("Bluetooth is not available: \(central.state.rawValue)")
23         }
24     }
25 }
```

Found a Bluetooth scanning call without a corresponding stop. Make sure to stop scanning when it's no longer needed. (bluetooth\_rule)

Line: 18 Col: 1

# Приклад роботи в XCode // Timer



The screenshot shows the Xcode IDE interface. At the top, the project name is "EnergyUsageAnaly...", the target is "iPhone 16 P", and the build status is "Build Succeeded | 24.05.2025 at 1:44 PM". The breadcrumb navigation shows the file path: "EnergyUsageAnalyzerTest > EnergyUsageAnalyzerTest > File > startTimer()".

```
1 import SwiftUI
2 import Combine
3
4 final class CombineTimerManager: ObservableObject {
5     private var cancellable: AnyCancellable?
6
7     func startTimer() {
8         cancellable = Timer.publish(every: 1.0, on: .main, in: .common)|
9             .autoconnect()
10            .sink { date in
11                print("Combine Timer Tick: \(date)")
12            }
13    }
14 }
```

Two warnings are displayed in a tooltip:

- Found Timer named cancellable that doesn't stop (publish\_timer\_rule)
- Found Timer that has no tolerance (tolerance\_rule)

The status bar at the bottom right indicates "Line: 8 Col: 72".

1

Визначено шкідливі практики написання коду

2

Сформовано правила пошуку антипатернів, що збільшують енергоспоживання застосунків

3

Створено аналізатор, який можливо інтегрувати в XCode та запускати з терміналу