От модуляризации к Clang
И обратно

Roman Gorbenko

Путешествия

2021 2022 2023 2024

2021 2022 2023 2024

3 👨‍💻

2021 2022 2023 2024

3 👩‍💻 17 👩‍💻

2021 2022 2023 2024

3 👨‍💻

17 👨‍💻

~60k

2021 2022 2023 2024

3 👩‍💻

~60k

17 👩‍💻

~300k

Нынешние стандарты

Нынешние стандарты

Тестируемость

Нынешние стандарты

Тестируемость

Скорость сборки

Нынешние стандарты

Тестируемость

Скорость сборки

Малая связность кода

Нынешние стандарты

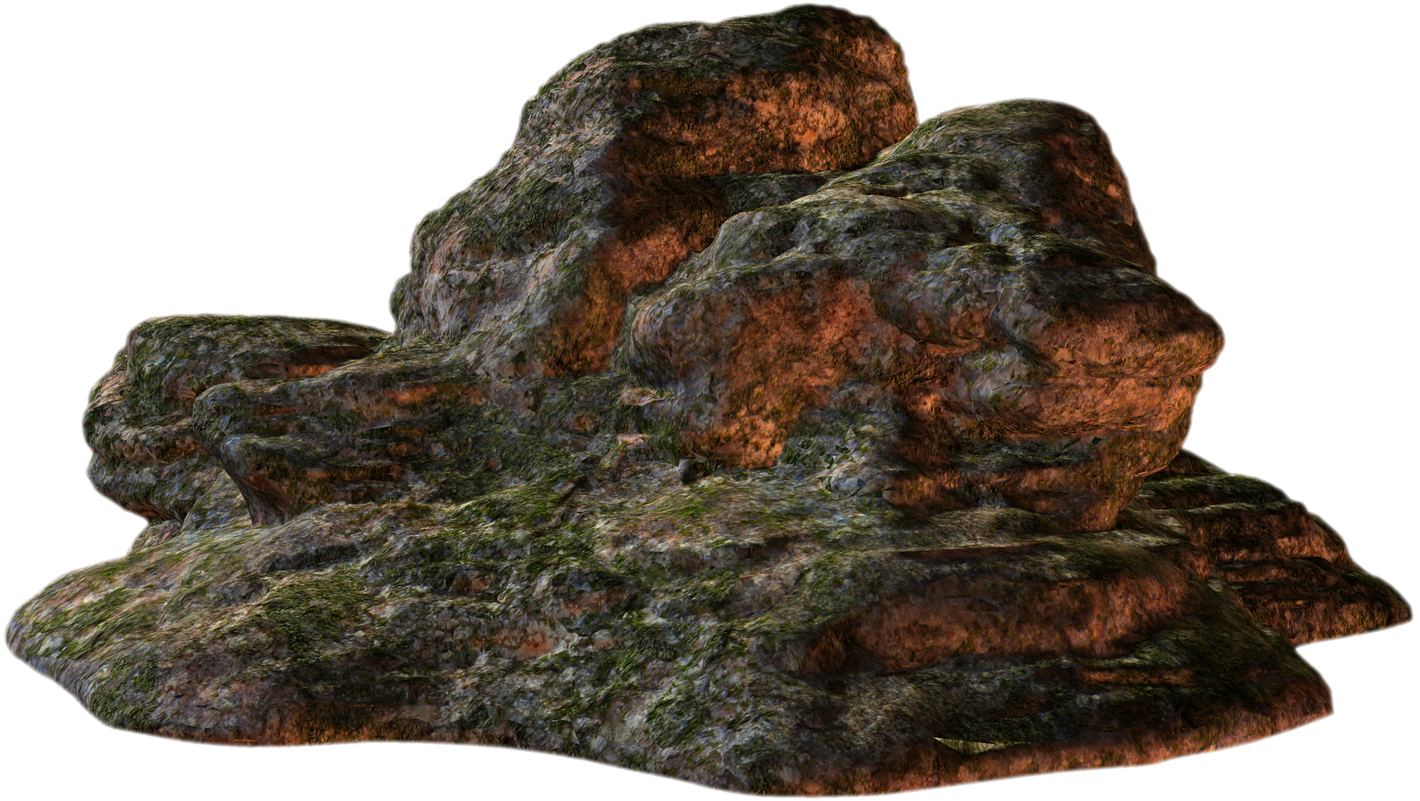
Тестируемость

Скорость сборки

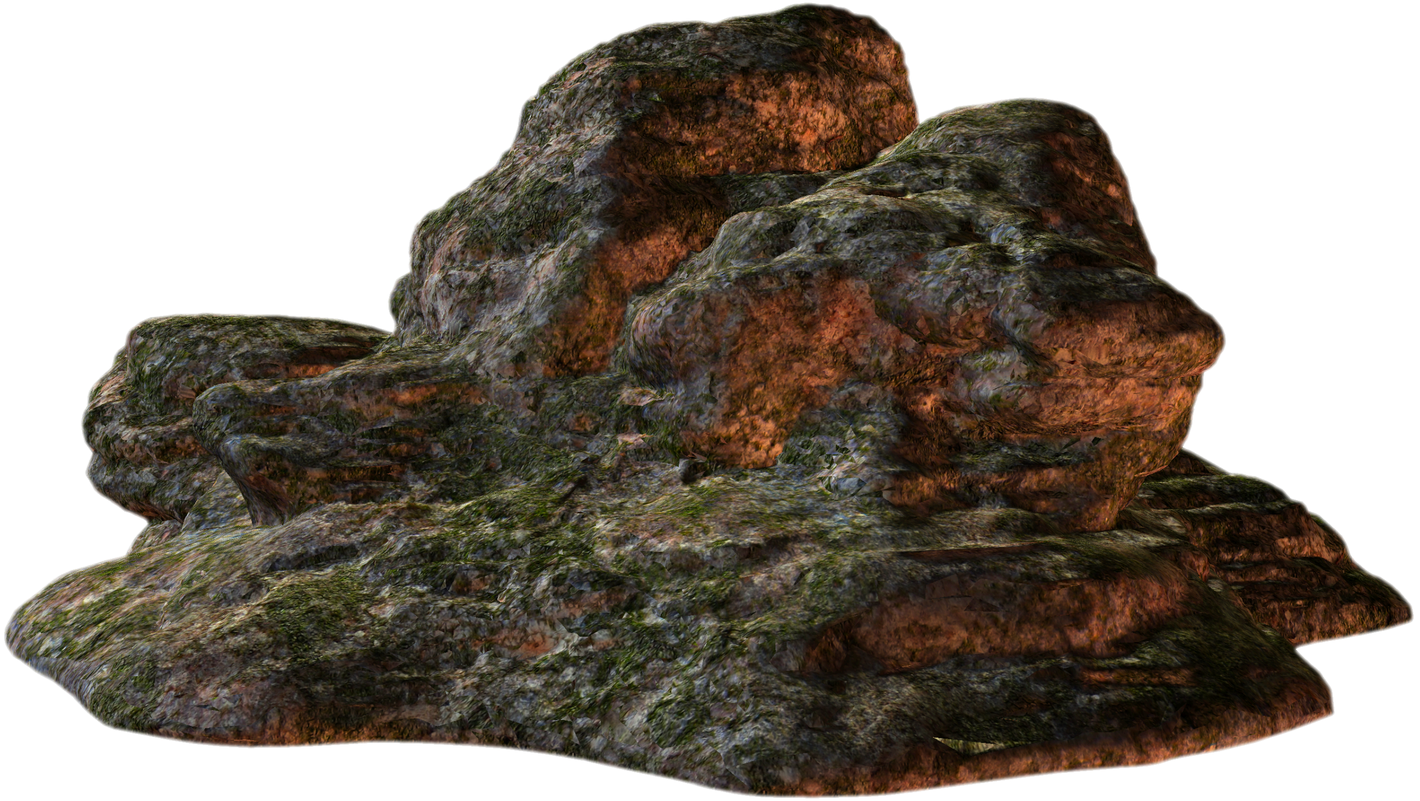
Малая связность кода

Переиспользуемость кода

Проблема



Проблема



Монолит

Нужно пилить, но как?


Модуляризация - это топ

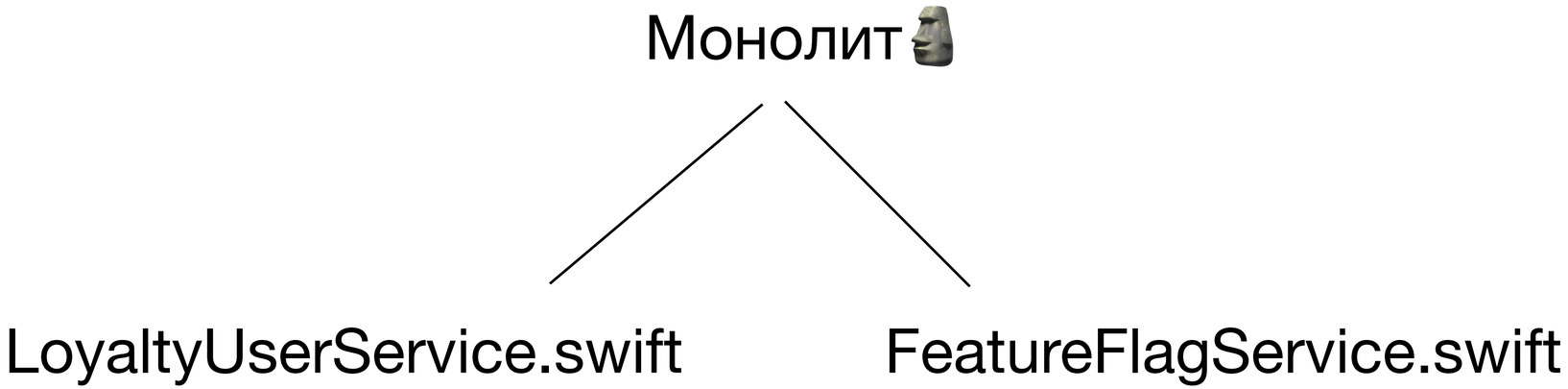
Модуляризация - это топ

Сколько SP?

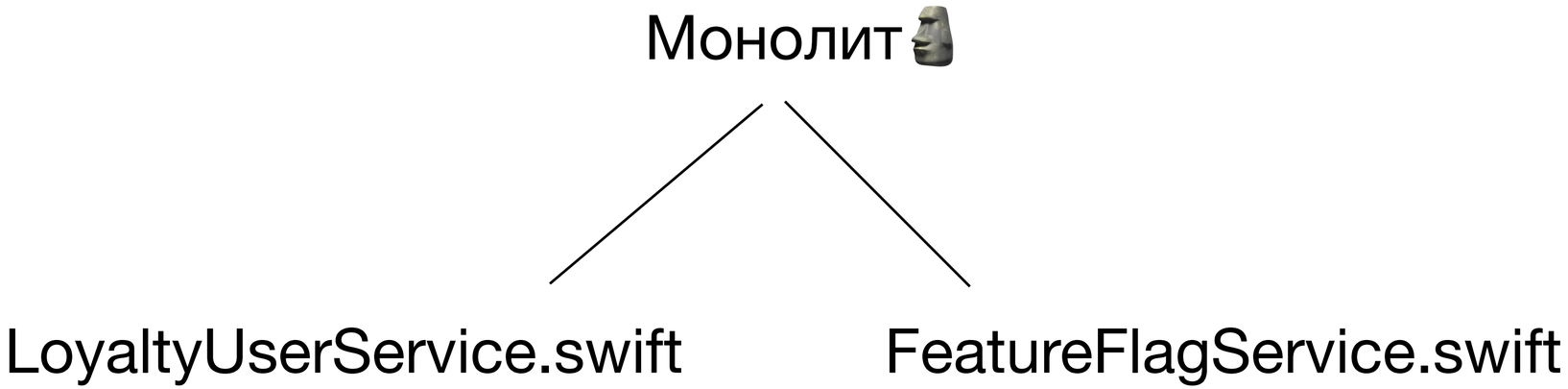
Тыры-пыры. Штук 40-50

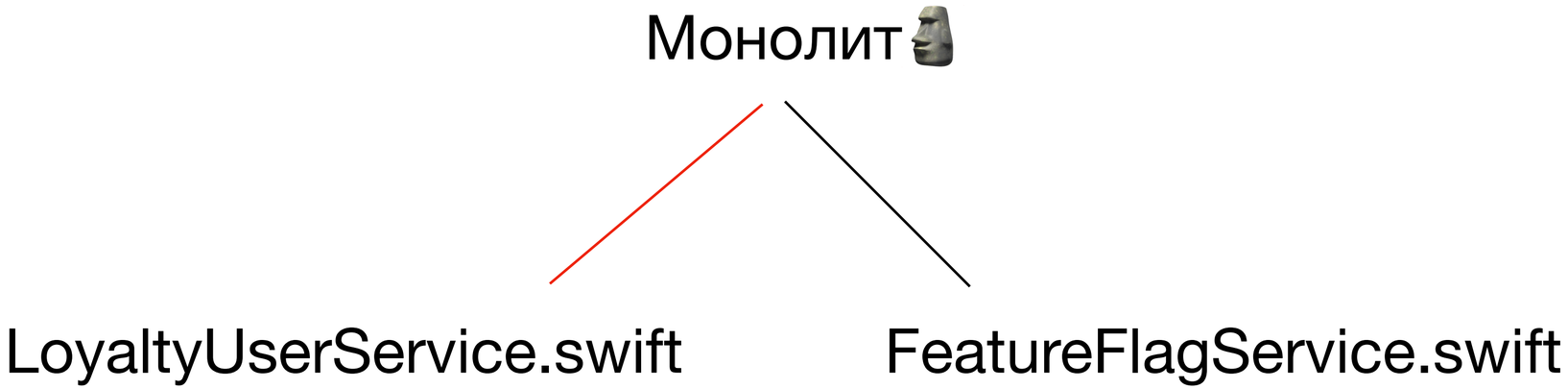
Сколько SP?

Отсутствие общей картины модуля


Монолит 🗿
LoyaltyUserService.swift
FeatureFlagService.swift

```
1   public class LoyaltyUserService {
2       let featureFlagService: FeatureFlagService
3   }
```


Монолит 🗿
LoyaltyUserService.swift
FeatureFlagService.swift


Монолит 🗿
LoyaltyUserService.swift
FeatureFlagService.swift

LoyaltyModule

LoyaltyUserService.swift

Монолит🗿

FeatureFlagService.swift

```swift
public class LoyaltyUserService {
    let featureFlagService: FeatureFlagService
}
```

LoyaltyUserService.swift

```swift
1  public class LoyaltyUserService {
2      let featureFlagService: FeatureFlagService
3  }
```

Cannot find type 'FeatureFlagService' in scope

LoyaltyUserService.swift

Глава 1. Статический анализ

Статический анализ

Ловим ошибку

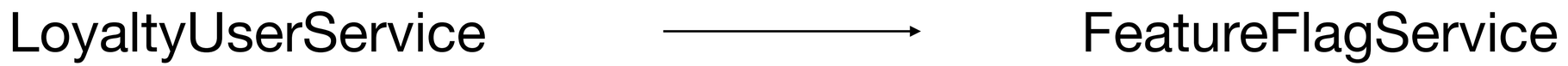
```
1   public class LoyaltyUserService {
2       let featureFlagService: FeatureFlagService
3   }
```

Cannot find type 'FeatureFlagService' in scope

LoyaltyUserService.swift

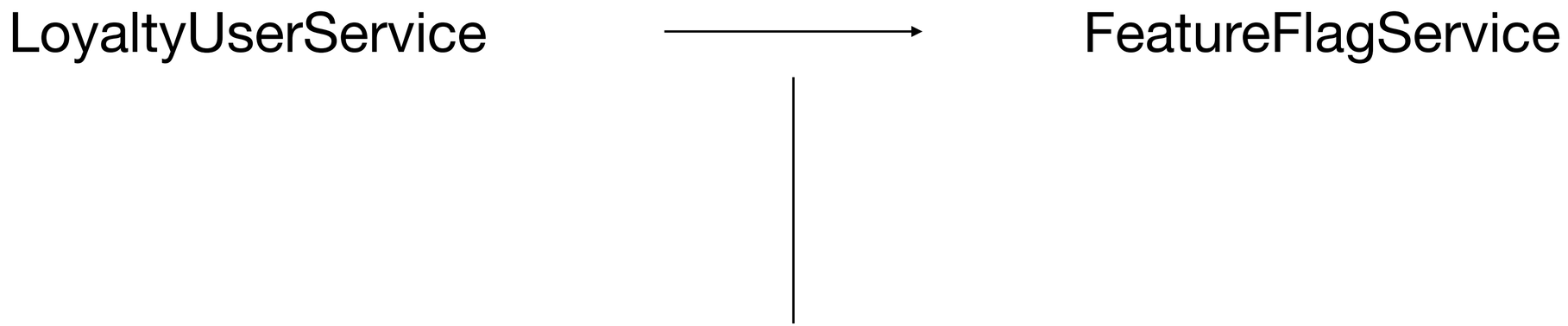
Статический анализ

Зависимость



Статический анализ

Зависимость



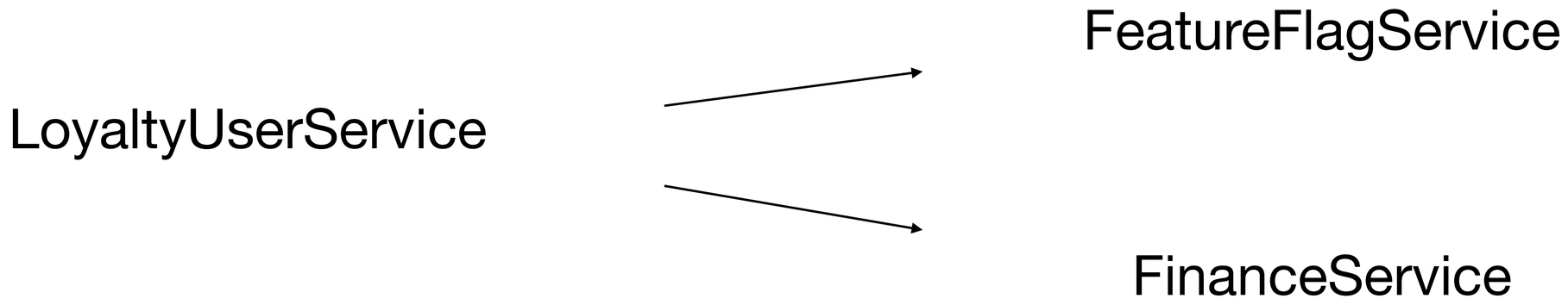
Статический анализ

Пример зависимости

```
1   public class LoyaltyUserService {
2       func apply(financeStore: FinanceService) {
3           …
4       }
5   }
```

Статический анализ

Пример зависимости



Статический анализ

Находим лексему с большой буквой в теле сущности -> заносим в граф

Статический анализ

Шаг 1: Найти объявление сущности

Шаг 2: Найти все лексемы с большой буквы и внести их в граф

Статический анализ

```
1   public class LoyaltyUserService {
2       let featureFlagService: FeatureFlagService
3   }
```

Статический анализ

```
1   public class LoyaltyUserService {
2       let featureFlagService: FeatureFlagService
3   }
```

Статический анализ

```
1   public class LoyaltyUserService {
2       let featureFlagService: FeatureFlagService
3   }
```

LoyaltyUserService.swift

Статический анализ

Пример зависимости

```
1   public class LoyaltyUserService {
2       func apply(financeStore: FinanceService) {
3           …
4       }
5   }
```

Статический анализ

Пример зависимости

```
1   public class LoyaltyUserService {
2       func apply(financeStore: FinanceService) {
3           …
4       }
5   }
```

Статический анализ

Пример зависимости

```
1   public class LoyaltyUserService {
2       func apply(financeStore: FinanceService) {
3           …
4       }
5   }
```

Статический анализ

Пример зависимости

```
protocol LoyaltyService {
    func addLoyaltyPoints(to user: User, points: LoyaltyPoints)
    func getLoyaltyPoints(to user: User) -> LoyaltyPoints
    func redeemLoyaltyPoints(from user: User, points: LoyaltyPoints) -> Bool
}
```

Статический анализ

Пример зависимости

```
1  protocol LoyaltyService {
2      func addLoyaltyPoints(to user: User, points: LoyaltyPoints)
3      func getLoyaltyPoints(to user: User) -> LoyaltyPoints
4      func redeemLoyaltyPoints(from user: User, points: LoyaltyPoints) -> Bool
5  }
```

Статический анализ

Пример зависимости

```
protocol LoyaltyService {
    func addLoyaltyPoints(to user: User, points: LoyaltyPoints)
    func getLoyaltyPoints(to user: User) -> LoyaltyPoints
    func redeemLoyaltyPoints(from user: User, points: LoyaltyPoints) -> Bool
}
```

Статический анализ

Пример зависимости

```
protocol LoyaltyService {
    func addLoyaltyPoints(to user: User, points: LoyaltyPoints)
    func getLoyaltyPoints(to user: User) -> LoyaltyPoints
    func redeemLoyaltyPoints(from user: User, points: LoyaltyPoints) -> Bool
}
```

Статический анализ

Не делайте название сущности с маленькой буквы(пожалуйста)

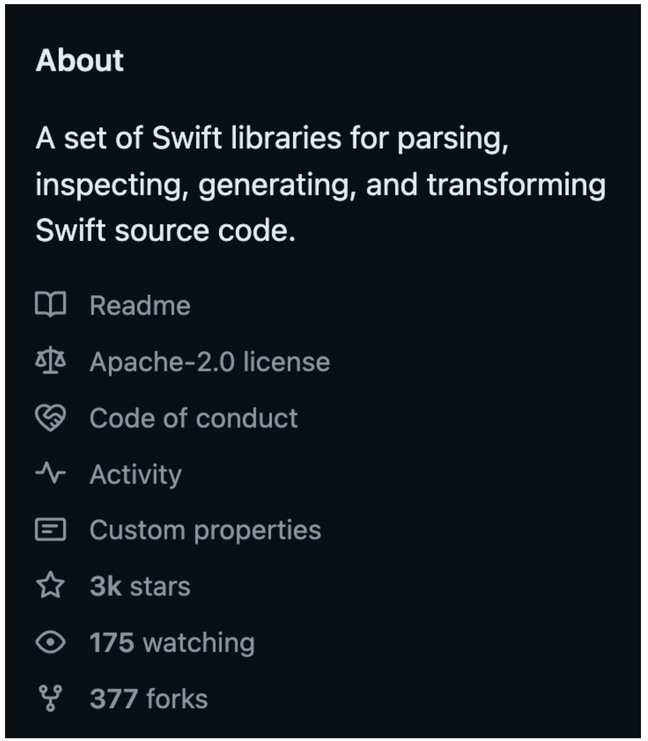
Статический анализ

Шаг 1: Объявление сущности

Шаг 2: Все лексемы с большой буквы

Статический анализ

На помощь пришел SwiftSyntax



Статический анализ

Что может SwiftSyntax?

