

AppCode под капотом

Айдар Мухаметзянов, JetBrains



Знакомство и история AppCode



Поддержка языков программирования



Сборка, запуск и отладка iOS приложения



Команда и процессы



Знакомство и история AppCode



Поддержка языков программирования



Сборка, запуск и отладка iOS приложения



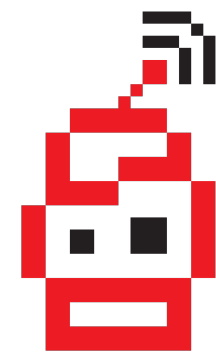
Команда и процессы

Знакомство

Про меня

7 лет

в iOS разработке



Mind



2019 г.

старт в JetBrains
(AppCode)



Знакомство

Про AppCode

IDE для разработки под
macOS и iOS

Поддерживает

Swift, Objective-C/C++, C/C++

Платформа

IntelliJ



История AppCode

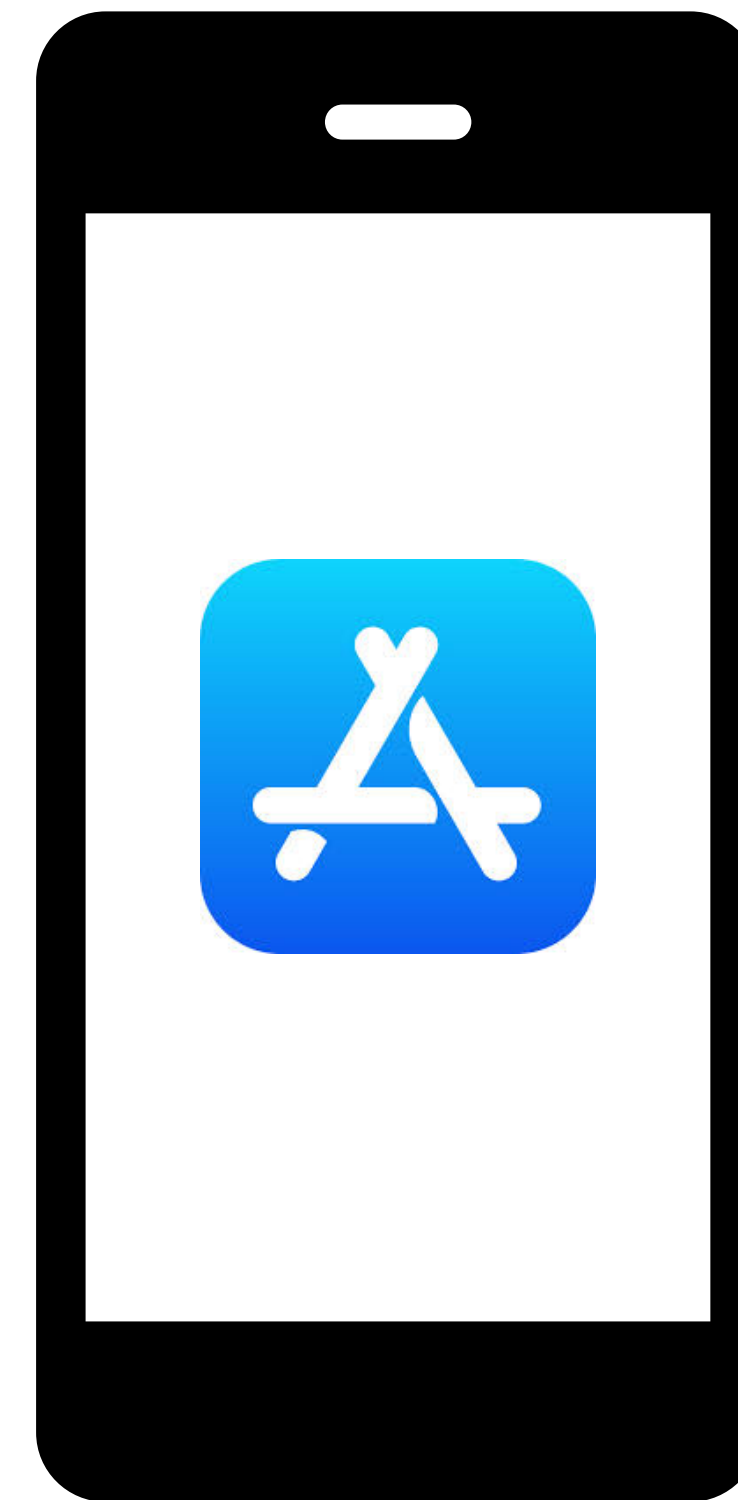
От идеи до воплощения

2008

зарождение идеи
IDE для Objective-C

2009

парсер для Objective-C



История AppCode

От идеи до воплощения

Objective-C → Objective-C++ → C/C++



CLion



ReSharper C++

История AppCode

От идеи до воплощения



2011