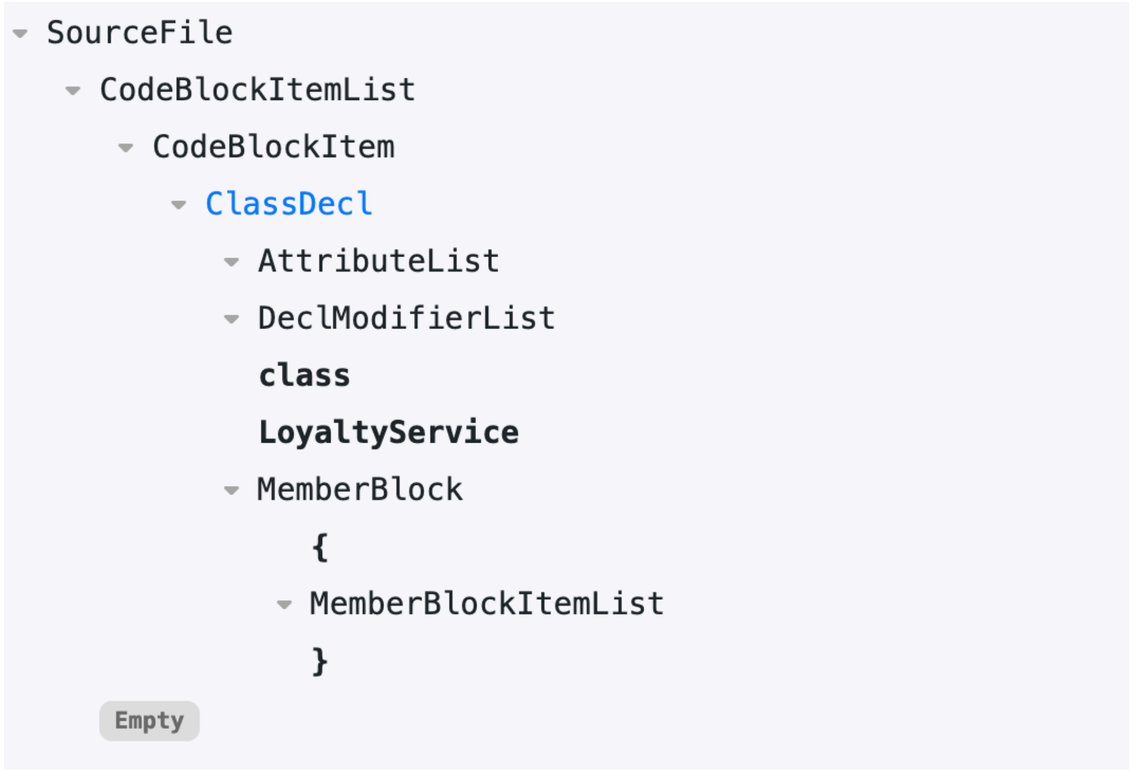
1. Анализировать AST дерево

Статический анализ

Что может SwiftSyntax?

1. Анализировать AST дерево

```
1   class LoyaltyService { }
```

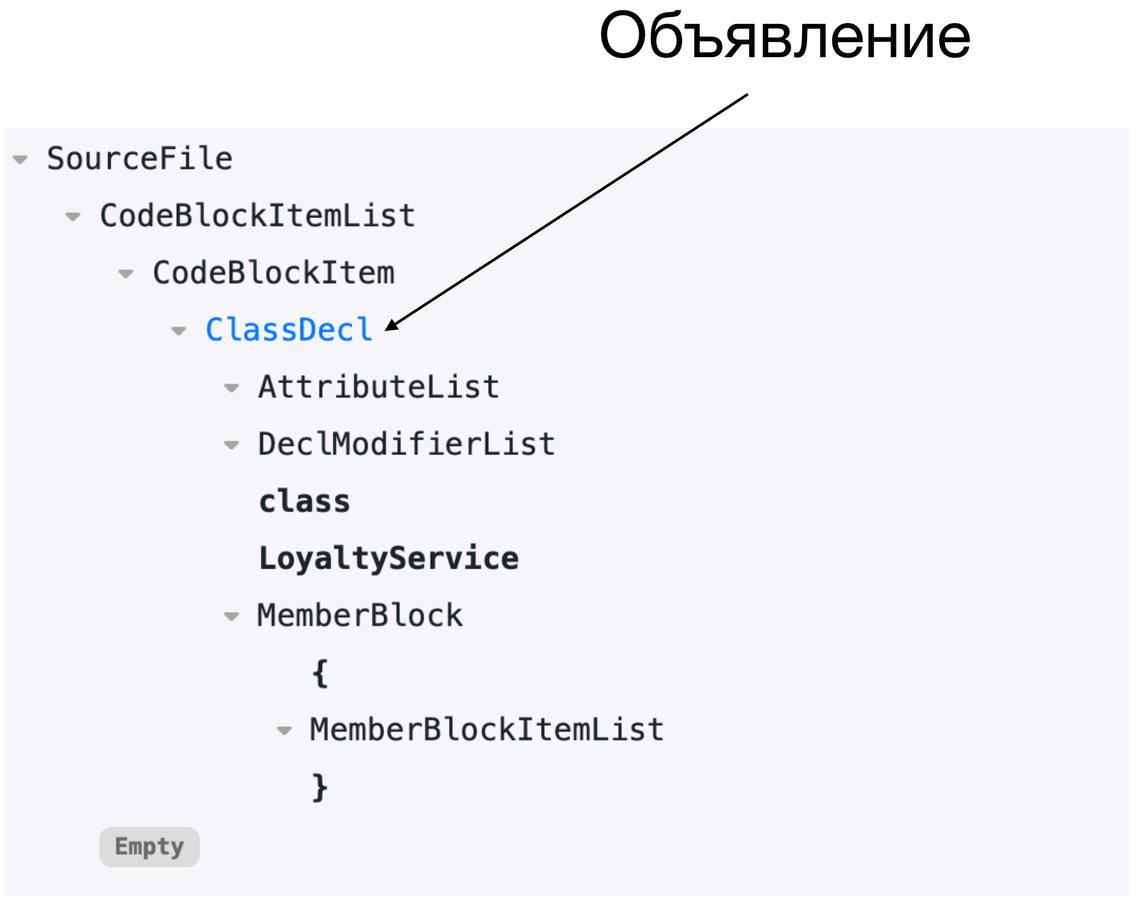


Статический анализ

Что может SwiftSyntax?

1. Анализировать AST дерево

```
1   class LoyaltyService { }
```



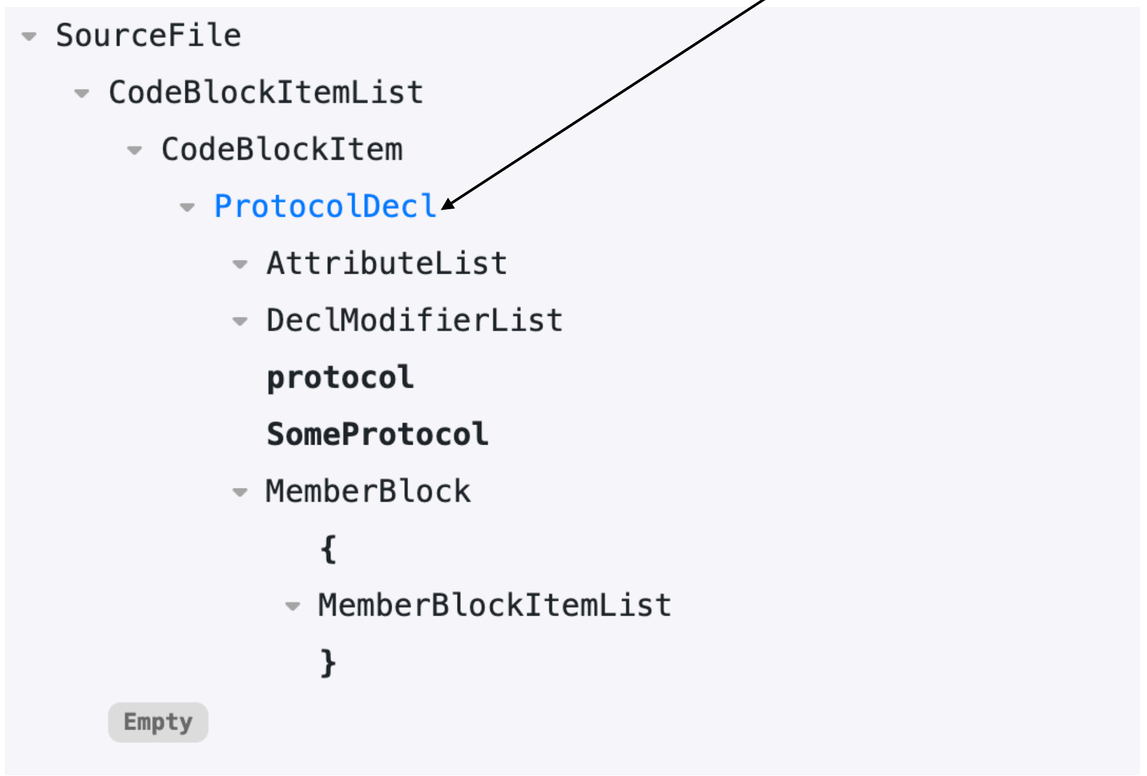
Статический анализ

Что может SwiftSyntax?

1. Анализировать AST дерево

```
1   protocol SomeProtocol { }
```

Объявление

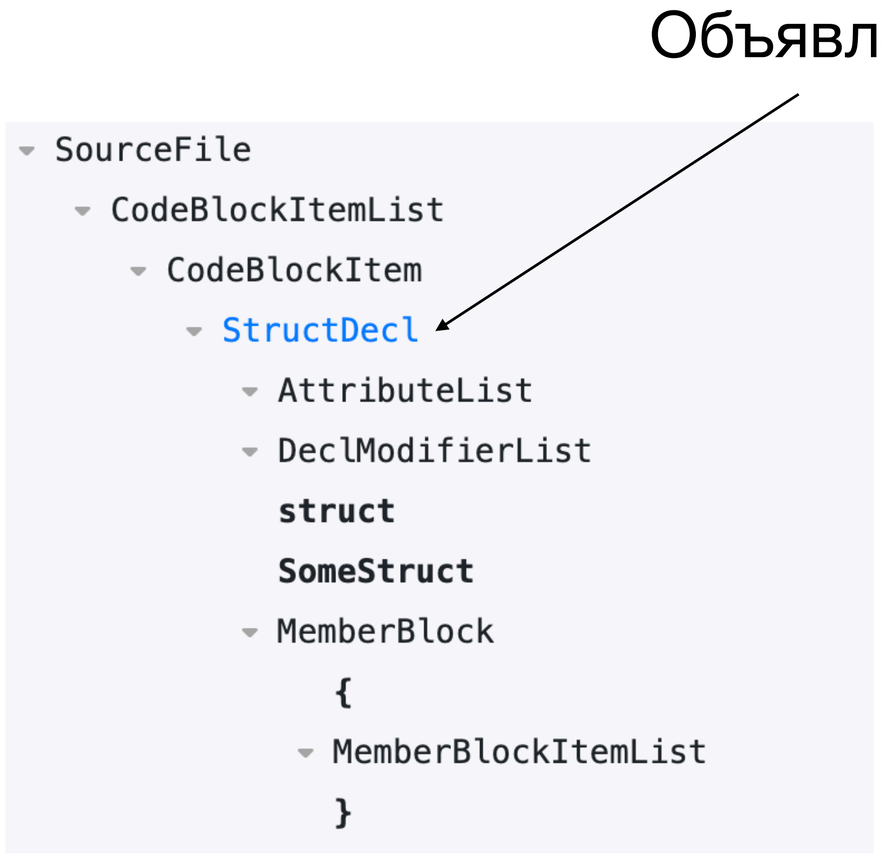


Статический анализ

Что может SwiftSyntax?

1. Анализировать AST дерево

```
1   struct SomeStruct { }
```

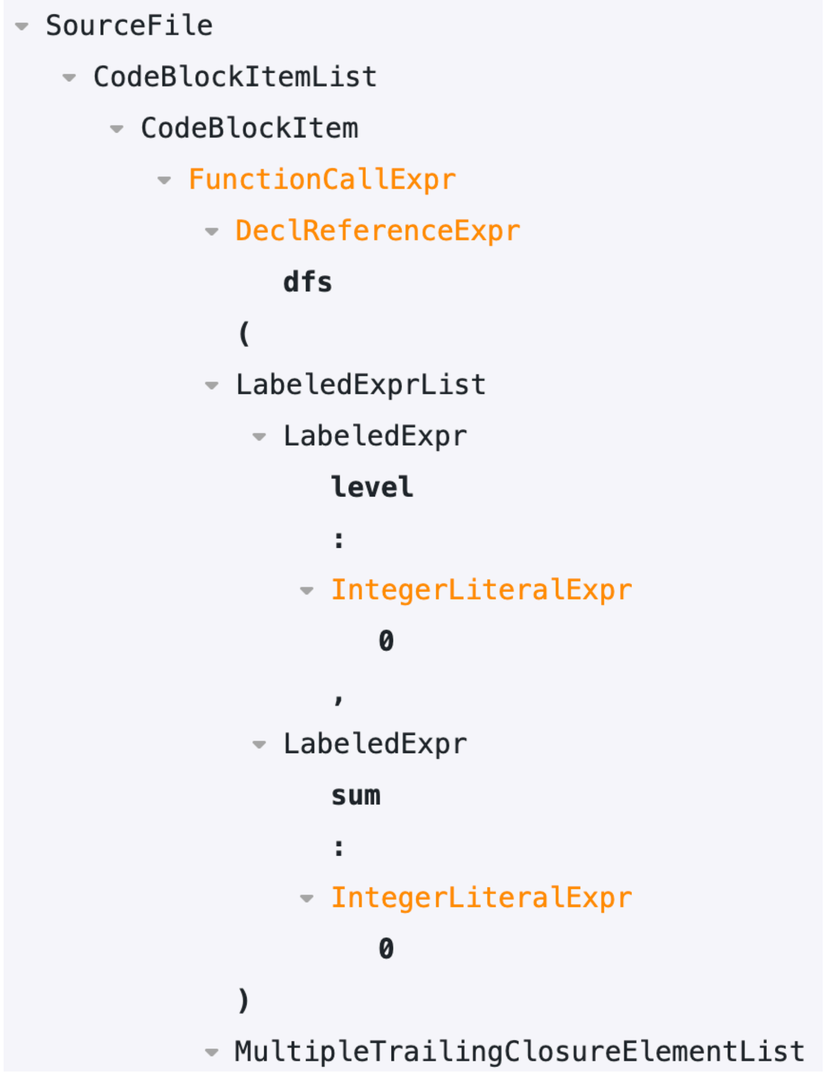


Статический анализ

Что может SwiftSyntax?

1. Анализировать AST дерево

```
1   dfs(level: 0, sum: 0)
```



Статический анализ

Что может SwiftSyntax?

1. Анализировать AST дерево

```
1   dfs(level: 0, sum: 0)
```

Статический анализ

Что может SwiftSyntax?

1. Анализировать AST дерево

```
1   dfs(level: 0, sum: 0)
```

Статический анализ

Что может SwiftSyntax?

1. Анализировать AST дерево

```
1   dfs(level: 0, sum: 0)
```

Статический анализ

Что может SwiftSyntax?

1. Анализировать AST дерево

```
1   dfs(level: 0, sum: 0)
```

Статический анализ

Что может SwiftSyntax?

1. Анализировать AST дерево

```
1   dfs(level: 0, sum: 0)
```

Статический анализ

Что может SwiftSyntax?

1. Анализировать AST дерево

```
1   dfs(level: 0, sum: 0)
```

Статический анализ

Что может SwiftSyntax?

1. Анализировать AST дерево

```
1   dfs(level: 0, sum: 0)
```

Статический анализ

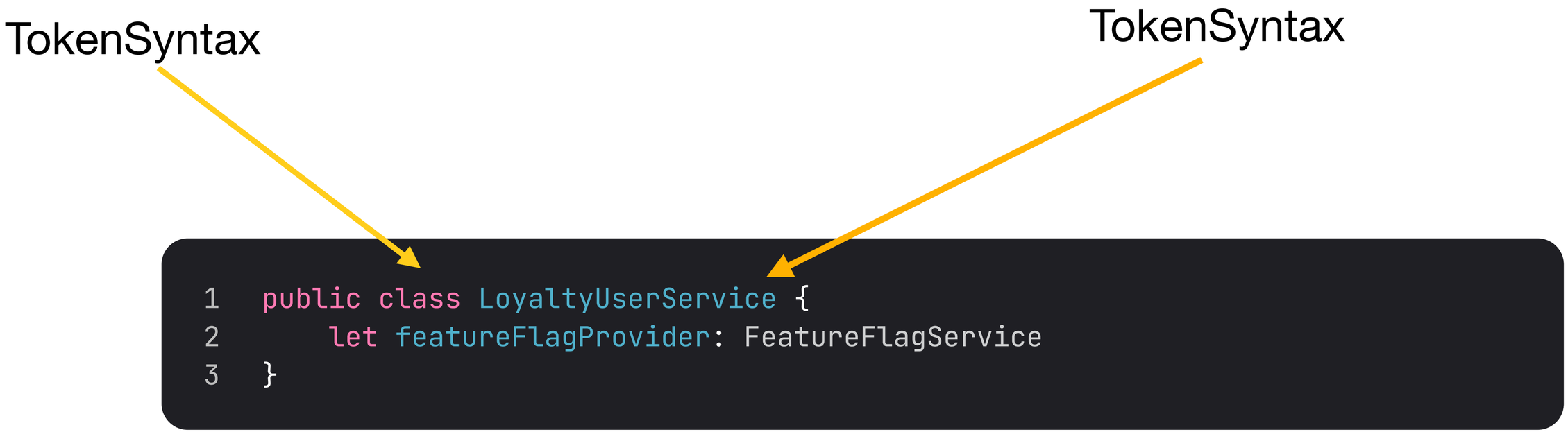
TokenSyntax

TokenSyntax

```
1   public class LoyaltyUserService {
2       let featureFlagProvider: FeatureFlagService
3   }
```

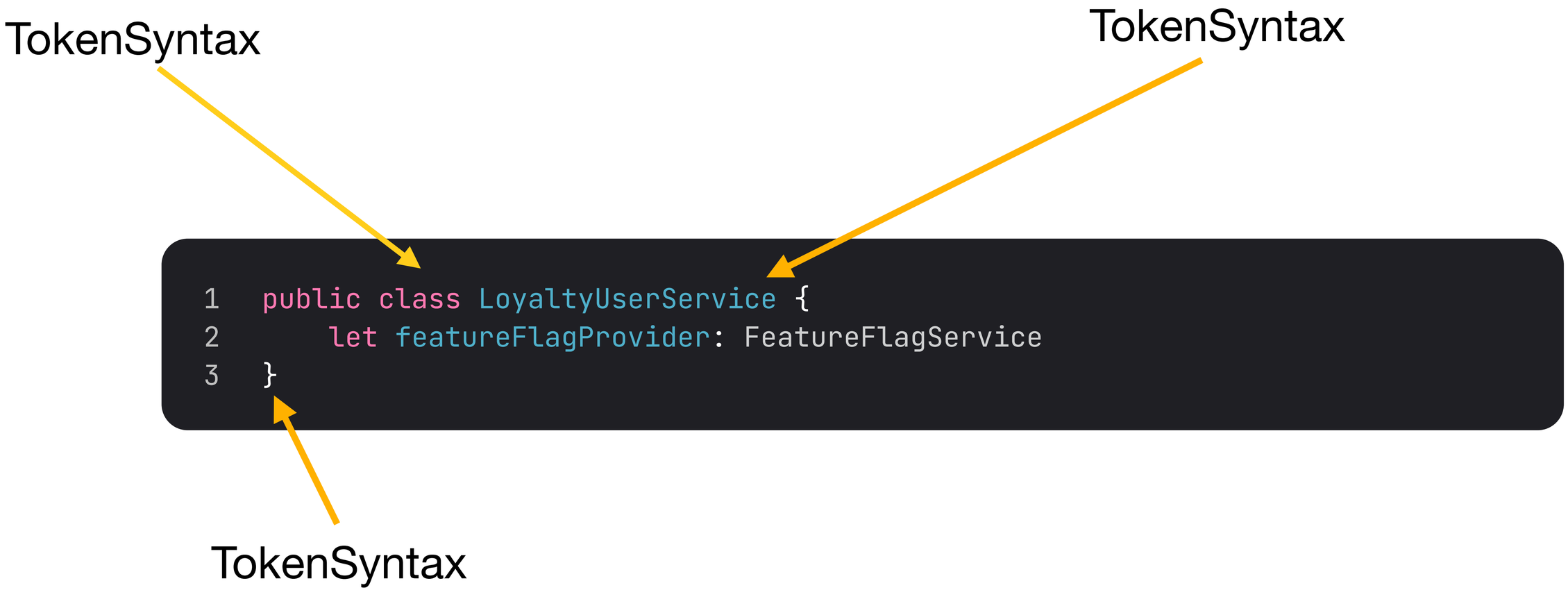
Статический анализ

TokenSyntax



Статический анализ

TokenSyntax



Статический анализ

TokenSyntax

Все есть TokenSyntax

Статический анализ

TokenSyntax

Шаг 1: Объявление сущности

Шаг 2: Все лексемы с большой буквы

Статический анализ

TokenSyntax

Шаг 1: Объявление сущности

Шаг 1: ClassDecl/StructDecl…

Статический анализ

TokenSyntax

Шаг 2: Все лексемы с большой буквы

Шаг 2: Все TokenSyntax с большой буквы

Статический анализ

TokenSyntax

Шаг 1: Найти ClassDecl/StructDecl…

Шаг 2: Найти все TokenSyntax с большой буквы

Статический анализ

Составление графа

```
1   public class LoyaltyUserService {
2       let featureFlagProvider: FeatureFlagService
3       func apply(financeStore: FinanceService) { }
4   }
```

Статический анализ

Составление графа

```
1   public class LoyaltyUserService {
2       let featureFlagProvider: FeatureFlagService
3       func apply(financeStore: FinanceService) { }
4   }
```

```
Dict {
  LoyaltyUserService: Set(FeatureFlagService, FinanceService)
}
```

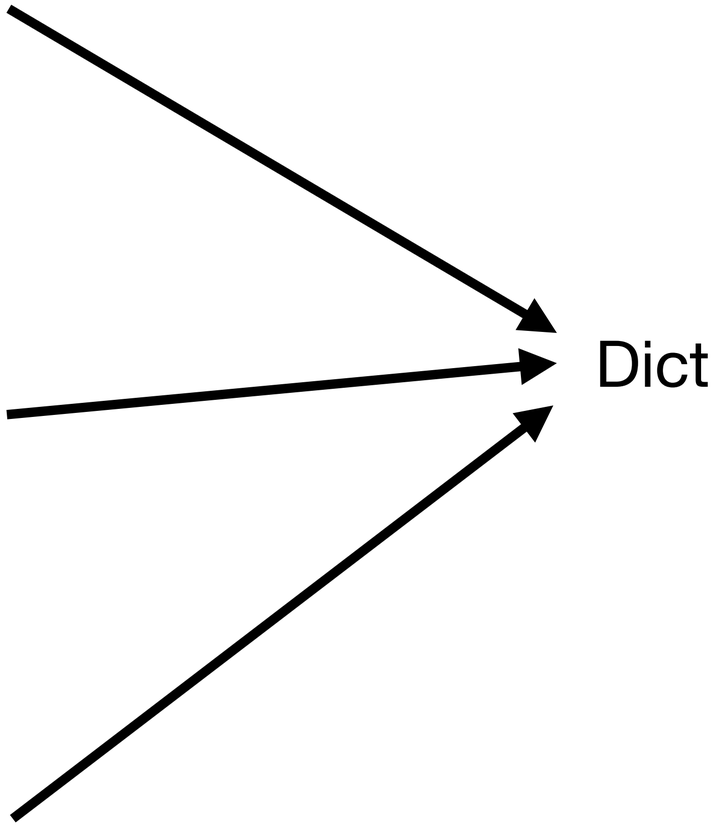
Статический анализ

Составление графа

```
1  public class LoyaltyUserService {
2      let featureFlagProvider: FeatureFlagService
3      func apply(financeStore: FinanceService) { }
4  }
```

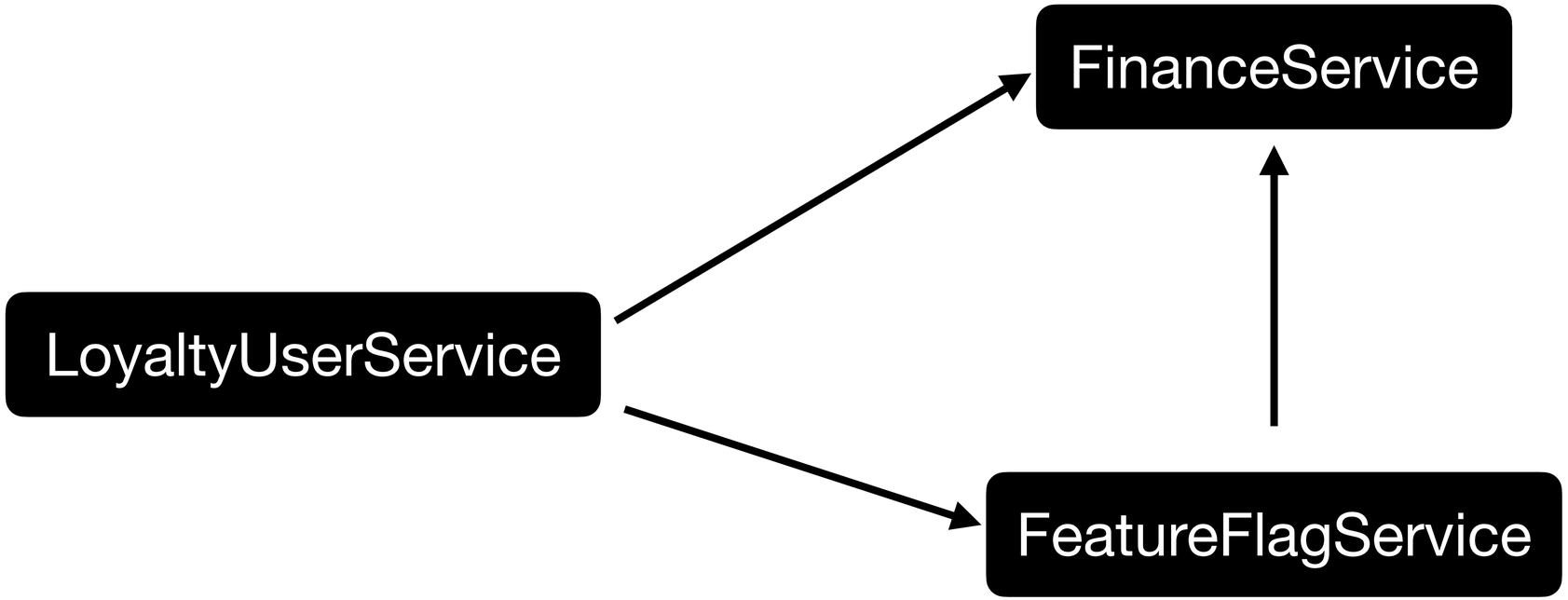
```
1  class FeatureFlagService {
2    let financeService: FinanceService
3  }
```

```
1  class FinanceService {
2  …
3  }
```



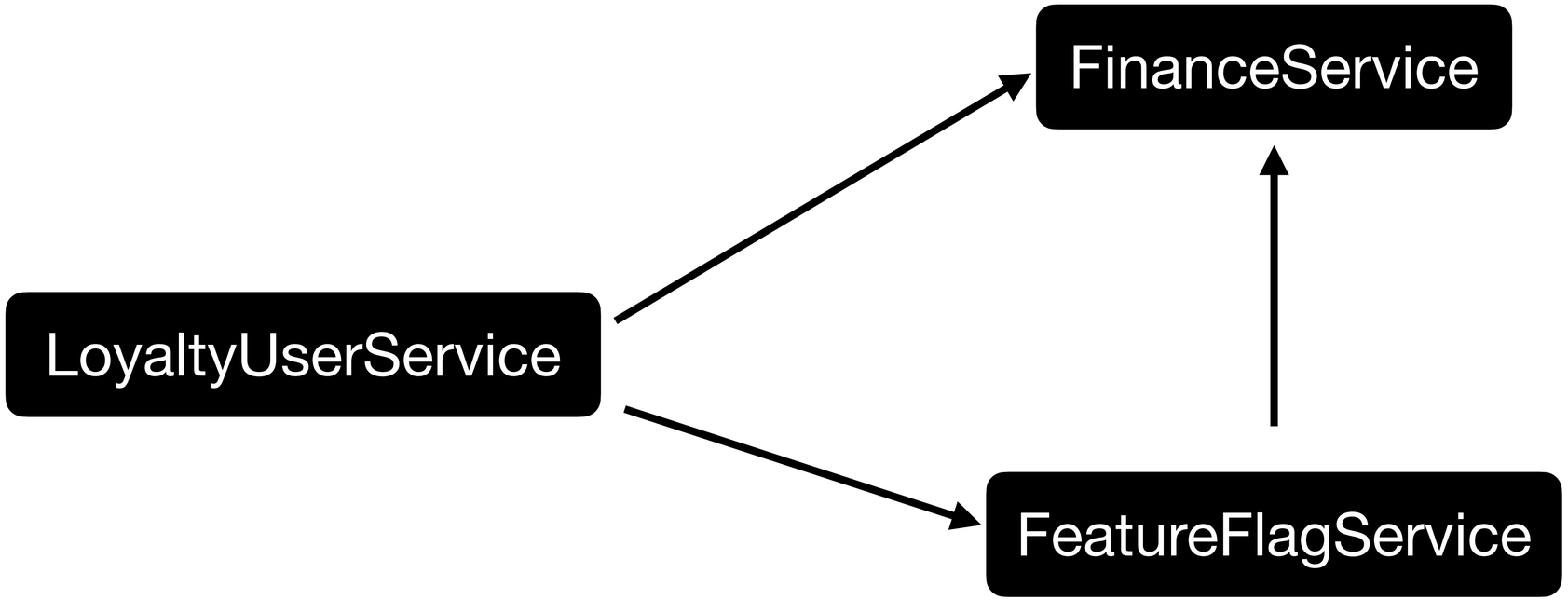
Статический анализ

Составление графа



Статический анализ

Составление графа



Ребро показывает их зависимость

Статический анализ

Полный путь

При помощи XcodeProj
Собирать пути ко всем файлам

Статический анализ

Пути к сорсам

```
1   static func getAllModules(projUrl: URL) throws -> [Module] {
2     let proj = try XcodeProj(path: .init(projUrl.path()))
3     return proj.pbxproj.nativeTargets
4       .map {
5         let moduleName = $0.name
6         var moduleFiles = [URL]()
7         let sourceBuildPhase = $0.buildPhases.filter {
8           $0 is PBXSourcesBuildPhase
9         }.first
10        if let sourceBuildPhase,
11          let files = sourceBuildPhase.files
12        {…}
```

Статический анализ

Полный путь

При помощи XcodeProj
собрать пути ко всем файлам

Провести статический
анализ файлов

Статический анализ

Полный путь

При помощи XcodeProj
собрать пути ко всем файлам

Провести статический
анализ файлов

Собрать из результатов
граф

Статический анализ

Таблица всех сущностей

Id	Label	..	module	path
MonolitApp.FinanceService	FinanceService		MonolitApp	/Users/rofle100lvl/Desktop/DD/MonolitApp/Monolit/Finance/FinanceService.swift
MonolitApp.FeatureFlagService	FeatureFlagService		MonolitApp	/Users/rofle100lvl/Desktop/DD/MonolitApp/Monolit/FeatureFlags/FeatureFlagService.swift
MonolitApp.rApp	rApp		MonolitApp	/Users/rofle100lvl/Desktop/DD/MonolitApp/Monolit/View.swift
MonolitApp.LoyaltyUserService	LoyaltyUserService		MonolitApp	/Users/rofle100lvl/Desktop/DD/MonolitApp/Monolit/Loyalty/LoyaltyUserService.swift

Статический анализ

Таблица всех ребер

Source	Target
MonolitApp.FeatureFlagService	MonolitApp.FinanceService
MonolitApp.LoyaltyUserService	MonolitApp.FinanceService
MonolitApp.LoyaltyUserService	MonolitApp.FeatureFlagService

Глава 2. Анализ графа

Как отрисовать?

Анализ графа

Инструменты отображения

GraphViz

\+

\-

Анализ графа

Инструменты отображения

GraphViz

\+

\-

Есть удобное API

Анализ графа

Создание графа при помощи GraphViz

```
1   import GraphViz
2
3   var graph = Graph(directed: true)
4
5   for edge in edges {
6       graph.append(
7           GraphViz.Edge(
8               from: GraphViz.Node(edge.from.id),
9               to: GraphViz.Node(edge.to.id)
10          )
11      )
12  }
13
14  graph.render(using: .circo, to: .svg, completion: { _ in  })
```

Анализ графа

Инструменты отображения

GraphViz

\+

Есть удобное API

Поддерживает много форматов

\-

Анализ графа

Инструменты отображения

GraphViz

\+

Есть удобное API

Поддерживает много форматов

\-

Мало настроек отображения

Анализ графа

Инструменты отображения

GraphViz

\+

Есть удобное API

Поддерживает много форматов

\-

Мало настроек отображения

Не позволяет взаимодействовать с графом

Анализ графа

Инструменты отображения

+

Поддерживает очень много форматов

-

Анализ графа

Инструменты отображения

+

Поддерживает очень много форматов

Содержит в себе большой функционал для работы с графом

-

Анализ графа

Инструменты отображения

+

Поддерживает очень много форматов

Содержит в себе большой функционал для работы с графом

-

Высокий порог входа

Анализ графа

Инструменты отображения

+	-
Поддерживает очень много форматов	Высокий порог входа
Содержит в себе большой функционал для работы с графом	Нет удобного апи*

Анализ графа

Инструменты отображения

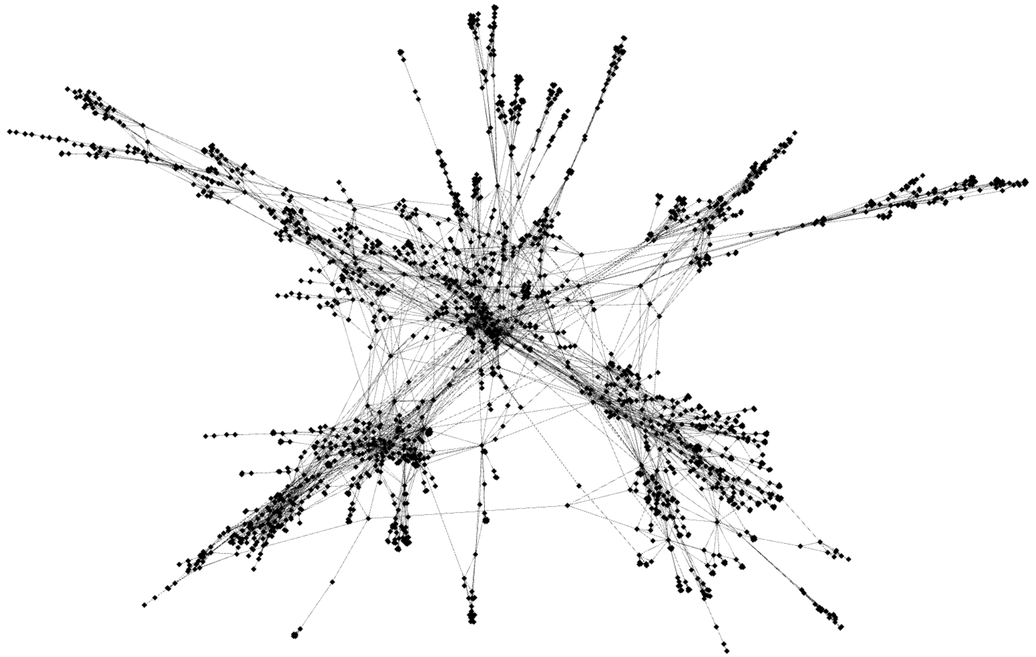
+	-
Поддерживает очень много форматов	Высокий порог входа
Содержит в себе большой функционал для работы с графом	Нет удобного апи*

*В каком-то виде оно есть, но написано на Java, как и сам Gephi

Итак, мы умеем рисовать граф, и…

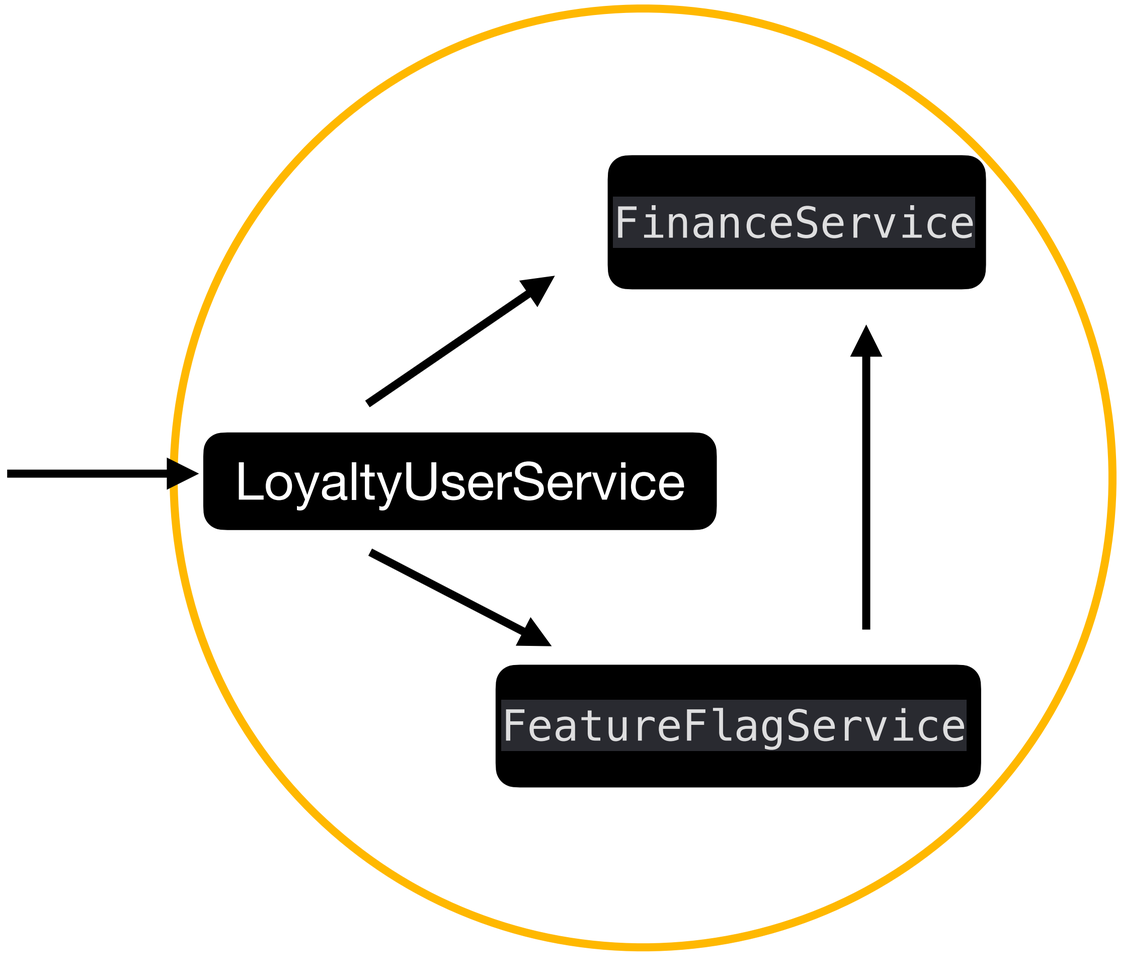
Анализ графа

Получившийся граф



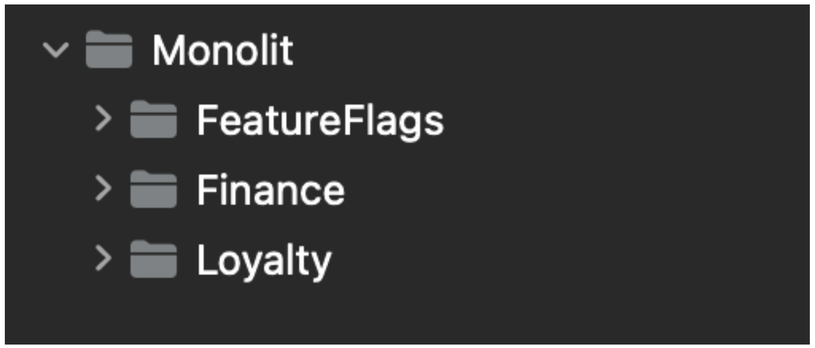
Анализ графа

Получившийся граф



Анализ графа

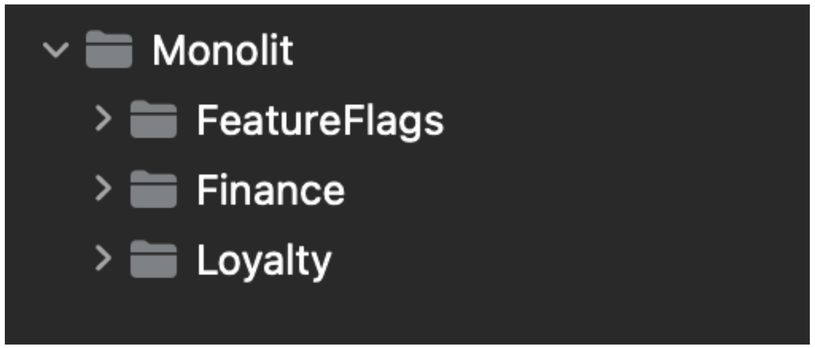
Наш проект



Неплохо сгруппированные по смыслу папки

Анализ графа

Связи



Анализ графа

Связи

Анализ графа

Связи

Анализ графа

Связи

Анализ графа

Связи

Как его распутывать?

Анализ графа

Задача

Найти все зависимости папки
Loyalty, мешающие нам вынести ее

Анализ графа

Собранные данные

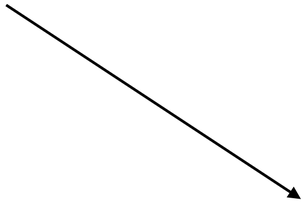
Id	Label	..	module	path
MonolitApp.FinanceService	FinanceService		MonolitApp	/Users/rofle100lvl/Desktop/DD/MonolitApp/Monolit/Finance/FinanceService.swift
MonolitApp.FeatureFlagService	FeatureFlagService		MonolitApp	/Users/rofle100lvl/Desktop/DD/MonolitApp/Monolit/FeatureFlags/FeatureFlagService.swift
MonolitApp.rApp	rApp		MonolitApp	/Users/rofle100lvl/Desktop/DD/MonolitApp/Monolit/View.swift
MonolitApp.LoyaltyUserService	LoyaltyUserService		MonolitApp	/Users/rofle100lvl/Desktop/DD/MonolitApp/Monolit/Loyalty/LoyaltyUserService.swift

Вспоминаем какие данные у нас есть

Анализ графа

Собранные данные

Путь к файлу

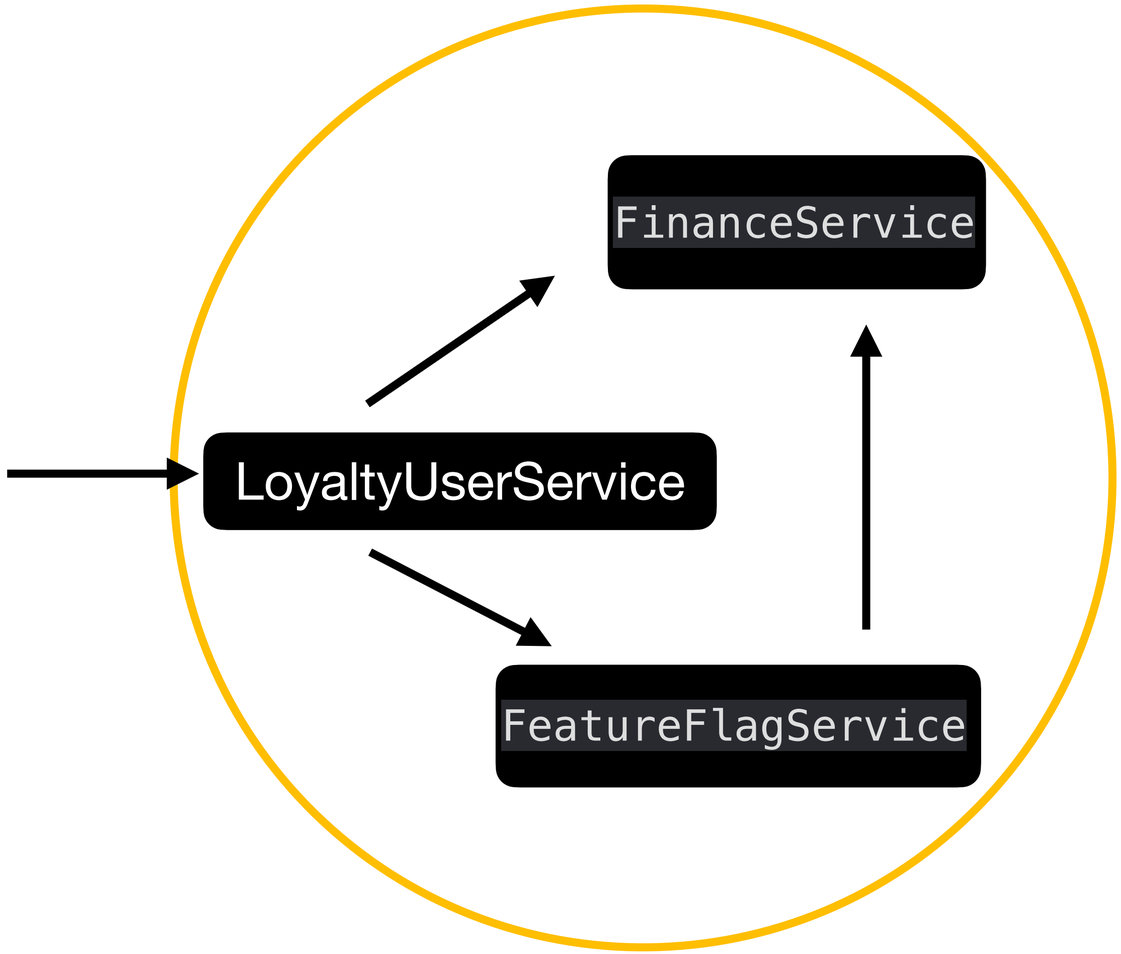


Id	Label	..	module	path
MonolitApp.FinanceService	FinanceService		MonolitApp	/Users/rofle100lvl/Desktop/DD/MonolitApp/Monolit/Finance/FinanceService.swift
MonolitApp.FeatureFlagService	FeatureFlagService		MonolitApp	/Users/rofle100lvl/Desktop/DD/MonolitApp/Monolit/FeatureFlags/FeatureFlagService.swift
MonolitApp.rApp	rApp		MonolitApp	/Users/rofle100lvl/Desktop/DD/MonolitApp/Monolit/View.swift
MonolitApp.LoyaltyUserService	LoyaltyUserService		MonolitApp	/Users/rofle100lvl/Desktop/DD/MonolitApp/Monolit/Loyalty/LoyaltyUserService.swift