первая публичная версия

История AppCode

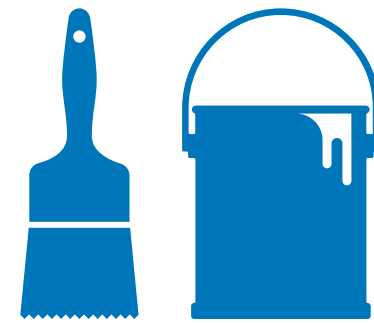
Трудности на пути

Xcode 4

Совпадающие релизы 

История AppCode

Трудности на пути



Interface Builder

2014

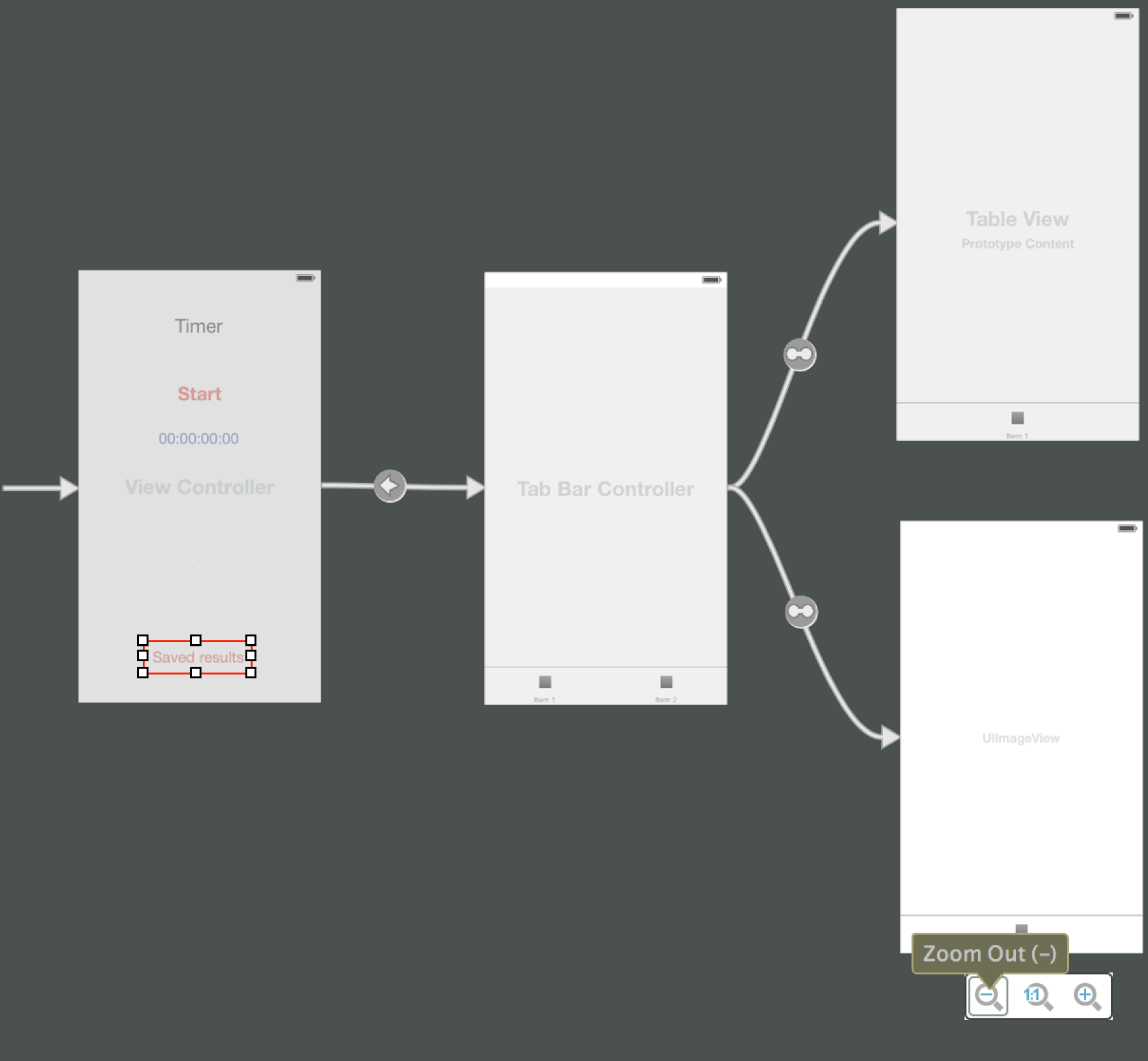
UI Designer

Component Tree

- Document
 - Jb View Controller Scene
 - Jb View Controller
 - Top Layout Guide
 - Bottom Layout Guide
 - View
 - Button
 - Label – 00:00:00:00
 - Activity Indicator – La
 - Label – Timer
 - Button**

First Responder
Exit
Push segue from Button to T

bounds	
Custom Class	
Type	System
State Config	Default
Title	Plain
Text	Saved results
Font	System 20.0
Text Color	[191, 15, 10, 255]
Shadow	[128, 128, 128]



Palette

- Controllers & Objects
 - View Controller
 - Table View Controller
 - Collection View Controller
 - Navigation Controller
 - Tab Bar Controller
 - Page View Controller
 - UIKit View Controller
 - Object
- Controls
 - Label
 - Button
 - Segmented Control
 - Text Field
 - Slider
 - Switch
 - Activity Indicator
 - Progress View
 - Page Control
 - Stepper
- Data Views
 - Table View
 - Table View Cell
 - Collection View
 - Collection View Cell
 - Collection Reusable View
 - Image View
 - Text View
 - Web View

История AppCode

Трудности на пути



История AppCode

2008

зарождение идеи

2011

первая публичная версия

2014

UIDesigner Swift



Знакомство и история AppCode



Поддержка языков программирования



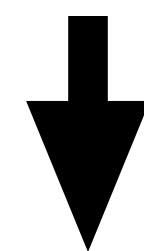
Сборка, запуск и отладка iOS приложения



Команда и процессы

Быстрые преобразования

```
if (b > 0 ? b * 4 : b / 5) != a { //...
```



```
if a != (b > 0 ? b * 4 : b / 5) { //...
```



Ctrl+C, Ctrl+V, Ctrl+C, Ctrl+V

Быстрые преобразования

```
if (b > 0 ? b * 4 : b / 5) != a { //...  
    print("Hello")  
}
```

Flip '!='

Negate '!=' to '=='