Вспоминаем какие данные у нас есть

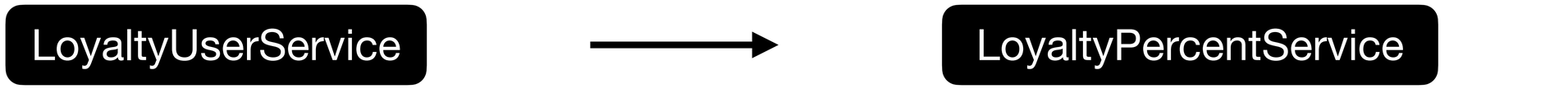
Анализ графа

Решение



Анализ графа

Решение

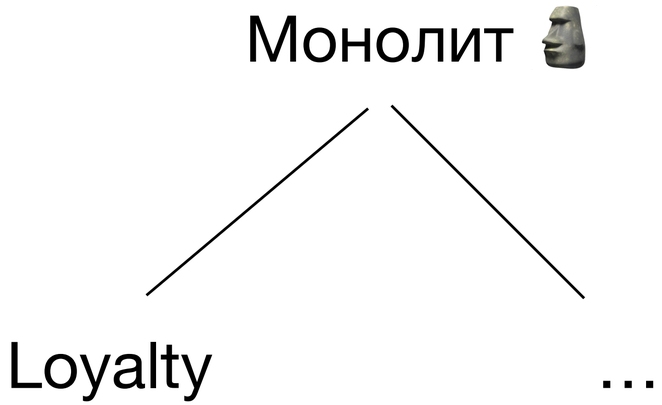


Monolit/Loyalty/LoyaltyUserService.swift

Monolit/Loyalty/LoyaltyPercentService.swift

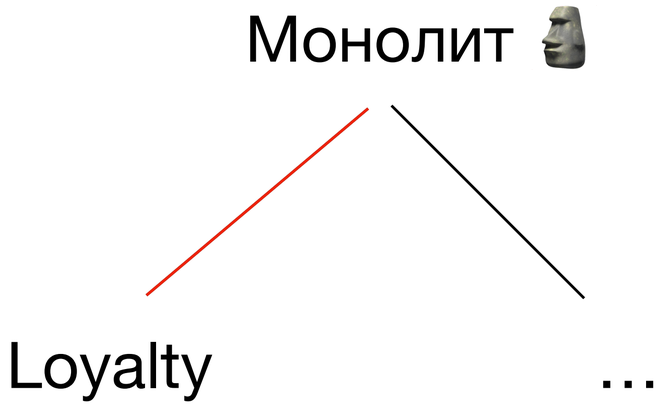
Анализ графа

Решение



Анализ графа

Решение



Анализ графа

Решение

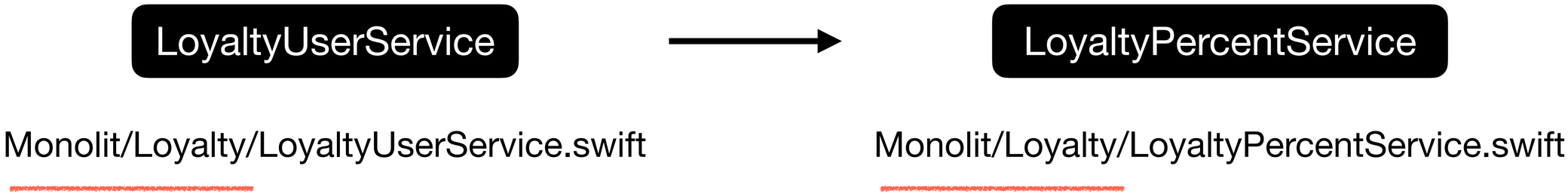
Loyalty

Монолит 🗿

...

Анализ графа

Решение



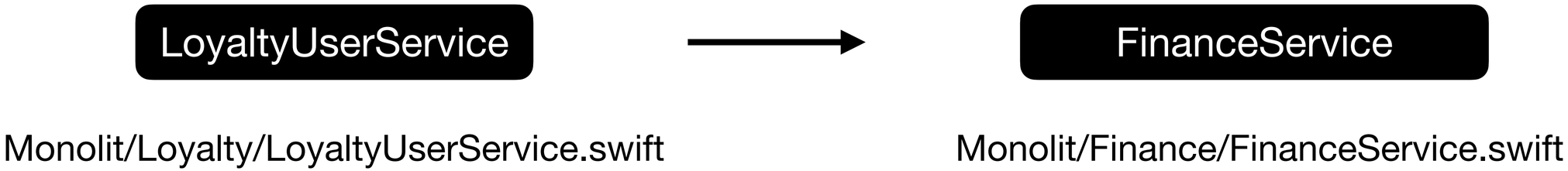
Анализ графа

Решение

Замечание 1: Если Source, Target $\in$ папке Loyalty - кейс не интересный

Анализ графа

Решение

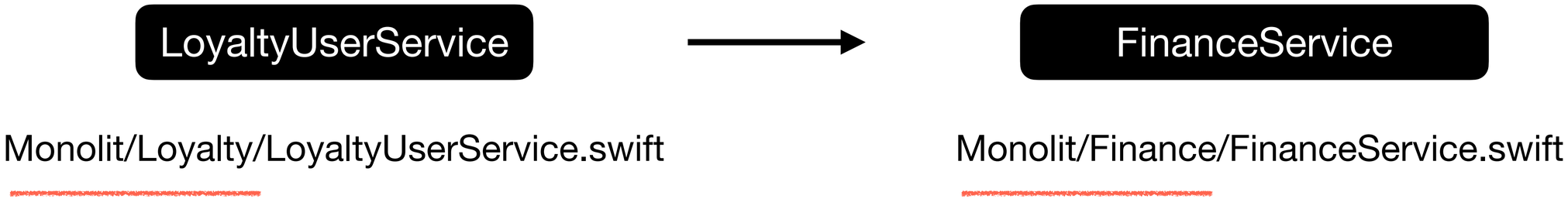


Monolit/Loyalty/LoyaltyUserService.swift

Monolit/Finance/FinanceService.swift

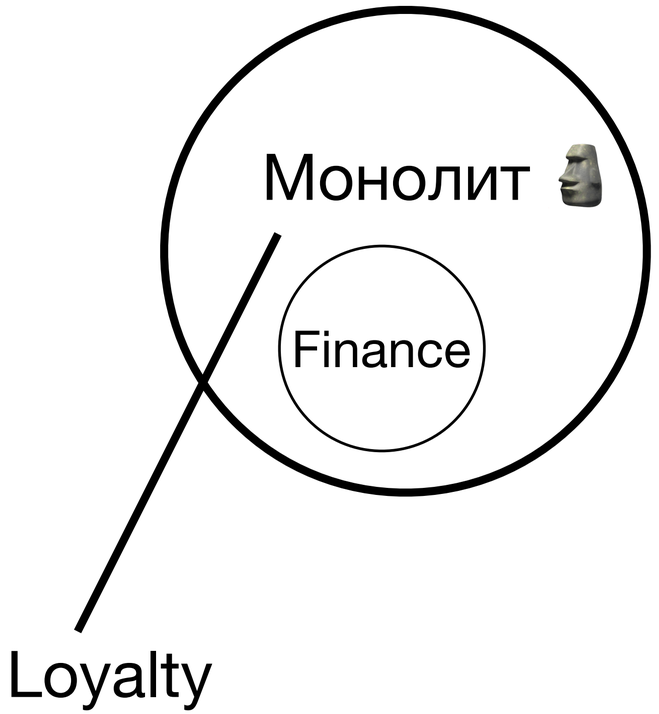
Анализ графа

Решение



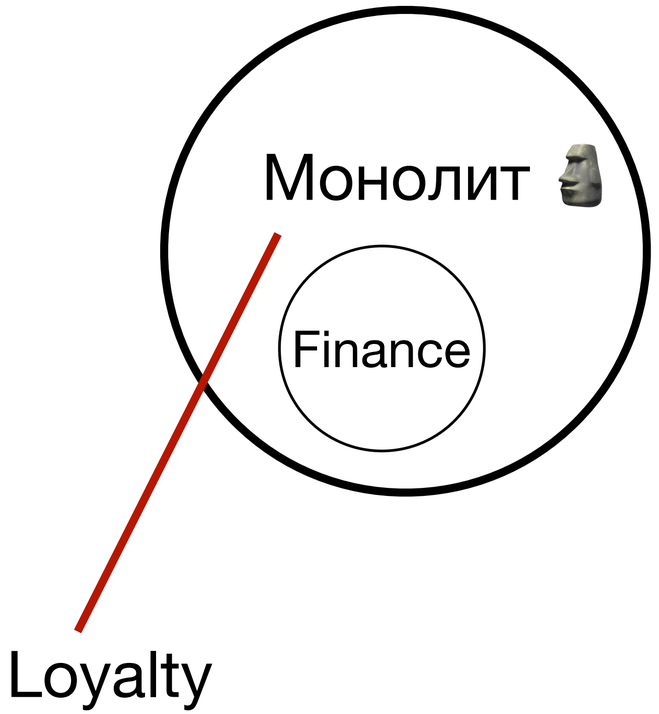
Анализ графа

Решение



Анализ графа

Решение



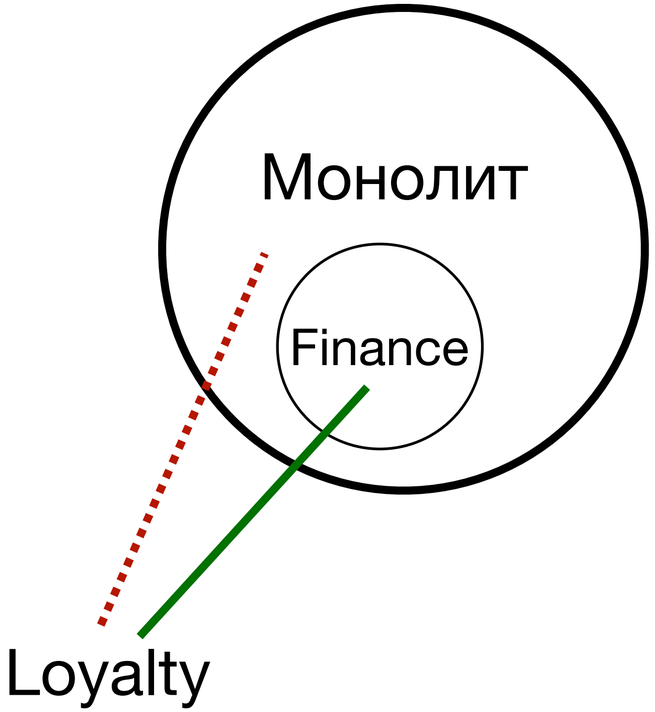
Анализ графа

Решение

Loyalty

Анализ графа

Решение



Анализ графа

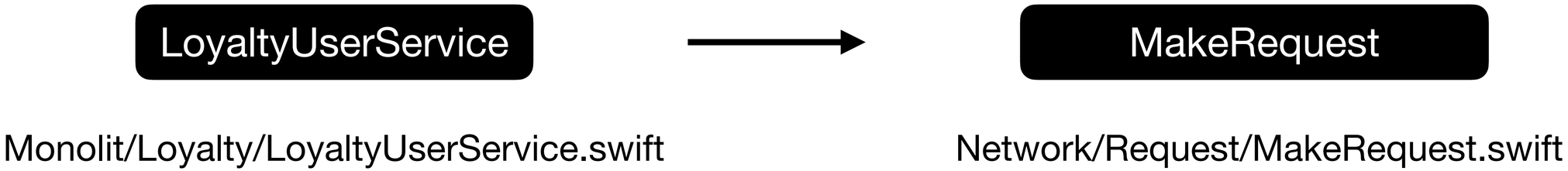
Решение

Замечание 1: Если Source, Target $\in$ папке Loyalty - кейс не интересный

Замечание 2: Если Source $\in$ папке Loyalty, Target $\notin$ Loyalty - кейс интересный

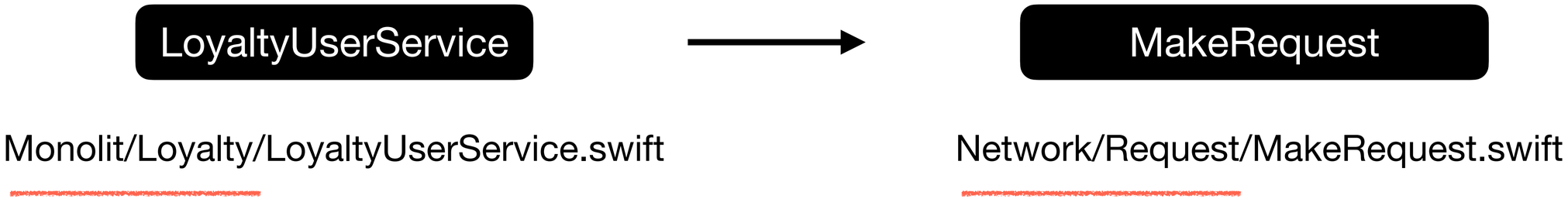
Анализ графа

Решение



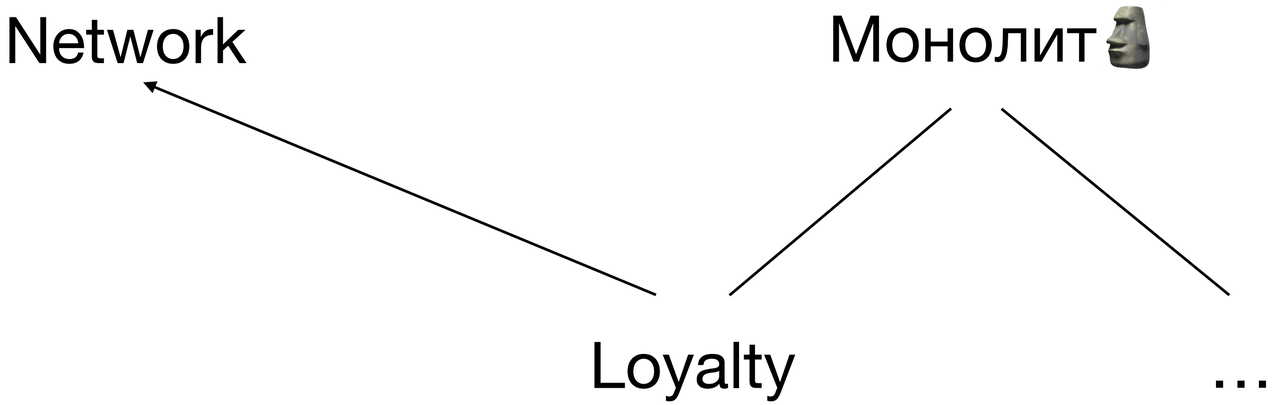
Анализ графа

Решение



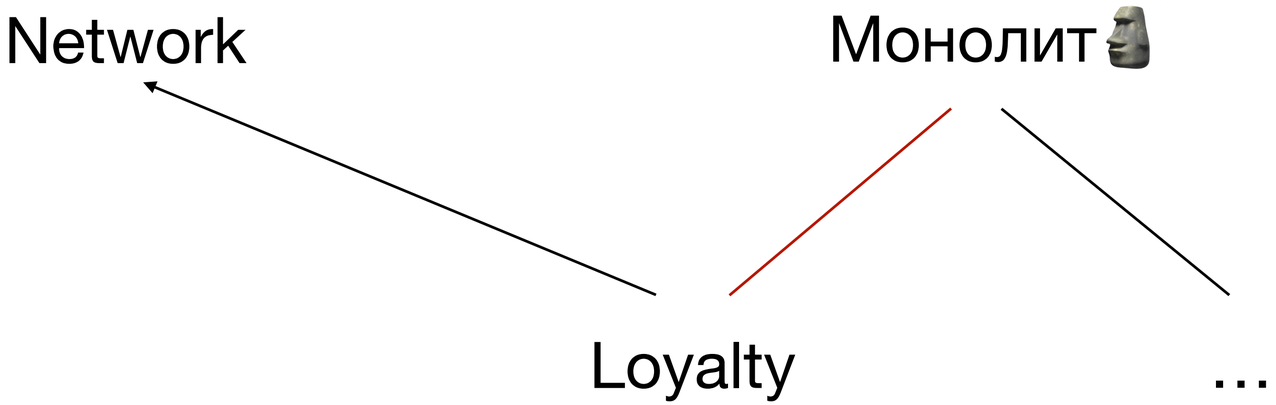
Анализ графа

Решение



Анализ графа

Решение



Анализ графа

Решение

Network

Loyalty

Монолит🗿

|

…

Анализ графа

Решение

Замечание 1: Если Source, Target ∈ папке Loyalty - кейс 👎

Замечание 2: Если Source ∈ папке Loyalty, Target ∉ Loyalty - кейс 👍

Замечание 3: Если Source ∈ Monolit, Target ∉ Monolit - кейс 👎

Анализ графа

Решение

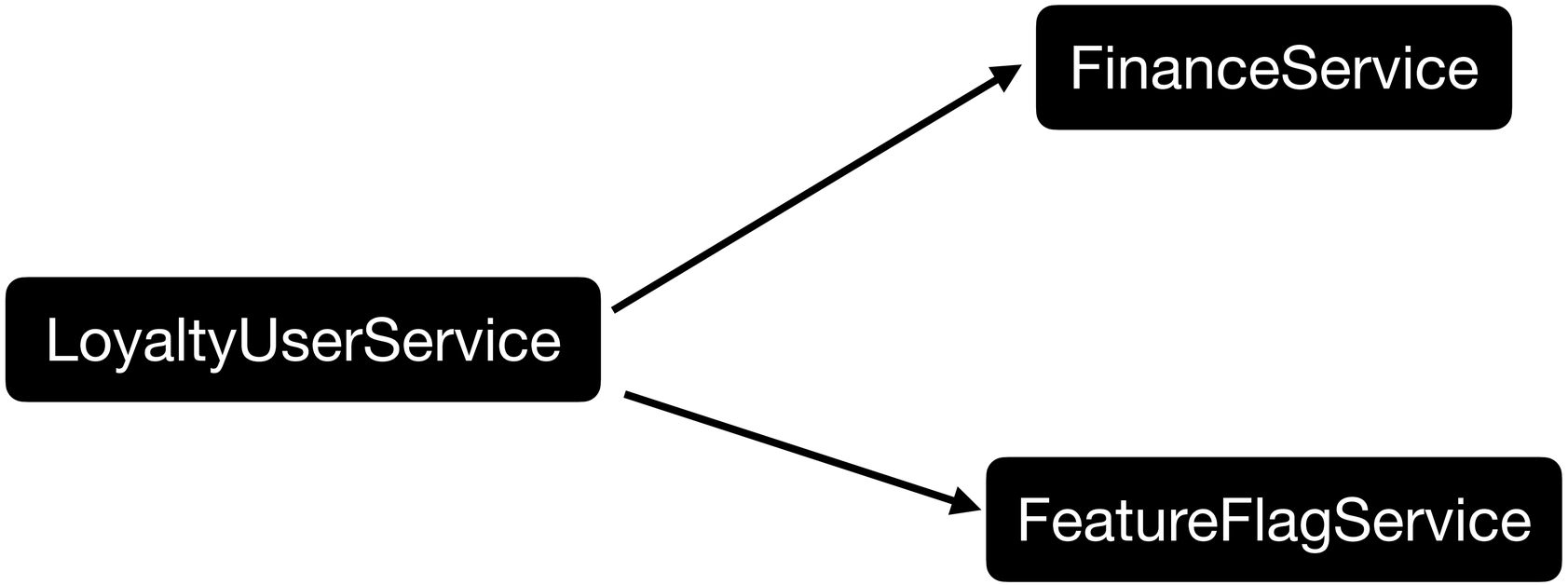
Фильтр 1: Source $\in$ папке Loyalty

Фильтр 2: Target $\notin$ Loyalty

Фильтр 3: Source, Target $\in$ Monolit

Анализ графа

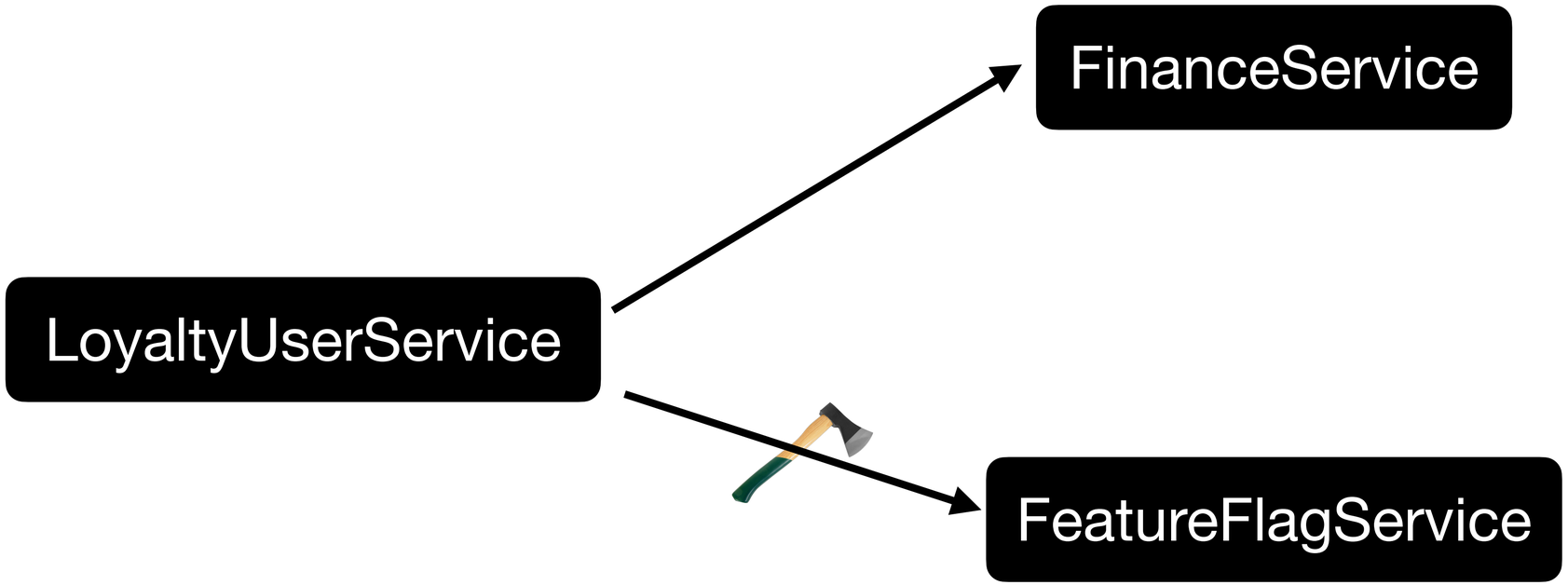
Что мы получили?