Press `\Space` to open preview



Alt+Enter, Enter

Быстрые преобразования

```
if (b > 0 ? b * 4 : b / 5) != a { //...
```



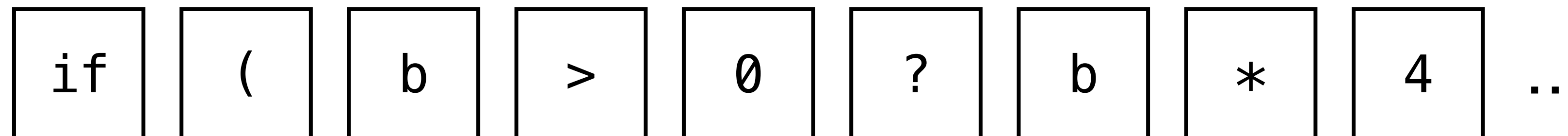
Работа с кодом

Введение

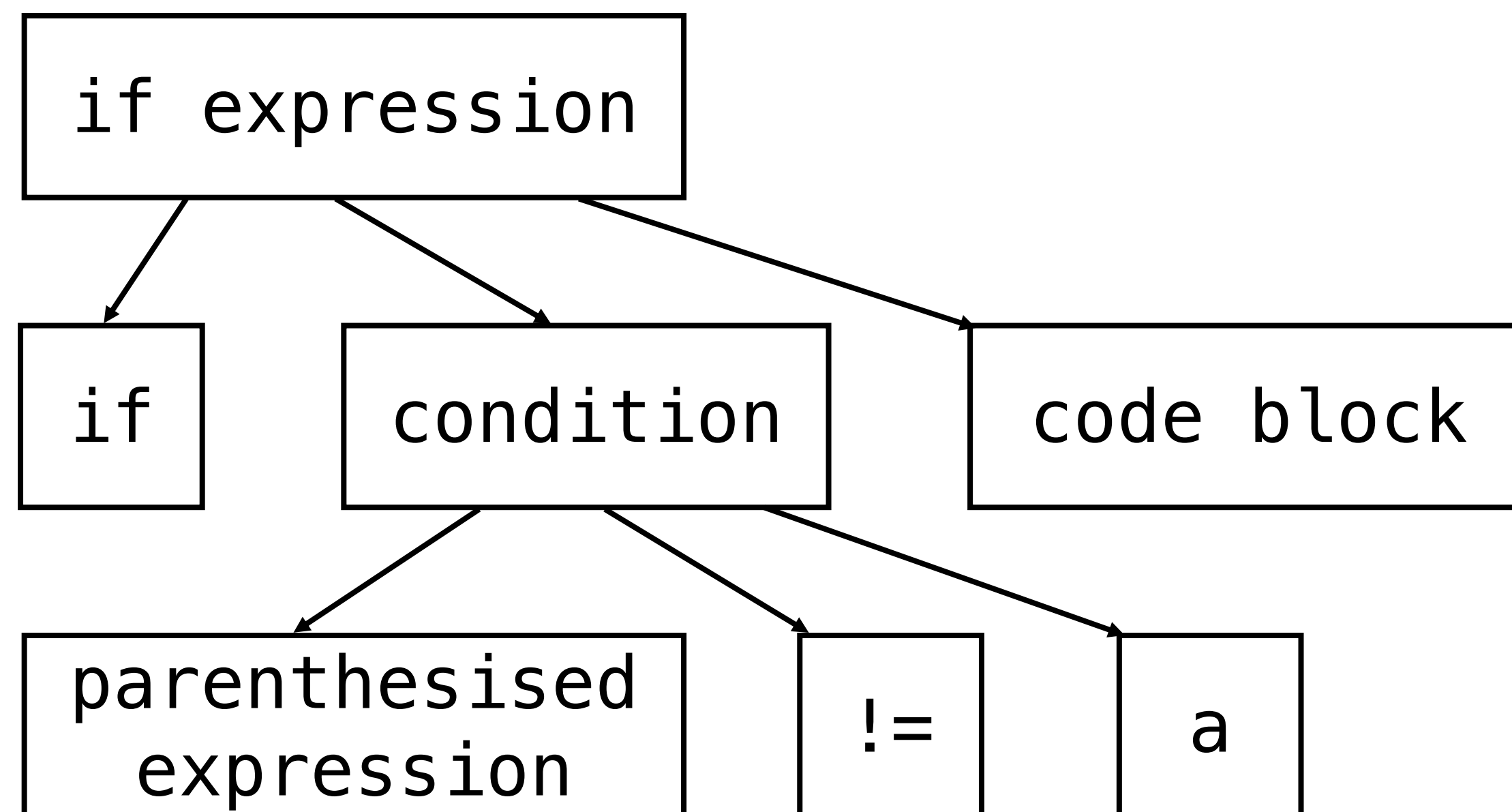
Текст

```
if (b > 0 ? b * 4 : b / 5) != a { //...
```