Граф, в котором каждое ребро - это спагетти мешающее модуляризировать эту папку

Анализ графа

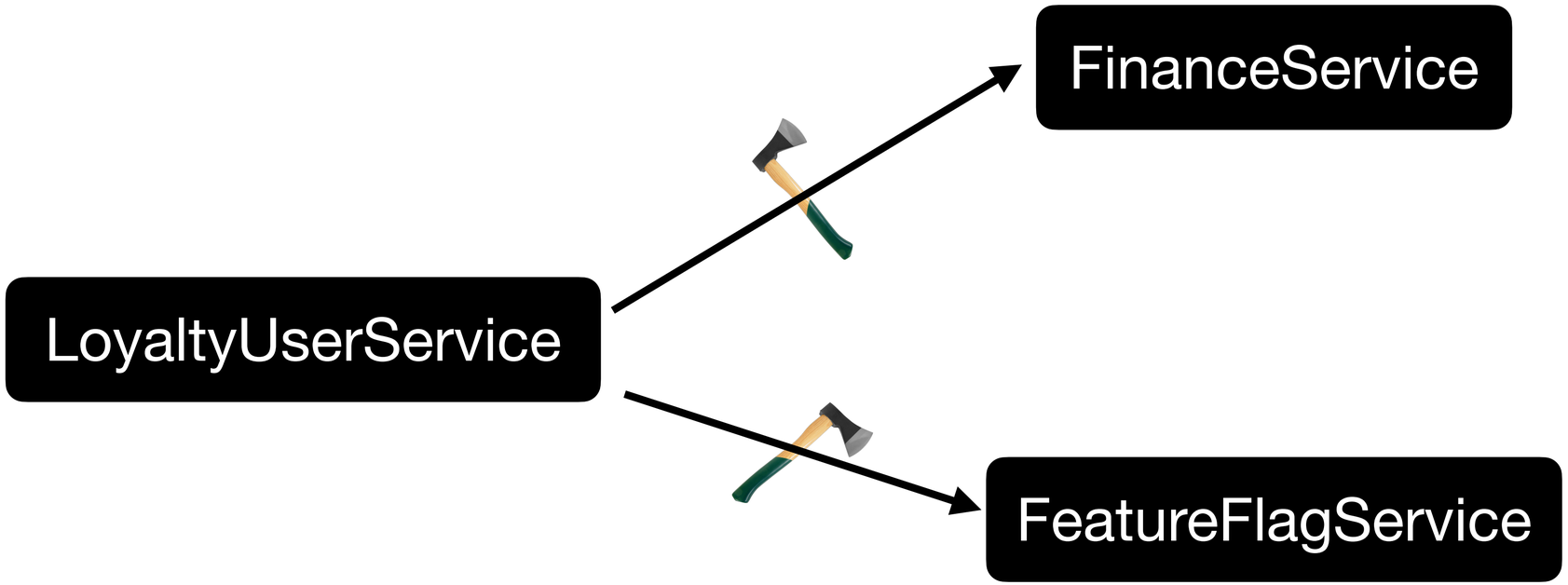
Что мы получили?



Граф, в котором каждое ребро - это спагетти мешающее модуляризировать эту папку

Анализ графа

Что мы получили?



Граф, в котором каждое ребро - это спагетти мешающее модуляризировать эту папку

Анализ графа

Что есть разрубить зависимость?

Анализ графа

Что есть разрубить зависимость?

Вынесение под протокол

Анализ графа

Что есть разрубить зависимость?

Вынесение под протокол

Вынесение в 3-ий модуль

Анализ графа

Пример с протоколом

Было

```
1  struct HotelReviewsContainer {
2    private var navigationState: RootNavigationState
3
4    func someFunc() {
5      navigationState.routes.presentCover(.slider(sliderInfo))
6    }
7  }
```

Стало

```
1  protocol HotelReviewsNavigator {
2    func presentSliderCover(sliderInfo: SliderInfo)
3  }
4
5  struct HotelReviewsContainer {
6    private var navigator: HotelReviewsNavigator
7
8    func someFunc() {
9      navigator.presentSliderCover(sliderInfo)
10 }
```

Анализ графа

Общая сущность

```
1  struct OrderIds {
2    let orderId: String
3  }
```

Анализ графа

Общая сущность

```
1  struct OrderIds {
2    let orderId: String
3  }
```

В таком случае выносим ее
в 3-ий модуль и импортируем

Сколько SP?

У нас X ребер. X / 3 SP будет стоить

Сколько SP?

Глава 3. Динамический анализ

Динамический анализ

Казалось бы, все?

Динамический анализ

Казалось бы, все?

Динамический анализ

Не все так радужно

Динамический анализ

Не все так радужно

Проблема #1

Прорастают не все зависимости

Динамический анализ

Проблема #1 Прорастают не все зависимости

```swift
// где-то в коде есть
enum Status {
  case pending
  case loading
}
func someFuncWithStatus(status: Status) {}

// в файле, который мы хотим вынести
func createStatus() {
  someFuncWithStatus(status: .pending)
}
```


Никак не можем
узнать, что за pending используется

Динамический анализ

Проблема #1 Прорастают не все зависимости

```
1  // где-то в коде есть
2  struct Status {}
3
4  func someFuncWithStatus(status: Status)
5
6  // в файле, который мы хотим вынести
7  func createStatus() {
8    someFuncWithStatus(status: .init())
9  }
```

Никак не можем
узнать, что за .init() используется

Динамический анализ

Проблема #1 Прорастают не все зависимости

```
1   func createStatus() {
2     someClass.someFuncWithStatus(status: 5)
3   }
```

Никак не можем узнать, что за функция используется

Динамический анализ

Что такое?

Способ анализа программы непосредственно
при её выполнении

Динамический анализ

Periphery

Динамический анализ

Periphery

Periphery - утилита для поиска не используемого кода в проекте

Динамический анализ

Как Periphery это делает?

Динамический анализ

Как Periphery это делает?

```
periphery scan

* Inspecting project...
* Building App...
```

При запуске periphery вы увидите

Динамический анализ

Как Periphery это делает?

```
periphery scan

* Inspecting project...
* Building App...
```

При запуске periphery вы увидите

Динамический анализ

Команда для билда у Periphery

```
xcodebuild -workspace \’…Travel.xcworkspace\’ -scheme \'App\' -parallelizeTargets
-derivedDataPath \’…DerivedData-a7a2af41-d4167733-b87d646\’ -quiet build
CODE_SIGNING_ALLOWED=\"NO\" ENABLE_BITCODE=\"NO\"
DEBUG_INFORMATION_FORMAT=\"dwarf\"
COMPILER_INDEX_STORE_ENABLE=\"YES\" INDEX_ENABLE_DATA_STORE=\"YES\"
```

Динамический анализ

Команда для билда у Periphery

```
xcodebuild -workspace \’…Travel.xcworkspace\’ -scheme \'App\' -parallelizeTargets
-derivedDataPath \’…DerivedData-a7a2af41-d4167733-b87d646\’ -quiet build
CODE_SIGNING_ALLOWED=\"NO\" ENABLE_BITCODE=\"NO\"
DEBUG_INFORMATION_FORMAT=\"dwarf\"
COMPILER_INDEX_STORE_ENABLE=\"YES\" INDEX_ENABLE_DATA_STORE=\"YES\"
```

Динамический анализ

Команда для билда у Periphery

```
xcodebuild -workspace \’…Travel.xcworkspace\’ -scheme \'App\' -parallelizeTargets
-derivedDataPath \’…DerivedData-a7a2af41-d4167733-b87d646\’ -quiet build
CODE_SIGNING_ALLOWED=\"NO\" ENABLE_BITCODE=\"NO\"
DEBUG_INFORMATION_FORMAT=\"dwarf\"
COMPILER_INDEX_STORE_ENABLE=\"YES\" INDEX_ENABLE_DATA_STORE=\"YES\"
```

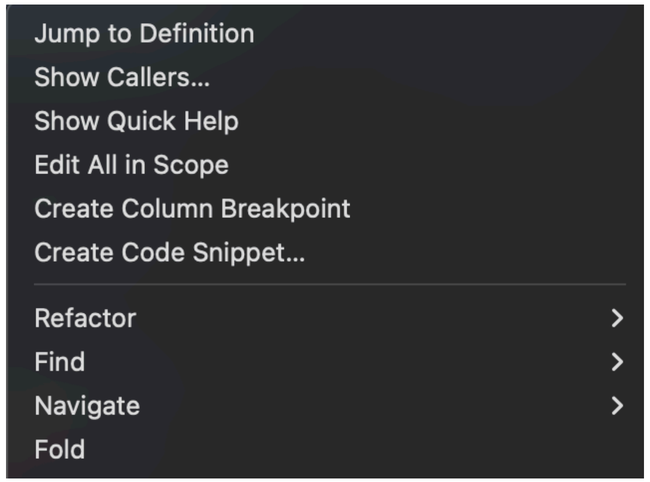
Динамический анализ

COMPILER_INDEX_STORE_ENABLE

Документация Apple

Динамический анализ

Фичи работающие благодаря Index Store



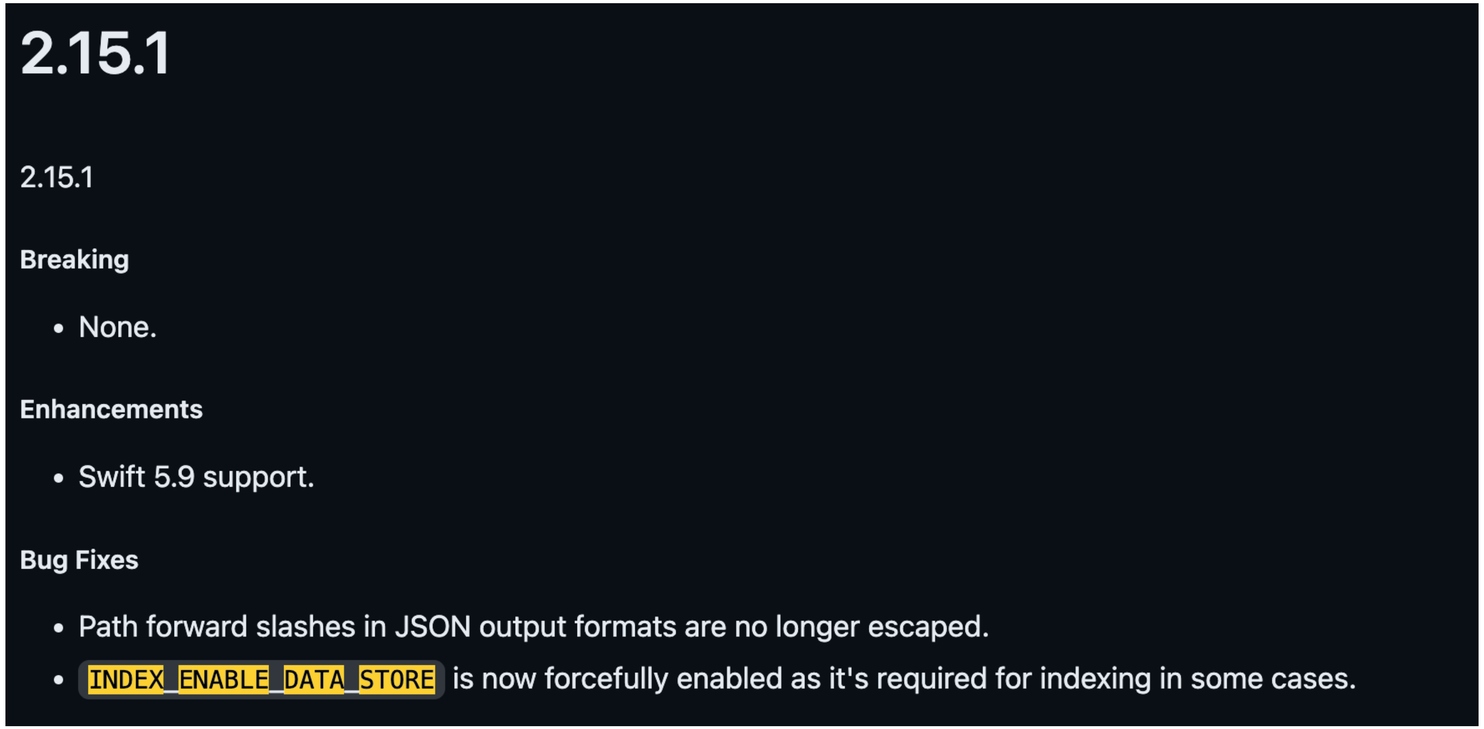
Динамический анализ

Что же касается второго флага?

INDEX_ENABLE_DATA_STORE=\"YES\"

Динамический анализ

Единственное упоминание



2.15.1

2.15.1

Breaking

- None.

Enhancements

- Swift 5.9 support.

Bug Fixes

- Path forward slashes in JSON output formats are no longer escaped.
- `INDEX_ENABLE_DATA_STORE` is now forcefully enabled as it's required for indexing in some cases.

Динамический анализ

Перенесемся в 2017 год…

Динамический анализ

Перенесемся в 2017 год…

Динамический анализ

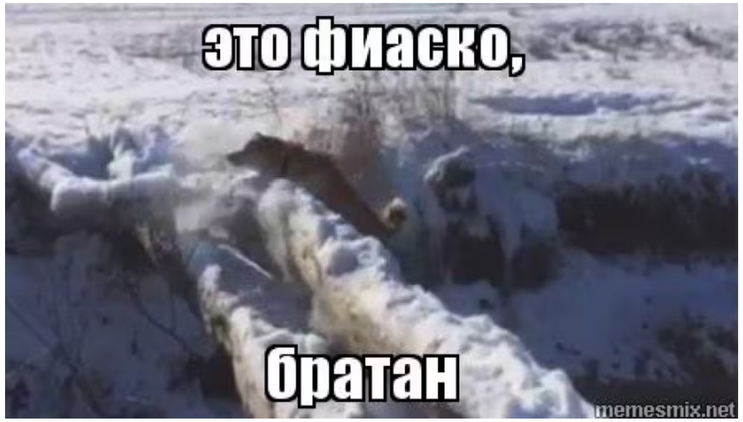
Перенесемся в 2017 год…

Динамический анализ

Перенесемся в 2017 год…

Динамический анализ

Перенесемся в 2017 год…

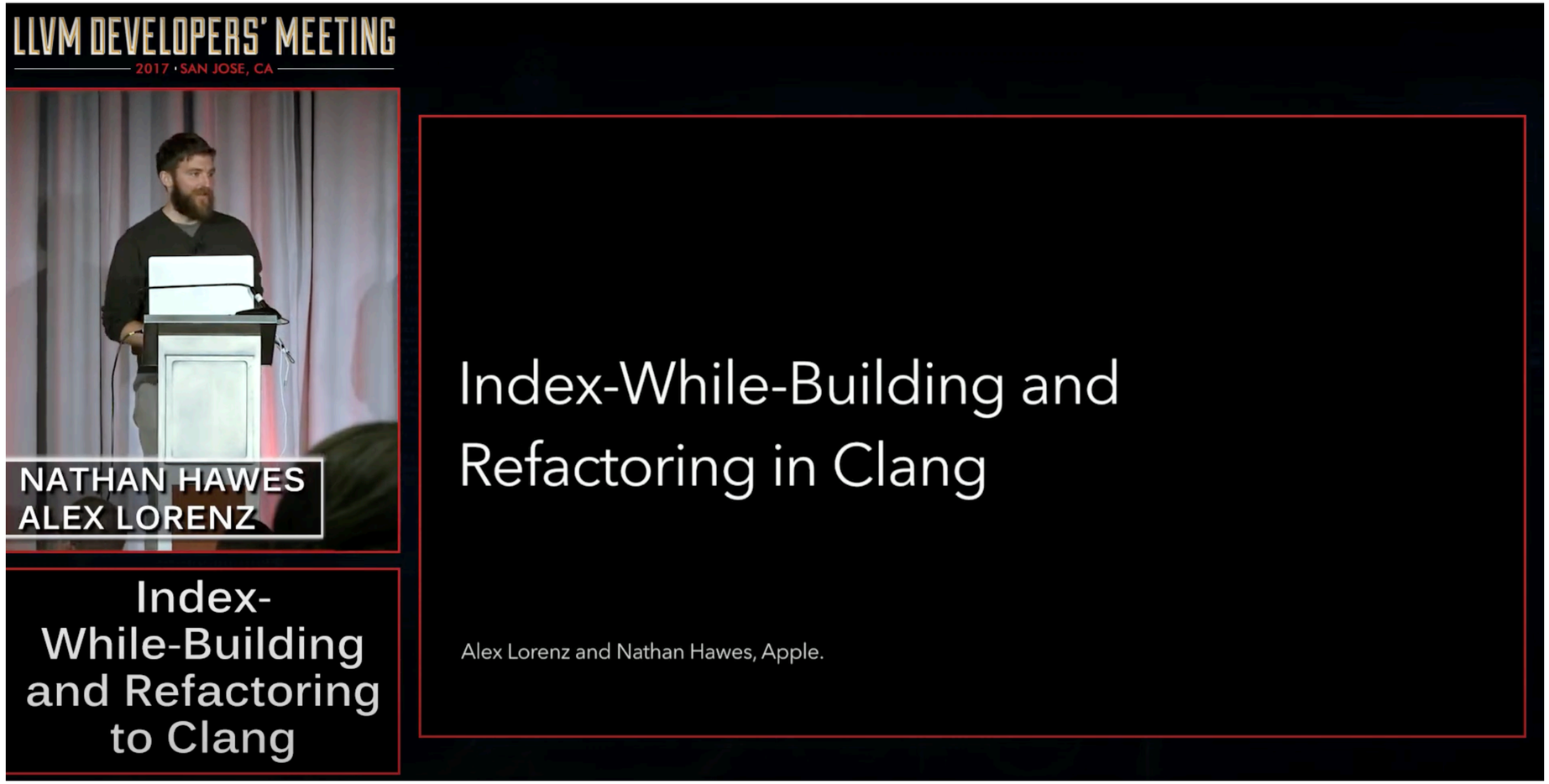


Динамический анализ

Xcode 8.0 beta

Динамический анализ

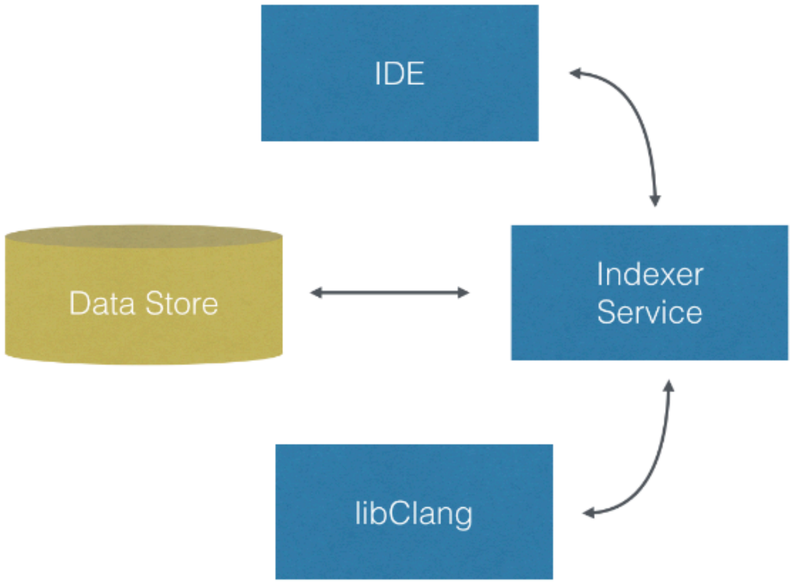
Таинственный доклад на LLVM конференции



1.6k просмотров на ютубе

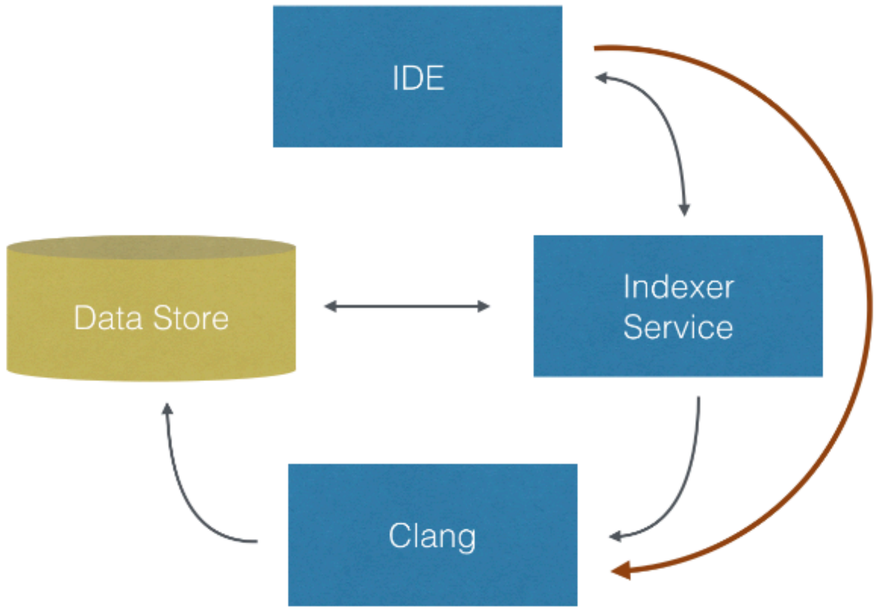
Динамический анализ

Индексация до Xcode 9



Динамический анализ

Индексация в Xcode 9 и после



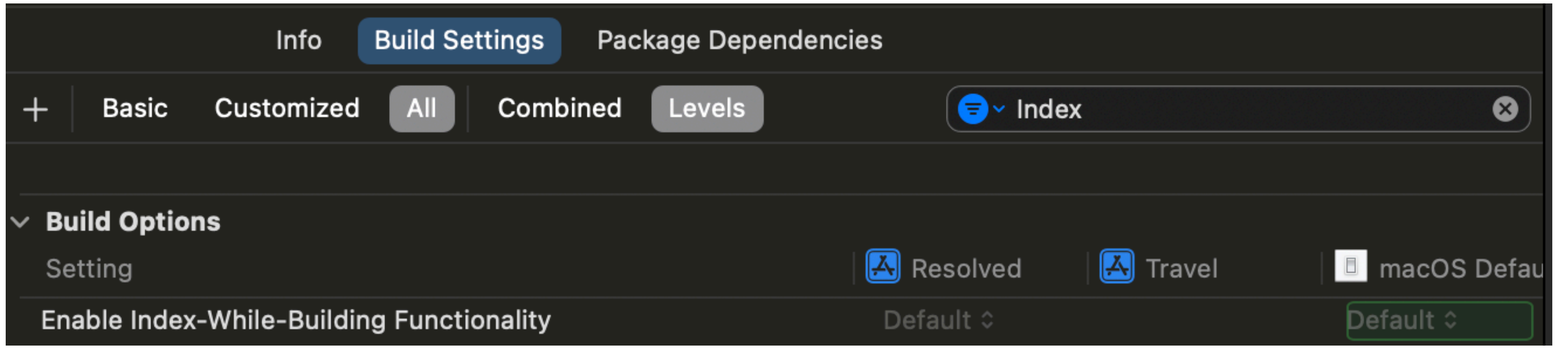
Динамический анализ

Index while building

+

Динамический анализ

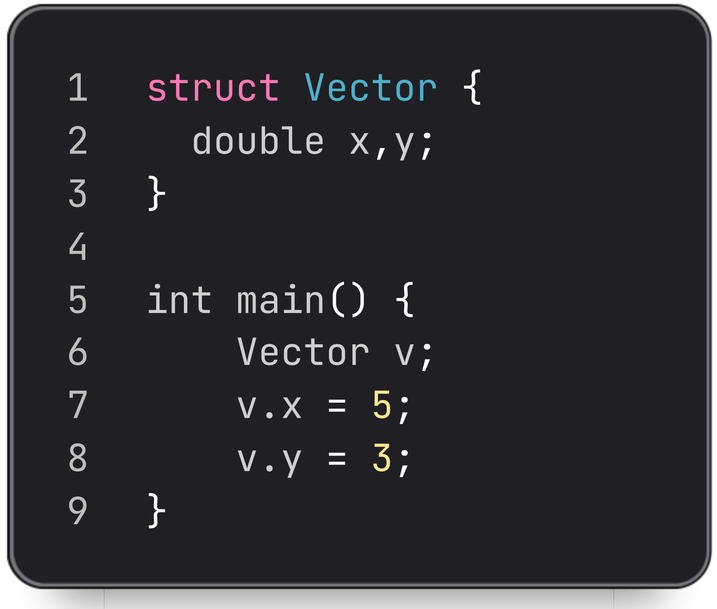
Index while building в Xcode



Index-While-Building в настройках проекта

Динамический анализ

Пример CodeExample.cpp



```cpp
struct Vector {
  double x,y;
}

int main() {
    Vector v;
    v.x = 5;
    v.y = 3;
}
```

CodeExample.cpp

Динамический анализ

Пример CodeExample.cpp

```
clang -index-store-path ~/Desktop/test/tmp Vector.cpp
```

Динамический анализ

Пример CodeExample.cpp

```
clang -index-store-path ~/Desktop/test/tmp Vector.cpp
```

Динамический анализ

Пример CodeExample.cpp

Структура папки tmp

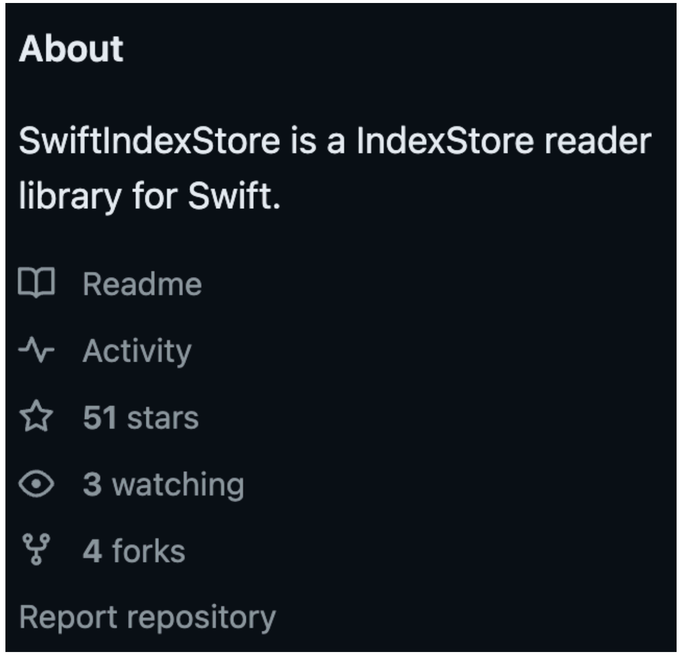
Динамический анализ

Пример CodeExample.cpp

Vector.cpp-2QBJE16P8S8II

Динамический анализ

SwiftIndexStore - читаем индексы



Динамический анализ

Пример CodeExample.cpp

```
git clone https://github.com/kateinoigakukun/swift-indexstore.git
cd swift-indexstore
swift run index-dump-tool print-record --index-store-path ../tmp/
```

Динамический анализ

Пример CodeExample.cpp

```
swift run index-dump-tool print-record --index-store-path ../tmp
Building for debugging...
[1/1] Write swift-version-6044DDE57671499D.txt
Build complete! (0.15s)
============================
Unit: "Vector-371411.o-2L6E8Q5T5XLBM"
----------------------------
Record: "/Users/rofle100lvl/Desktop/test/Vector.cpp"
----------Symbols----------
| usr = c:@S@Vector | name = Vector | kind = struct | subKind = none | language = c |
| usr = c:@S@Vector@FI@x | name = x | kind = field | subKind = none | language = c |
| usr = c:@S@Vector@FI@y | name = y | kind = field | subKind = none | language = c |
| usr = c:@F@main# | name = main | kind = function | subKind = none | language = c |
----------Occurrences----------
| roles = Roles(["definition"]) | usr = c:@S@Vector | location = /Users/rofle100lvl/Desktop/test/Vector.cpp:1:8 |
| roles = Roles(["definition", "childOf"]) | usr = c:@S@Vector@FI@x | location = /Users/rofle100lvl/Desktop/test/Vector.cpp:2:9 |
| roles = Roles(["definition", "childOf"]) | usr = c:@S@Vector@FI@y | location = /Users/rofle100lvl/Desktop/test/Vector.cpp:2:11 |
| roles = Roles(["definition"]) | usr = c:@F@main# | location = /Users/rofle100lvl/Desktop/test/Vector.cpp:5:5 |
| roles = Roles(["reference", "containedBy"]) | usr = c:@S@Vector | location = /Users/rofle100lvl/Desktop/test/Vector.cpp:7:5 |
| roles = Roles(["reference", "write", "containedBy"]) | usr = c:@S@Vector@FI@x | location = /Users/rofle100lvl/Desktop/test/Vector.cpp:8:7 |
| roles = Roles(["reference", "write", "containedBy"]) | usr = c:@S@Vector@FI@y | location = /Users/rofle100lvl/Desktop/test/Vector.cpp:9:7 |
```

Динамический анализ

Пример CodeExample.cpp Symbol

usr	name	kind	subKind	language
c:@S@Vector	Vector	struct	none	c
c:@S@Vector@FI@x	x	field	none	c
c:@S@Vector@FI@y	y	field	none	c
c:@F@main#	main	function	none	c

Динамический анализ

Пример CodeExample.cpp Occurrences

usr	roles	additional	location
c:@S@Vector	definition		…/Vector.cpp:1:8
c:@S@Vector@FI@x	definition	childOf	…/Vector.cpp:2:9
c:@S@Vector@FI@y	definition	childOf	…/Vector.cpp:2:11
c:@F@main^F	definition		…/Vector.cpp:5:5
c:@S@Vector	reference	containedBy	…/Vector.cpp:7:5
c:@S@Vector@FI@x	reference	containedBy,write	…/Vector.cpp:8:7
c:@S@Vector@FI@y	reference	containedBy,write	…/Vector.cpp:9:7

Динамический анализ

Пример Vector.swift

```swift
struct Vector {
  var x: Double
  var y: Double
}

func main() {
  var v = Vector(x: 0, y: 0)
  v.x = 5
  v.y = 3
}
```

Динамический анализ

Пример Vector.swift

```
swiftc -index-store-path ~/Desktop/test/tmp Vector.swift
```

Динамический анализ

Пример Vector.swift

```
tree tmp
tmp
└── v5
    ├── records
    │   ├── 19
    │   │   └── arm64e-apple-macos.swiftinterface_Math-1XV1PJVF5ZT19
    │   ├── 1F
    │   │   └── arm64e-apple-macos.swiftinterface_String-I8090IL041F
    │   ├── 1J
    │   │   └── arm64e-apple-macos.swiftinterface_Misc-18WQ0HLTMHU1J
    │   ├── 34
    │   │   └── arm64e-apple-macos.swiftinterface_Collection-3LDRFETQC4534
    │   ├── 35
    │   │   └── arm64e-apple-macos.swiftinterface_Collection_HashedCollections-X1RM6KOM2435
    │   ├── 6M
    │   │   └── arm64e-apple-macos.swiftinterface_Pointer-3R033K09WUS6M
    │   ├── 6Y
    │   │   └── arm64e-apple-macos.swiftinterface_Math_Vector-HD98BLXTX66Y
    │   ├── 79
    │   │   └── arm64e-apple-macos.swiftinterface_Reflection-23WEYYATJ6M79
    │   ├── 85
    │   │   └── arm64e-apple-macos.swiftinterface_Result-196P5TZD7SO85
    │   ├── D8
    │   │   └── arm64e-apple-macos.swiftinterface_Protocols-1JR191BQ51XD8
    │   ├── DD
    │   │   └── arm64e-apple-macos.swiftinterface_Bool-3EU87B9YY44DD
    │   ├── DZ
    │   │   └── arm64e-apple-macos.swiftinterface_Assert-QC2TV6Y48ADZ
    │   ├── G1
    │   │   └── arm64e-apple-macos.swiftinterface_KeyPaths-OUD75WWHZGG1
    │   ├── GA
    │   │   └── Vector.swift-10OVAM0LV4JGA
    │   ├── H8
    │   │   └── arm64e-apple-macos.swiftinterface-3GW4HX7FVO5H8
    │   ├── I4
    │   │   └── arm64e-apple-macos.swiftinterface-11BDAXDMMKNI4
    │   ├── JF
    │   │   └── arm64e-apple-macos.swiftinterface_Math_Integers-2DSE6LQZ2DTJF
    │   ├── JZ
    │   │   └── arm64e-apple-macos.swiftinterface_Playground-2L1EQDSAOP7JZ
    │   ├── MC
    │   │   └── arm64e-apple-macos.swiftinterface_C-1FNQ7LD38MOMC
    │   ├── NW
    │   │   └── arm64e-apple-macos.swiftinterface_Collection_Type-erased-1MK0RNO1A5BNW
    │   ├── OF
    │   │   └── _SwiftConcurrency.h-2FG2M9GJFDOOF
    │   ├── RU
    │   │   └── arm64e-apple-macos.swiftinterface_Collection_Array-10EIHNNGSBGRU
    │   ├── UN
    │   │   └── arm64e-apple-macos.swiftinterface_Optional-2O9Y4AX0UEMUN
    │   ├── XJ
    │   │   └── arm64e-apple-macos.swiftinterface-2UH7QT48E3VXJ
    │   ├── XT
    │   │   └── arm64e-apple-macos.swiftinterface_Math_Floating-J539T6A8RKXT
    │   ├── Y7
    │   │   └── arm64e-apple-macos.swiftinterface_Hashing-2HET5KICG9OY7
    │   └── ZL
    │       └── arm64e-apple-macos.swiftinterface_Collection_Lazy_Views-3SIDA942HC6ZL
    └── units
        ├── Vector-1.o-0XAJLZ2H712C
        ├── _SwiftConcurrencyShims-3V0QEGOKRK7C8.pcm-1LZGA9P7T2KUE
        ├── arm64e-apple-macos.swiftinterface-23VM4QBUDURI2
        ├── arm64e-apple-macos.swiftinterface-2KXP38Z76D24X
        └── arm64e-apple-macos.swiftinterface-647VWWA3H8HC

31 directories, 32 files
```

Динамический анализ

Пример Vector.swift

```
tree tmp
tmp
└── v5
    ├── records
    │   ├── 19
    │   │   └── arm64e-apple-macos.swiftinterface_Math-1XV1PJVF5ZT19
    │   ├── 1F
    │   │   └── arm64e-apple-macos.swiftinterface_String-I8090IL041F
    │   ├── 1J
    │   │   └── arm64e-apple-macos.swiftinterface_Misc-18WQ0HLTMHU1J
    │   ├── 34
    │   │   └── arm64e-apple-macos.swiftinterface_Collection-3LDRFETQC4534
    │   ├── 35
    │   │   └── arm64e-apple-macos.swiftinterface_Collection_HashedCollections-X1RM6KOM2435
    │   ├── 6M
    │   │   └── arm64e-apple-macos.swiftinterface_Pointer-3R033K09WUS6M
    │   ├── 6Y
    │   │   └── arm64e-apple-macos.swiftinterface_Math_Vector-HD98BLXTX66Y
    │   ├── 79
    │   │   └── arm64e-apple-macos.swiftinterface_Reflection-23WEYYATJ6M79
    │   ├── 85
    │   │   └── arm64e-apple-macos.swiftinterface_Result-196P5TZD7SO85
    │   ├── D8
    │   │   └── arm64e-apple-macos.swiftinterface_Protocols-1JR191BQ51XD8
    │   ├── DD
    │   │   └── arm64e-apple-macos.swiftinterface_Bool-3EU87B9YY44DD
    │   ├── DZ
    │   │   └── arm64e-apple-macos.swiftinterface_Assert-QC2TV6Y48ADZ
    │   ├── G1
    │   │   └── arm64e-apple-macos.swiftinterface_KeyPaths-OUD75WWHZGG1
    │   ├── GA
    │   │   └── Vector.swift-10OVAM0LV4JGA
    │   ├── H8
    │   │   └── arm64e-apple-macos.swiftinterface-3GW4HX7FVO5H8
    │   ├── I4
    │   │   └── arm64e-apple-macos.swiftinterface-11BDAXDMMKNI4
    │   ├── JF
    │   │   └── arm64e-apple-macos.swiftinterface_Math_Integers-2DSE6LQZ2DTJF
    │   ├── JZ
    │   │   └── arm64e-apple-macos.swiftinterface_Playground-2L1EQDSAOP7JZ
    │   ├── MC
    │   │   └── arm64e-apple-macos.swiftinterface_C-1FNQ7LD38MOMC
    │   ├── NW
    │   │   └── arm64e-apple-macos.swiftinterface_Collection_Type-erased-1MK0RNO1A5BNW
    │   ├── OF
    │   │   └── _SwiftConcurrency.h-2FG2M9GJFDOOF
    │   ├── RU
    │   │   └── arm64e-apple-macos.swiftinterface_Collection_Array-10EIHNNGSBGRU
    │   ├── UN
    │   │   └── arm64e-apple-macos.swiftinterface_Optional-2O9Y4AX0UEMUN
    │   ├── XJ
    │   │   └── arm64e-apple-macos.swiftinterface-2UH7QT48E3VXJ
    │   ├── XT
    │   │   └── arm64e-apple-macos.swiftinterface_Math_Floating-J539T6A8RKXT
    │   ├── Y7
    │   │   └── arm64e-apple-macos.swiftinterface_Hashing-2HET5KICG9OY7
    │   └── ZL
    │       └── arm64e-apple-macos.swiftinterface_Collection_Lazy_Views-3SIDA942HC6ZL
    └── units
        ├── Vector-1.o-0XAJLZ2H712C
        ├── _SwiftConcurrencyShims-3V0QEG0KRK7C8.pcm-1LZGA9P7T2KUE
        ├── arm64e-apple-macos.swiftinterface-23VM4QBUDURI2
        ├── arm64e-apple-macos.swiftinterface-2KXP38Z76D24X
        └── arm64e-apple-macos.swiftinterface-647VWWA3H8HC

31 directories, 32 files
```

Знакомьтесь std lib

Динамический анализ

Пример Vector.swift

```
swift run index-dump-tool print-record --index-store-path ../tmp
```

Динамический анализ

Пример Vector.swift

```
------------------------------
Record: "/Users/rofle100lvl/Desktop/test/Vector.swift"
----------Symbols----------
| usr = s:6VectorAAV1x1yABSd_Sdtcfc | name = init(x:y:) | kind = constructor | subKind = none | language = swift |
| usr = s:6VectorAAV | name = Vector | kind = struct | subKind = none | language = swift |
| usr = s:6VectorAAV1xSdvg | name = getter:x | kind = instanceMethod | subKind = accessorGetter | language = swift |
| usr = s:6VectorAAV1xSdvp | name = x | kind = instanceProperty | subKind = none | language = swift |
| usr = s:6VectorAAV1xSdvs | name = setter:x | kind = instanceMethod | subKind = accessorSetter | language = swift |
| usr = s:Sd | name = Double | kind = struct | subKind = none | language = swift |
| usr = s:6VectorAAV1ySdvg | name = getter:y | kind = instanceMethod | subKind = accessorGetter | language = swift |
| usr = s:6VectorAAV1ySdvp | name = y | kind = instanceProperty | subKind = none | language = swift |
| usr = s:6VectorAAV1ySdvs | name = setter:y | kind = instanceMethod | subKind = accessorSetter | language = swift |
| usr = s:6Vector4mainyyF | name = main() | kind = function | subKind = none | language = swift |
| usr = s:6Vector4mainyyF1vL_A2AVvp | name = v | kind = variable | subKind = none | language = swift |
----------Occurrences----------
| roles = Roles(["definition", "implicit", "childOf"]) | usr = s:6VectorAAV1x1yABSd_Sdtcfc | location = /Users/rofle100lvl/Desktop/test/Vector.swift:1:8 |
| roles = Roles(["definition"]) | usr = s:6VectorAAV | location = /Users/rofle100lvl/Desktop/test/Vector.swift:1:8 |
| roles = Roles(["definition", "implicit", "childOf", "accessorOf"]) | usr = s:6VectorAAV1xSdvg | location = /Users/rofle100lvl/Desktop/test/Vector.swift:2:7 |
| roles = Roles(["definition", "implicit", "childOf", "accessorOf"]) | usr = s:6VectorAAV1xSdvs | location = /Users/rofle100lvl/Desktop/test/Vector.swift:2:7 |
| roles = Roles(["definition", "childOf"]) | usr = s:6VectorAAV1xSdvp | location = /Users/rofle100lvl/Desktop/test/Vector.swift:2:7 |
| roles = Roles(["reference", "containedBy"]) | usr = s:Sd | location = /Users/rofle100lvl/Desktop/test/Vector.swift:2:10 |
| roles = Roles(["definition", "implicit", "childOf", "accessorOf"]) | usr = s:6VectorAAV1ySdvg | location = /Users/rofle100lvl/Desktop/test/Vector.swift:3:7 |
| roles = Roles(["definition", "implicit", "childOf", "accessorOf"]) | usr = s:6VectorAAV1ySdvs | location = /Users/rofle100lvl/Desktop/test/Vector.swift:3:7 |
| roles = Roles(["definition", "childOf"]) | usr = s:6VectorAAV1ySdvp | location = /Users/rofle100lvl/Desktop/test/Vector.swift:3:7 |
| roles = Roles(["reference", "containedBy"]) | usr = s:Sd | location = /Users/rofle100lvl/Desktop/test/Vector.swift:3:10 |
| roles = Roles(["definition"]) | usr = s:6Vector4mainyyF | location = /Users/rofle100lvl/Desktop/test/Vector.swift:6:6 |
| roles = Roles(["reference", "containedBy"]) | usr = s:6VectorAAV | location = /Users/rofle100lvl/Desktop/test/Vector.swift:7:13 |
| roles = Roles(["reference", "call", "calledBy", "containedBy"]) | usr = s:6VectorAAV1x1yABSd_Sdtcfc | location = /Users/rofle100lvl/Desktop/test/Vector.swift:7:13 |
| roles = Roles(["reference", "containedBy"]) | usr = s:6VectorAAV1xSdvp | location = /Users/rofle100lvl/Desktop/test/Vector.swift:7:20 |
| roles = Roles(["reference", "containedBy"]) | usr = s:6VectorAAV1ySdvp | location = /Users/rofle100lvl/Desktop/test/Vector.swift:7:26 |
| roles = Roles(["reference", "call", "implicit", "calledBy", "containedBy"]) | usr = s:6VectorAAV1xSdvs | location = /Users/rofle100lvl/Desktop/test/Vector.swift:8:7 |
| roles = Roles(["reference", "write", "containedBy"]) | usr = s:6VectorAAV1xSdvp | location = /Users/rofle100lvl/Desktop/test/Vector.swift:8:7 |
| roles = Roles(["reference", "call", "implicit", "calledBy", "containedBy"]) | usr = s:6VectorAAV1ySdvs | location = /Users/rofle100lvl/Desktop/test/Vector.swift:9:7 |
| roles = Roles(["reference", "write", "containedBy"]) | usr = s:6VectorAAV1ySdvp | location = /Users/rofle100lvl/Desktop/test/Vector.swift:9:7 |
```

Динамический анализ

Пример Vector.swift Symbol

usr	name	kind	subKind	language
5:6VectorAAV1x1yABSd_Sdtcfc	init(x:y:)	constructor		swift
s:6VectorAAV	Vector	struct		swift
s:6VectorAA1xSdvg	Getter:X	instanceMethod	accessorGetter	swift
s:6VectorAA1xSdvp	X	Instance Property		swift
s:6VectorAA1xSdvs	Setter X	instanceMethod	accessorSetter	swift
s:Sd	Double	Struct		swift
s:6VectorAA1ySdvg	Getter Y	instanceMethod	accessorGetter	swift
s:6VectorAA1ySdvp	Y	Instance Property		swift
s:6VectorAA1ySdvs	Setter Y	instanceMethod	accessorSetter	swift
6Vector4mainyyF	main	function		swift
6Vector4mainyyF1vL_A2AVvp	V	variable	none	Swift

```
1   struct Vector {
2     var x: Double
3     var y: Double
4   }
5
6   func main() {
7     var v = Vector(x: 0, y: 0)
8     v.x = 5
9     v.y = 3
10  }
```