Лексемы

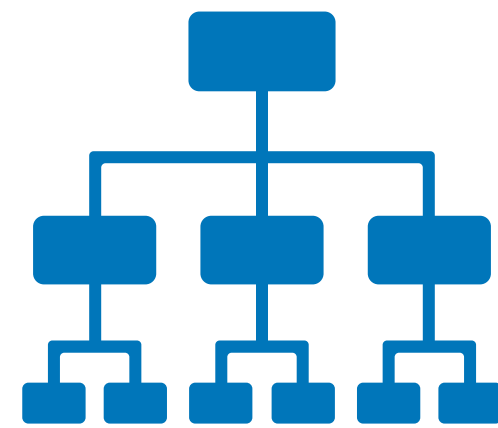


Дерево разбора



Работа с кодом

PSI дерево



Program Structure Interface (PSI)

Работа с кодом

PSI дерево

```
▼ ψ SwiftIfStatementGenImpl(IF_STATEMENT)
  ψ PsiElement(if)
  ─ PsiWhiteSpace
  ▼ ψ SwiftConditionClauseGenImpl(CONDITION_CLAUSE)
    ▼ ψ SwiftExpressionConditionGenImpl(EXPRESSION_CONDITION)
      ▼ ψ SwiftComplexOperatorExpressionGenImpl(COMPLEX_OPERATOR_EXPRESSION)
        ▶ ψ SwiftParenthesizedExpressionGenImpl(PARENTHESIZED_EXPRESSION)
          ─ PsiWhiteSpace
        ▶ ψ SwiftBinaryOperatorGenImpl(BINARY_OPERATOR)
          ─ PsiWhiteSpace
        ▶ ψ SwiftReferenceExpressionGenImpl(REFERENCE_EXPRESSION)
```

PsiViewer

<https://plugins.jetbrains.com/plugin/227-psiviewer>

Работа с кодом

Грамматика языка

БНФ

Форма Бэкуса—Наура

```
while-statement ::= while condition-list code-block
```

<https://docs.swift.org/swift-book/ReferenceManual/zzSummaryOfTheGrammar.html>

Работа с кодом

GrammarKit

Редактируем **Swift.bnf**

GrammarKit генерирует парсер


Парсер строит **PSI-дерево**

<https://github.com/JetBrains/Grammar-Kit>

Работа с кодом

Пример быстрого преобразования

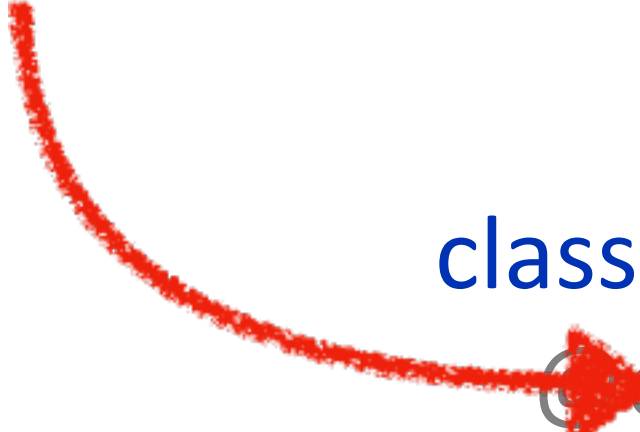
```
if (b > 0 ? b * 4 : b / 5) != a { //...
```



1. Обновляется PSI
2. Применяются настройки форматирования
3. Генерируется текст

Работа с кодом

Инспекции



```
class MyObject: AnotherObject {  
    @objc override func foo() {}  
    @objc override func bar() {}  
}
```

Alt+Enter —> *Remove redundant @objc attribute*

Работа с кодом

Инспекции

```
class MyObject: AnotherObject {  
    @objc override func foo() {}  
    @objc override func bar() {}  
}
```

```
class AnotherObject: NSObject {  
    @objc func foo() {}  
    func bar() {}  
}
```

Работа с кодом

Инспекции

```
class MyObject: AnotherObject {  
    @objc override func foo() {}  
    @objc override func bar() {}  
}
```



PSI

Поиск декларации элемента

Применение

Навигация по коду

Подсветка кода

Рефакторинги

Инспекции

Быстрые преобразования

...