Динамический анализ

Пример Vector.swift

usr	name	kind	subKind	language
5:6VectorAAV1x1yABSd_Sdtcfc	init(x:y:)	constructor		swift
s:6VectorAAV	Vector	struct		swift
s:6VectorAA1xSdvg	Getter X	instanceMethod	accessorGetter	swift
s:6VectorAA1xSdvp	X	Instance Property		swift
s:6VectorAA1xSdvs	Setter X	instanceMethod	accessorSetter	swift
s:Sd	Double	Struct		swift
s:6VectorAA1ySdvg	Getter Y	instanceMethod	accessorGetter	swift
s:6VectorAA1ySdvp	Y	Instance Property		swift
s:6VectorAA1ySdvs	Setter Y	instanceMethod	accessorSetter	swift
6Vector4mainyyF	main	function		swift
6Vector4mainyyF1vL_A2AVvp	V	variable	none	Swift

```swift
struct Vector {
  var x: Double
  var y: Double
}

func main() {
  var v = Vector(x: 0, y: 0)
  v.x = 5
  v.y = 3
}
```

Динамический анализ

Пример Vector.swift

usr	name	kind	subKind	language
5:6VectorAAV1x1yABSd_Sdtcfc	init(x:y:)	constructor		swift
s:6VectorAAV	Vector	struct		swift
s:6VectorAA1xSdvg	Getter X	instanceMethod	accessorGetter	swift
s:6VectorAA1xSdvp	X	Instance Property		swift
s:6VectorAA1xSdvs	Setter X	instanceMethod	accessorSetter	swift
s:Sd	Double	Struct		swift
s:6VectorAA1ySdvg	Getter Y	instanceMethod	accessorGetter	swift
s:6VectorAA1ySdvp	Y	Instance Property		swift
s:6VectorAA1ySdvs	Setter Y	instanceMethod	accessorSetter	swift
6Vector4mainyyF	main	function		swift
6Vector4mainyyF1vL_A2AVvp	V	variable	none	swift

```swift
struct Vector {
  var x: Double
  var y: Double
}

func main() {
  var v = Vector(x: 0, y: 0)
  v.x = 5
  v.y = 3
}
```

Динамический анализ

Пример Vector.swift

usr	name	kind	subKind	language
5:6VectorAAV1x1yABSd_Sdtcfc	init(x:y:)	constructor		swift
s:6VectorAAV	Vector	struct		swift
s:6VectorAA1xSdvg	Getter X	instanceMethod	accessorGetter	swift
s:6VectorAA1xSdvp	X	Instance Property		swift
s:6VectorAA1xSdvs	Setter X	instanceMethod	accessorSetter	swift
s:Sd	Double	Struct		swift
s:6VectorAA1ySdvg	Getter Y	instanceMethod	accessorGetter	swift
s:6VectorAA1ySdvp	Y	Instance Property		swift
s:6VectorAA1ySdvs	Setter Y	instanceMethod	accessorSetter	swift
6Vector4mainyyF	main	function		swift
6Vector4mainyyF1vL_A2AVvp	v	variable	none	Swift

```swift
struct Vector {
  var x: Double
  var y: Double
}

func main() {
  var v = Vector(x: 0, y: 0)
  v.x = 5
  v.y = 3
}
```

Динамический анализ

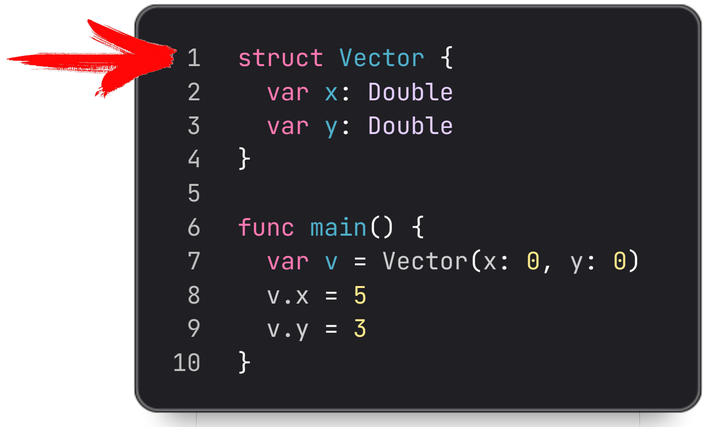
Пример Vector.swift

roles	usr	location
definition, implicit, childOf	s:6VectorAA1x1yABSd_Sdtcfc	Vector.swift:1:8
definition	s:6VectorAAV	Vector.swift:1:8
definition, implicit, childOf, accesorOf	s:6VectorAAV1xSdvg	Vector.swift:2:7
definition, implicit, childOf, accesorOf	s:6VectorAAV1xSdvs	Vector.swift:2:7
definition, childOf	s:6VectorAAV1×Sdvp	Vector.swift:2:7
reference, containedBy	s:Sd	Vector.swift:2:10
definition, implicit, childOf, accesorOf	s:6VectorAAV1ySdvg	Vector.swift:3:7
definition, implicit, childOf, accesorOf	s:6VectorAAV1ySdvs	Vector.swift:3:7
definition, childOf	s:6VectorAAV1ySdvp	Vector.swift:3:7
Reference, containedBy	s:Sd	Vector.swift:3:10
Definition	S:6Vector4mainyyF	Vector.swift:6:6
Reference, containedBy	s:6VectorAAV	Vector.swift:7:13
Reference, Call, calledBy, ContainedBy	s:6VectorAAV1x1yABSd_Sdtcfc	Vector.swift:7:13
Reference, containedBy	s:6VectorAAV1×Sdvp	Vector.swift:7:20
Reference, containedBy	s:6VectorAAV1ySdvp	Vector.swift:7:26
Reference, Call, implicit calledBy, ContainedBy	s:6VectorAAV1xSdvs	Vector.swift:8:7
Reference,write,containedBy	s:6VectorAAV1xSdvp	Vector.swift:8:7
Reference, Call, implicit calledBy, ContainedBy	s:6VectorAAV1ySdvs	Vector.swift:9:7
Reference,write,containedBy	s:6VectorAAV1ySdvp	Vector.swift:9:7

Динамический анализ

Пример Vector.swift

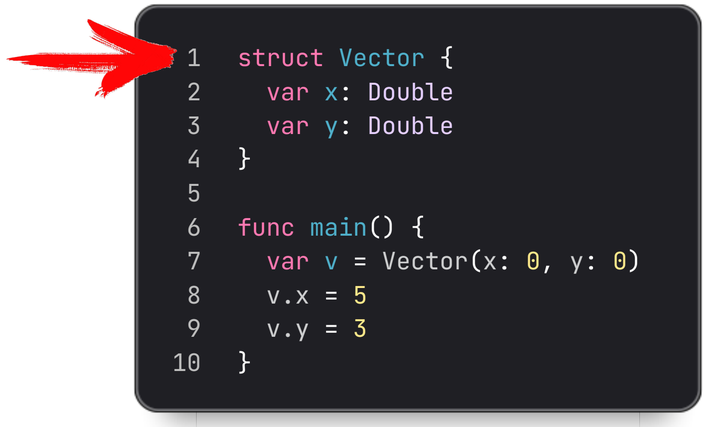
roles	usr	location
definition, implicit, childOf	s:6VectorAA1x1yABSd_Sdtcfc	Vector.swift:1:8
definition	s:6VectorAAV	Vector.swift:1:8
definition, implicit, childOf, accesorOf	s:6VectorAAV1xSdvg	Vector.swift:2:7
definition, implicit, childOf, accesorOf	s:6VectorAAV1xSdvs	Vector.swift:2:7
definition, childOf	s:6VectorAAV1xSdvp	
reference, containedBy	s:Sd	Vector.swift:2:10
definition, implicit, childOf, accesorOf	s:6VectorAAV1ySdvg	Vector.swift:3:7
definition, implicit, childOf, accesorOf	s:6VectorAAV1ySdvs	Vector.swift:3:7
definition, childOf	s:6VectorAAV1ySdvp	Vector.swift:3:7
reference, containedBy	s:Sd	Vector.swift:3:10



Динамический анализ

Пример Vector.swift

roles	usr	location
definition, implicit, childOf	s:6VectorAA1x1yABSd_Sdtcfc	Vector.swift:1:8
definition	s:6VectorAAV	Vector.swift:1:8
definition, implicit, childOf, accesorOf	s:6VectorAAV1xSdvg	Vector.swift:2:7
definition, implicit, childOf, accesorOf	s:6VectorAAV1xSdvs	Vector.swift:2:7
definition, childOf	s:6VectorAAV1×Sdvp	
reference, containedBy	s:Sd	Vector.swift:2:10
definition, implicit, childOf, accesorOf	s:6VectorAAV1ySdvg	Vector.swift:3:7
definition, implicit, childOf, accesorOf	s:6VectorAAV1ySdvs	Vector.swift:3:7
definition, childOf	s:6VectorAAV1ySdvp	Vector.swift:3:7
reference, containedBy	s:Sd	Vector.swift:3:10

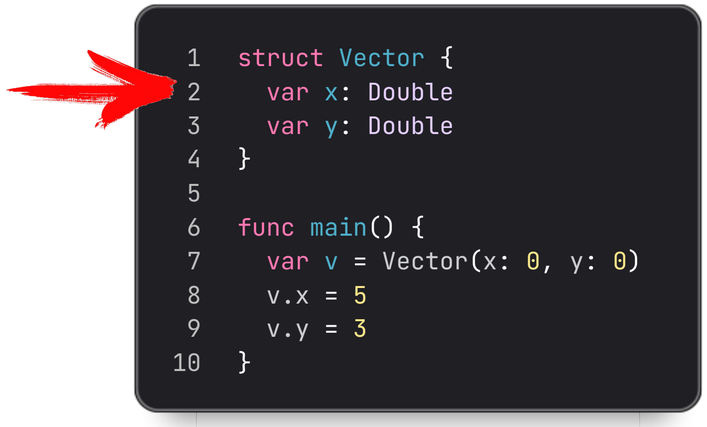


```
1  struct Vector {
2    var x: Double
3    var y: Double
4  }
5
6  func main() {
7    var v = Vector(x: 0, y: 0)
8    v.x = 5
9    v.y = 3
10 }
```

Динамический анализ

Пример Vector.swift

roles	usr	location
definition, implicit, childOf	s:6VectorAA1x1yABSd_Sdtcfc	Vector.swift:1:8
definition	s:6VectorAAV	Vector.swift:1:8
definition, implicit, childOf, accesorOf	s:6VectorAAV1xSdvg	Vector.swift:2:7
definition, implicit, childOf, accesorOf	s:6VectorAAV1xSdvs	Vector.swift:2:7
definition, childOf	s:6VectorAAV1×Sdvp	
reference, containedBy	s:Sd	Vector.swift:2:10
definition, implicit, childOf, accesorOf	s:6VectorAAV1ySdvg	Vector.swift:3:7
definition, implicit, childOf, accesorOf	s:6VectorAAV1ySdvs	Vector.swift:3:7
definition, childOf	s:6VectorAAV1ySdvp	Vector.swift:3:7
reference, containedBy	s:Sd	Vector.swift:3:10

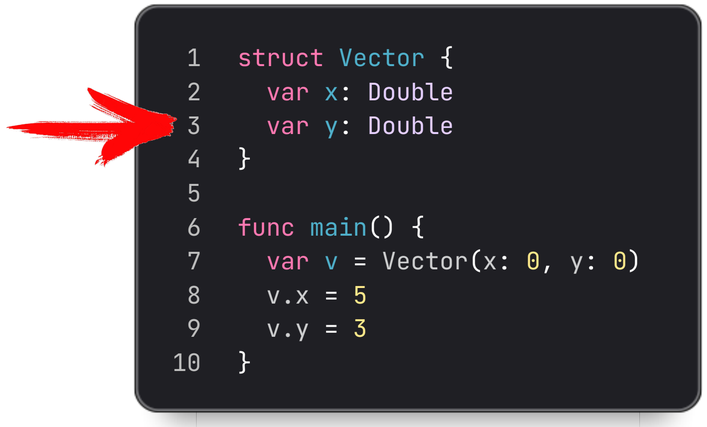


```
1  struct Vector {
2    var x: Double
3    var y: Double
4  }
5
6  func main() {
7    var v = Vector(x: 0, y: 0)
8    v.x = 5
9    v.y = 3
10 }
```

Динамический анализ

Пример Vector.swift

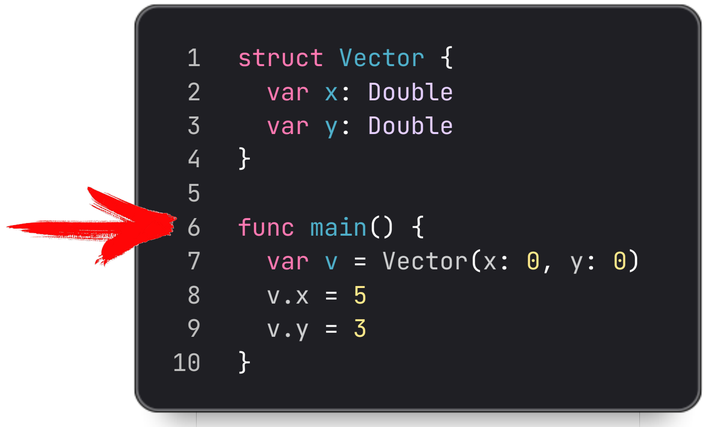
roles	usr	location
definition, implicit, childOf	s:6VectorAA1x1yABSd_Sdtcfc	Vector.swift:1:8
definition	s:6VectorAAV	Vector.swift:1:8
definition, implicit, childOf, accesorOf	s:6VectorAAV1xSdvg	Vector.swift:2:7
definition, implicit, childOf, accesorOf	s:6VectorAAV1xSdvs	Vector.swift:2:7
definition, childOf	s:6VectorAAV1×Sdvp	
reference, containedBy	s:Sd	Vector.swift:2:10
definition, implicit, childOf, accesorOf	s:6VectorAAV1ySdvg	Vector.swift:3:7
definition, implicit, childOf, accesorOf	s:6VectorAAV1ySdvs	Vector.swift:3:7
definition, childOf	s:6VectorAAV1ySdvp	Vector.swift:3:7
reference, containedBy	s:Sd	Vector.swift:3:10



Динамический анализ

Пример Vector.swift

roles	usr	location
Definition	S:6Vector4mainyyF	Vector.swift:6:6
Reference, containedBy	s:6VectorAAV	Vector.swift:7:13
Reference, Call, calledBy, ContainedBy	s:6VectorAAV1x1yABSd_Sdtcfc	Vector.swift:7:13
Reference, containedBy	s:6VectorAAV1×Sdvp	Vector.swift:7:20
Reference, containedBy	s:6VectorAAV1ySdvp	Vector.swift:7:26
Reference, Call, implicit calledBy, ContainedBy	s:6VectorAAV1xSdvs	Vector.swift:8:7
Reference,write,containedBy	s:6VectorAAV1xSdvp	Vector.swift:8:7
Reference, Call, implicit calledBy, ContainedBy	s:6VectorAAV1ySdvs	Vector.swift:9:7
Reference,write,containedBy	s:6VectorAAV1ySdvp	Vector.swift:9:7

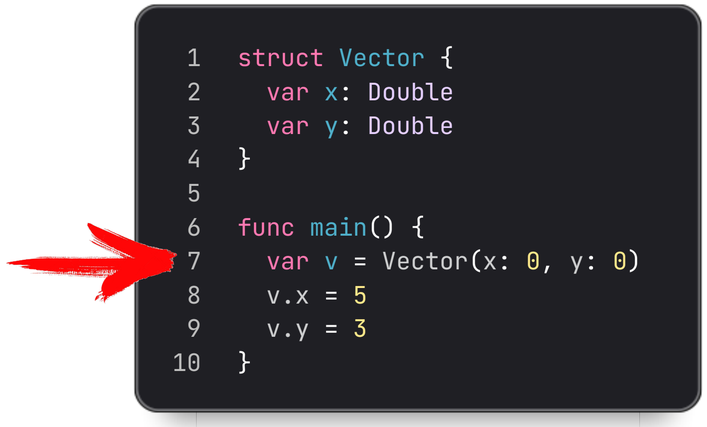


```
1  struct Vector {
2    var x: Double
3    var y: Double
4  }
5
6  func main() {
7    var v = Vector(x: 0, y: 0)
8    v.x = 5
9    v.y = 3
10 }
```

Динамический анализ

Пример Vector.swift

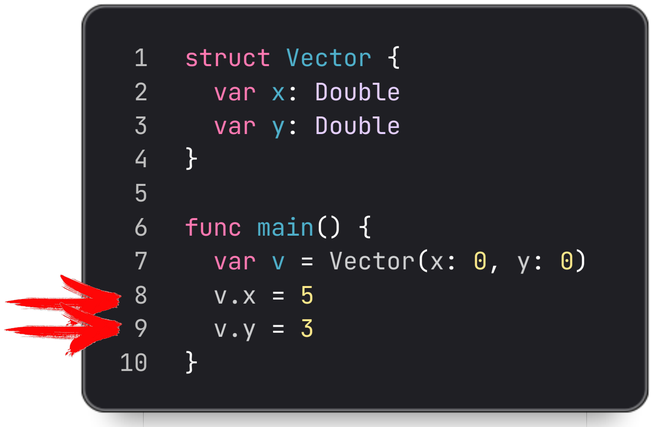
roles	usr	location
Definition	S:6Vector4mainyyF	Vector.swift:6:6
Reference, containedBy	s:6VectorAAV	Vector.swift:7:13
Reference, Call, calledBy, ContainedBy	s:6VectorAAV1x1yABSd_Sdtcfc	Vector.swift:7:13
Reference, containedBy	s:6VectorAAV1×Sdvp	Vector.swift:7:20
Reference, containedBy	s:6VectorAAV1ySdvp	Vector.swift:7:26
Reference, Call, implicit calledBy, ContainedBy	s:6VectorAAV1xSdvs	Vector.swift:8:7
Reference,write,containedBy	s:6VectorAAV1xSdvp	Vector.swift:8:7
Reference, Call, implicit calledBy, ContainedBy	s:6VectorAAV1ySdvs	Vector.swift:9:7
Reference,write,containedBy	s:6VectorAAV1ySdvp	Vector.swift:9:7



Динамический анализ

Пример Vector.swift

roles	usr	location
Definition	S:6Vector4mainyyF	Vector.swift:6:6
Reference, containedBy	s:6VectorAAV	Vector.swift:7:13
Reference, Call, calledBy, ContainedBy	s:6VectorAAV1x1yABSd_Sdtcfc	Vector.swift:7:13
Reference, containedBy	s:6VectorAAV1xSdvp	Vector.swift:7:20
Reference, containedBy	s:6VectorAAV1ySdvp	Vector.swift:7:26
Reference, Call, implicit calledBy, ContainedBy	s:6VectorAAV1xSdvs	Vector.swift:8:7
Reference,write,containedBy	s:6VectorAAV1xSdvp	Vector.swift:8:7
Reference, Call, implicit calledBy, ContainedBy	s:6VectorAAV1ySdvs	Vector.swift:9:7
Reference,write,containedBy	s:6VectorAAV1ySdvp	Vector.swift:9:7



Динамический анализ

Periphery

Динамический анализ + Статический анализ

Динамический анализ

Нет API

```
public final class SourceGraph {
    // Global shared instance to prevent costly deinitialization.
    public static var shared = SourceGraph()

    private(set) public var allDeclarations: Set<Declaration> = []
    private(set) public var usedDeclarations: Set<Declaration> = []
    private(set) public var redundantProtocols: [Declaration: Set<Reference>] = [:]
    private(set) public var rootDeclarations: Set<Declaration> = []
    private(set) public var redundantPublicAccessibility: [Declaration: Set<String>] = [:]

    private(set) var rootReferences: Set<Reference> = []
    private(set) var allReferences: Set<Reference> = []
    private(set) var retainedDeclarations: Set<Declaration> = []
    private(set) var potentialAssignOnlyProperties: Set<Declaration> = []
    private(set) var ignoredDeclarations: Set<Declaration> = []
    private(set) var assetReferences: Set<AssetReference> = []
    private(set) var mainAttributedDeclarations: Set<Declaration> = []
    private(set) var letShorthandContainerDeclarations: Set<Declaration> = []
```

Динамический анализ

Proposal сделать PeripheryKit публичным

Proposal to Support Swift Package Manager in PeripheryKit to Enhance Collaboration and Maintainability #749

rofle100lvl started this conversation in General

rofle100lvl on Jun 1

I am currently developing an open-source tool aimed at analyzing and visualizing project dependency graphs. During this process, I've come to realize that my project shares similar objectives with PeripheryKit. Inspired by the potential synergy, I took a closer look at your codebase and appreciated the clean and structured implementation you have maintained.

To leverage PeripheryKit more effectively in my project and potentially aid others, I propose introducing Swift Package Manager (SPM) support for PeripheryKit. This addition would significantly streamline the integration process for numerous developers who rely on SPM for their dependency management, including myself.

Incorporating SPM support would not only broaden the reach and usability of PeripheryKit but also attract new contributors who can further enhance its capabilities. Additionally, maintaining SPM compatibility would encourage us to uphold the high standards of code quality and organization that PeripheryKit exemplifies.

Looking forward to your thoughts.

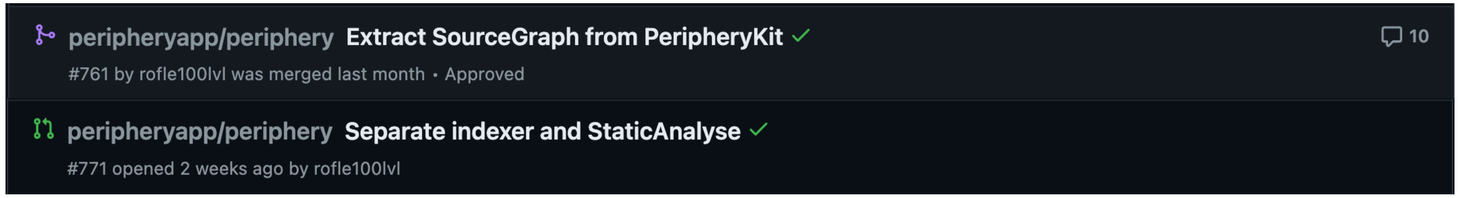
↑ 4 2

Динамический анализ

Положительный ответ

Динамический анализ

Periphery работы



Динамический анализ

Как мы тут оказались?

Монолит 

Динамический анализ

Как мы тут оказались?

Монолит🗿 ⟶ SwiftSyntax

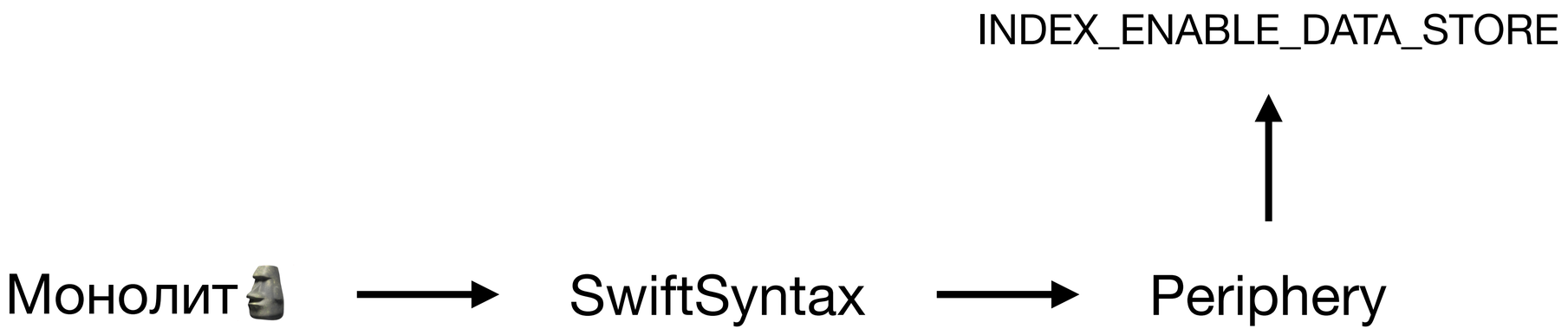
Динамический анализ

Как мы тут оказались?

Монолит  → SwiftSyntax → Periphery

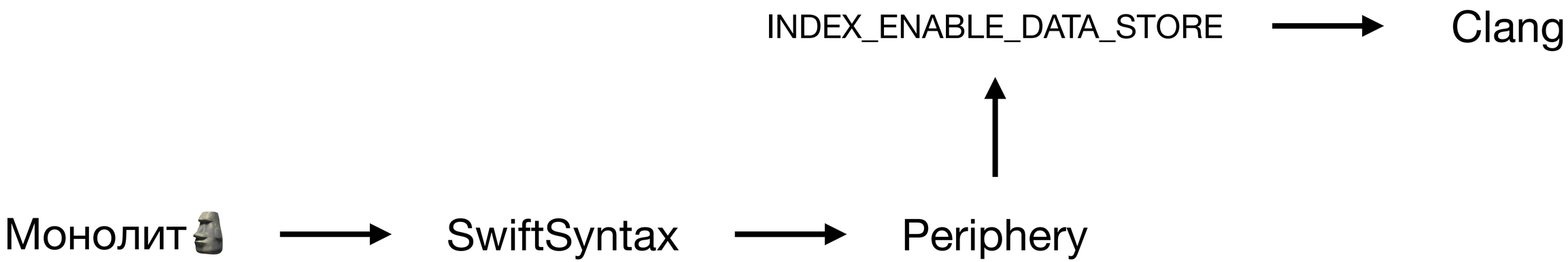
Динамический анализ

Как мы тут оказались?



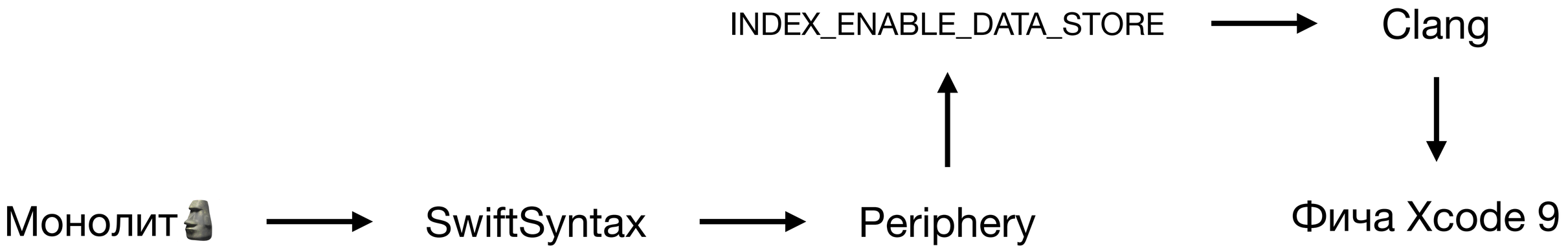
Динамический анализ

Как мы тут оказались?



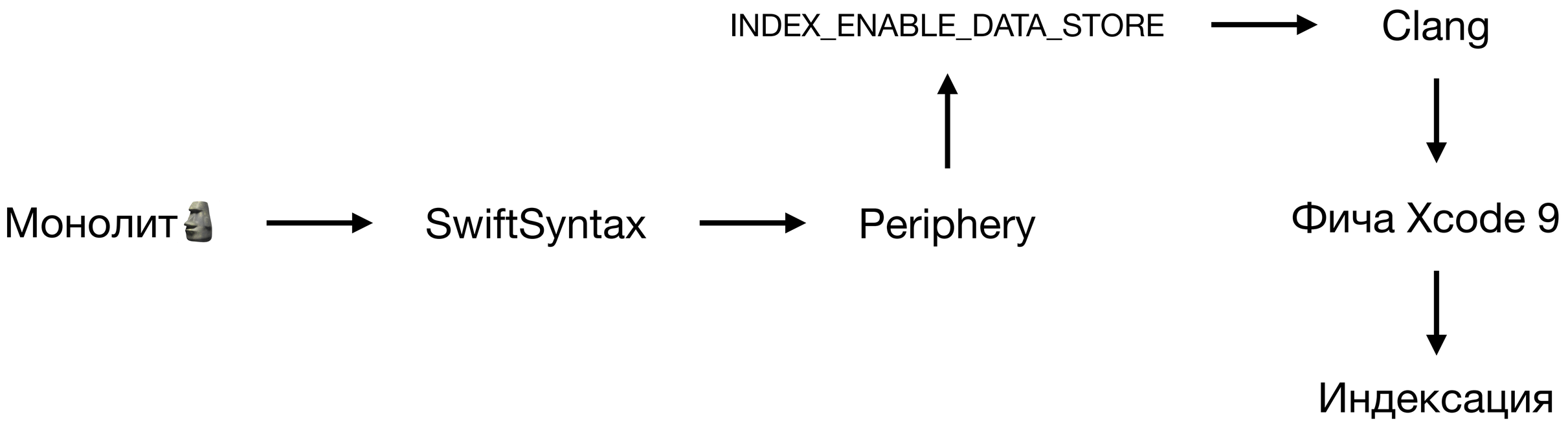
Динамический анализ

Как мы тут оказались?



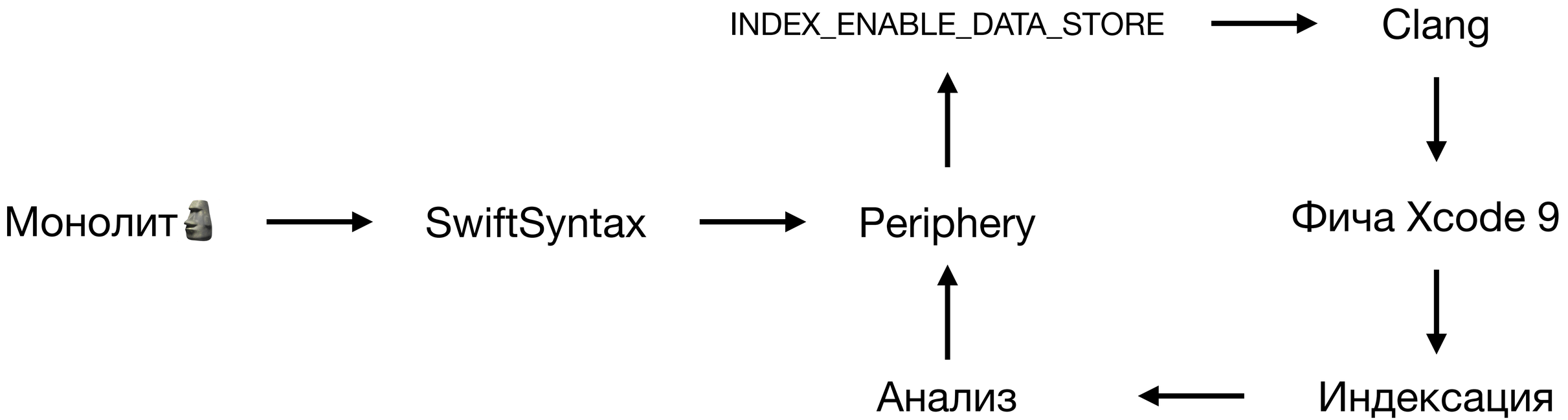
Динамический анализ

Как мы тут оказались?



Динамический анализ

Как мы тут оказались?



Глава 4. Итоги

Итоги

Результаты статический анализ

Nodes: 1041

Edges: 1536

Итоги

Результаты динамический анализ

Nodes: 1821

Edges: 4272

Итоги

Проблема #2

Большой порог входа в неочевидный алгоритм

Итоги

Проблема #2 Большой порог входа в неочевидный алгоритм

1. Сбилдить утилиту
2. Вписать все параметры при запуске
3. Получить два csv файла с описанием ребер и узлов
4. Установить Gephi

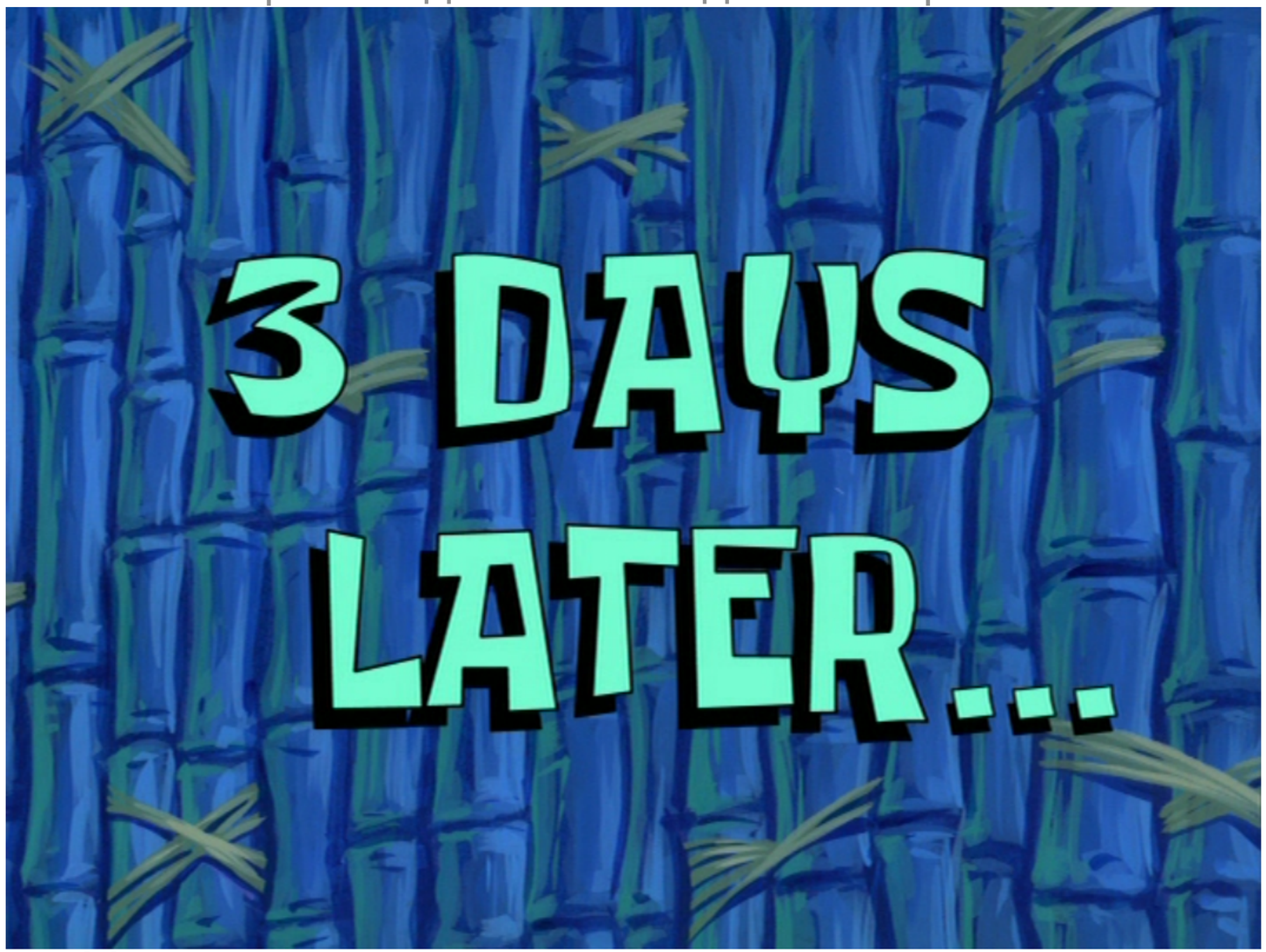
4. Импортировать их в крайне не очевидный Gephi

5. Вспомнить какие фильтры нужно применить

6. Настроить фильтры в неудобном редакторе

Итоги

Проблема #2 Большой порог входа в неочевидный алгоритм

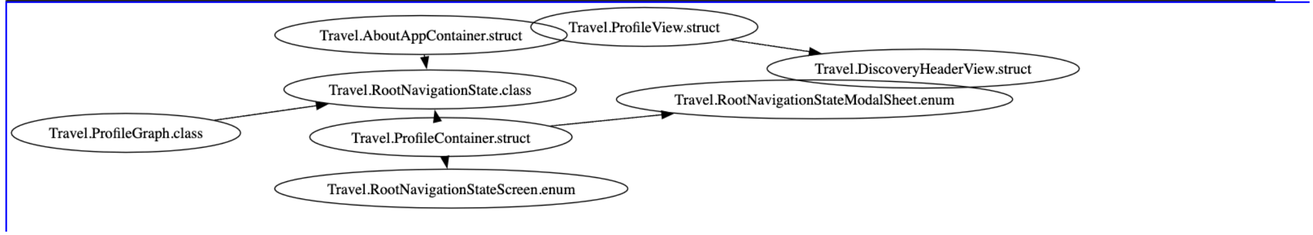


На все это уходило очень много времени

Итоги

Решение проблемы #2

From	To	Lines
Travel.ProfileView.struct	Travel.DiscoveryHeaderView.struct	15, 15
Travel.AboutAppContainer.struct	Travel.RootNavigationState.class	13, 21
Travel.ProfileContainer.struct	Travel.RootNavigationState.class	20, 30, 61, 103, 128, 163, 186, 223
Travel.ProfileGraph.class	Travel.RootNavigationState.class	25, 45
Travel.ProfileContainer.struct	Travel.RootNavigationStateModalSheet.enum	163
Travel.ProfileContainer.struct	Travel.RootNavigationStateScreen.enum	104, 129, 151, 175, 186, 198, 209, 269



Итоги

Что мы получили?

Количество строк в монолите 80k -> 29k

Итоги

Что мы получили?

Количество строк в монолите 80k -> 29k

% монолита от общего кода: 36% -> 9%

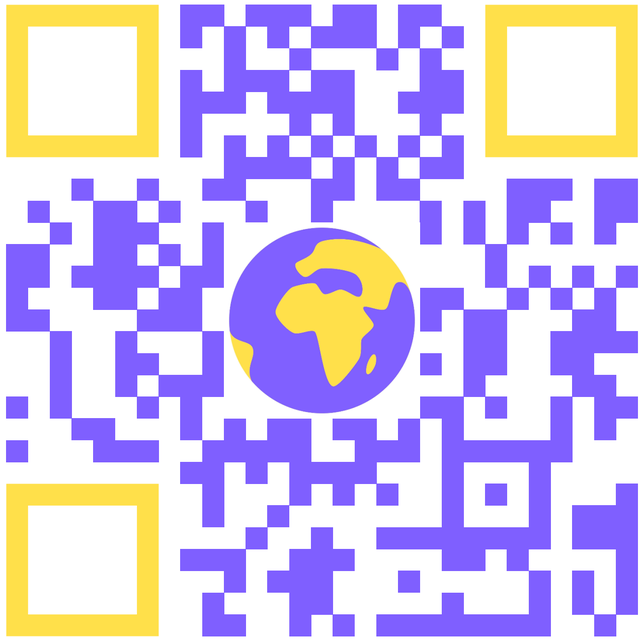
Как мы адаптировали проект к Strict Concurrency Checking

Доклад про

Глава 5. Что мне с этим делать?

Что мне с этим делать?

Шаг 1: Склонировать UseGraph

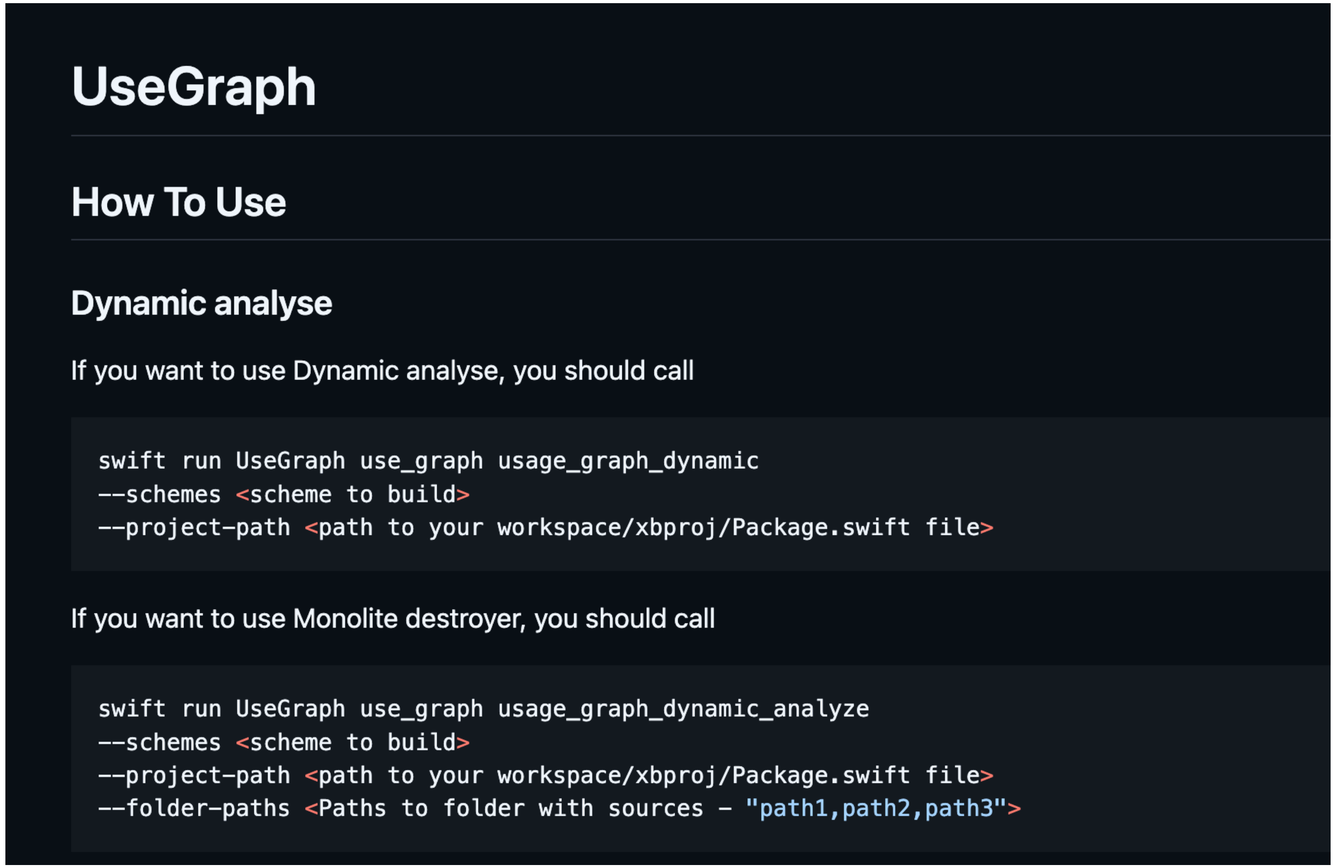


Что мне с этим делать?

Шаг 1: Склонировать UseGraph

Что мне с этим делать?

Шаг 2: Получить граф



UseGraph

How To Use

Dynamic analyse

If you want to use Dynamic analyse, you should call

```
swift run UseGraph use_graph usage_graph_dynamic
--schemes <scheme to build>
--project-path <path to your workspace/xbproj/Package.swift file>
```

If you want to use Monolite destroyer, you should call

```
swift run UseGraph use_graph usage_graph_dynamic_analyze
--schemes <scheme to build>
--project-path <path to your workspace/xbproj/Package.swift file>
--folder-paths <Paths to folder with sources - "path1,path2,path3">
```

Что мне с этим делать?

Шаг 3: Анализировать

Nodes.csv

Edges.csv

Что мне с этим делать?

Идеи для анализа

Найти и избавиться от циклов в коде

Что мне с этим делать?

Идеи для анализа

Найти и избавиться от циклов в коде

Найти узлы с высокой связностью

Что мне с этим делать?

Идеи для анализа

Найти и избавиться от циклов в коде

Найти узлы с высокой связностью

Оценить архитектуру проекта

Lobzik: SemiAutomatic Modularization

Доклад про метрики

Что мне с этим делать?

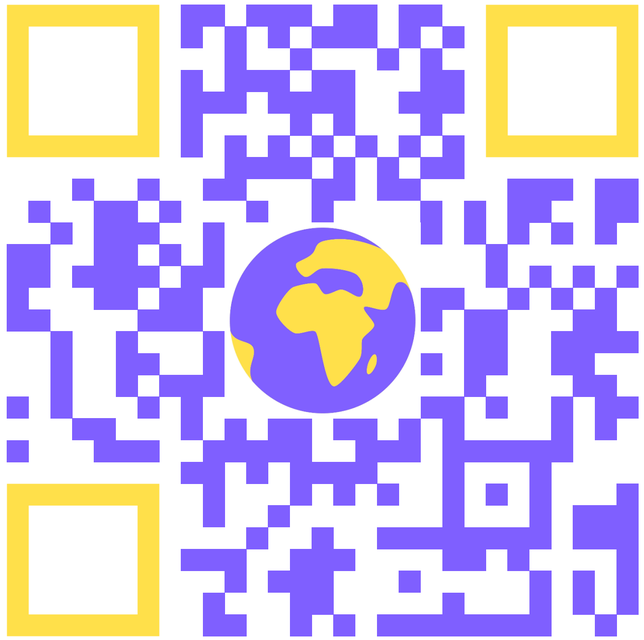
Идеи для анализа

Найти и избавиться от циклов в коде

Найти узлы с высокой связностью

Оценить архитектуру проекта

Изолированные узлы



Спасибо за внимание

Путешествия