Поиск декларации элемента

Построение символьных таблиц

```
class Person {  
  let name = "John"  
  let surname = "Doe"  
  let age = 42  
  
  // Very important function  
  func greet() {  
    print("Hi \!(name)!")  
  }  
}
```

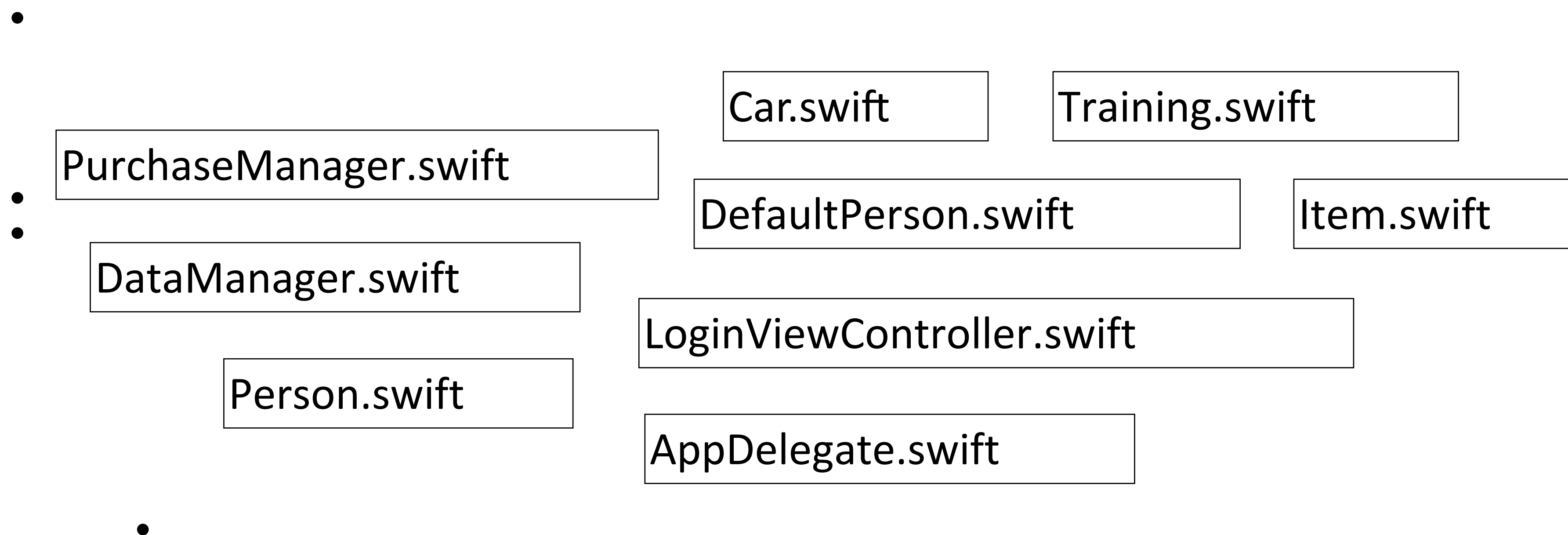
Символьная таблица

```
class Person  
  let name: String  
  let surname: String  
  let age: Int  
  func greet() -> Void
```

Поиск декларации элемента

Построение символьных таблиц

Кэш символьных таблиц

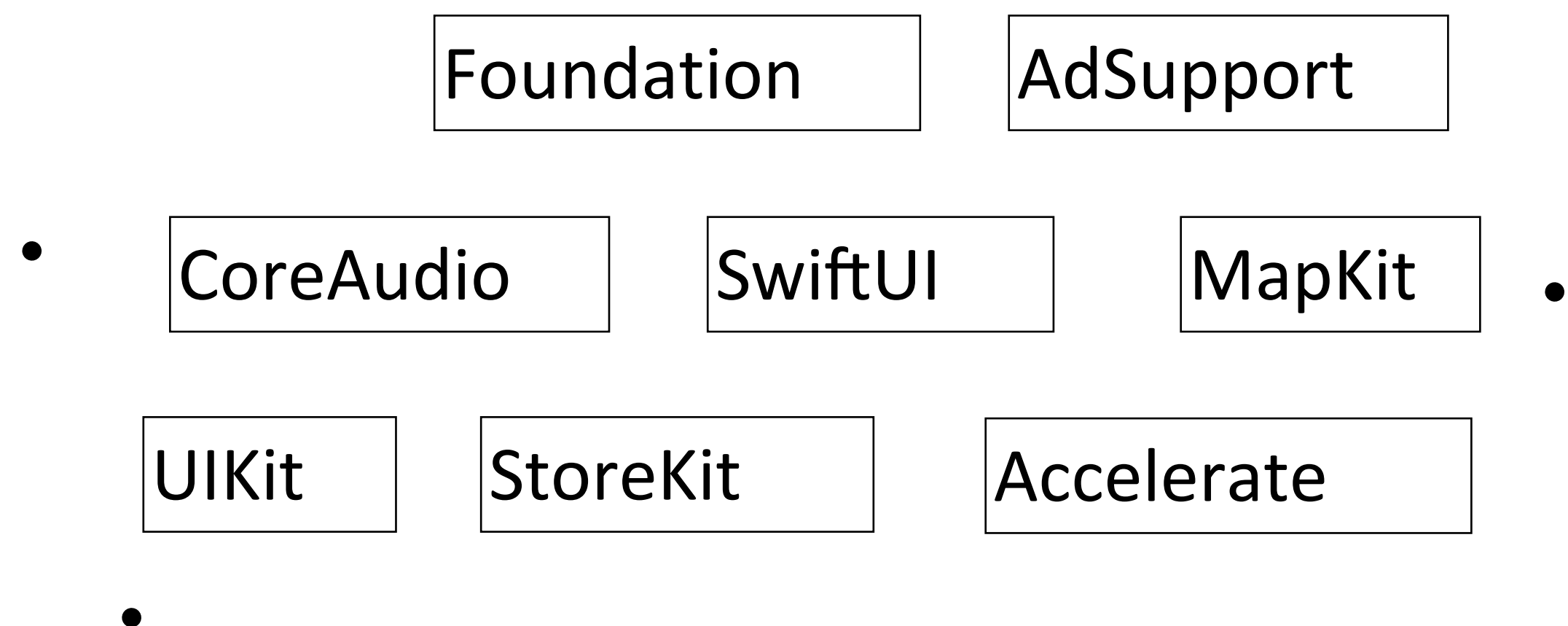


Поиск декларации элемента

Построение символьных таблиц

```
import Foundation
import StoreKit
import CoreAudio
import Accelerate
import SwiftUI
import MapKit
import AdSupport
...
```

Кэш символьных таблиц модулей



Поиск декларации элемента

Инвалидация символьных таблиц

```
class Person {  
    let name = "John"  
    let surname = "Doe"  
    let age = 42  
  
    // Very important function  
    func greet() {  
        print("Hi \!(name)!")  
    }  
}
```

Символьная таблица

```
class Person  
let name: String  
let surname: String  
let age: Int  
func greet() -> Void
```

Поиск декларации элемента

Инвалидация символьных таблиц

```
class Person {  
    let name = "John"  
    let surname = "Doe"  
    let dateOfBirth: Date? = nil  
  
    // Very important function  
    func sayHello() {  
        print("Hi \!(name)!")  
    }  
}
```

Символьная таблица

```
class Person  
let name: String  
let surname: String  
let age: Int  
func greet() -> Void
```



Поиск декларации элемента

Инвалидация символьных таблиц

Objective-C/Objective-C++
C/C++

Поиск декларации элемента

Инвалидация символьных таблиц

```
#include "SwiftASTManager.h"  
#include "SwiftEditorDiagConsumer.h"  
#include "SwiftLangSupport.h"  
//...
```

Поиск декларации элемента

Инвалидация символьных таблиц

```
1  #include <iostream>
2
3  ► int main() {
4      std::cout << "Hello, World!" << std::endl;
5      return 0;
6  }
7
```

```
clang -E main.cpp > preprocessed.cpp
```

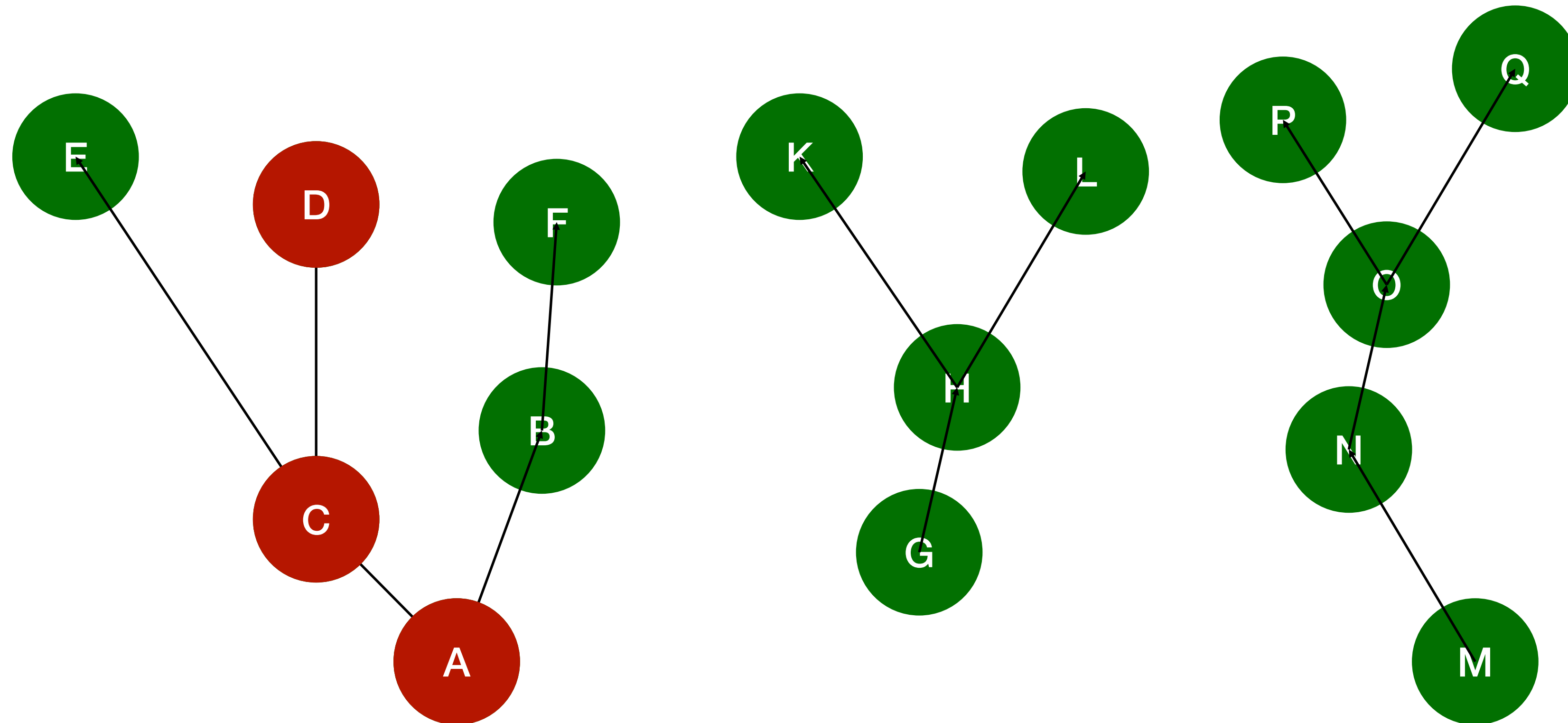
Поиск декларации элемента

Инвалидация символьных таблиц

```
41504 int main() {  
41505     std::cout << "Hello, World!" << std::endl;  
41506     return 0;  
41507 }
```

Поиск декларации элемента

Инвалидация символьных таблиц



Поиск декларации элемента

Инвалидация символьных таблиц



```
#if DEBUG
    print("Hi \(name)!")
#else
    print("Hello \(name)!")
#endif
```

Поиск декларации элемента

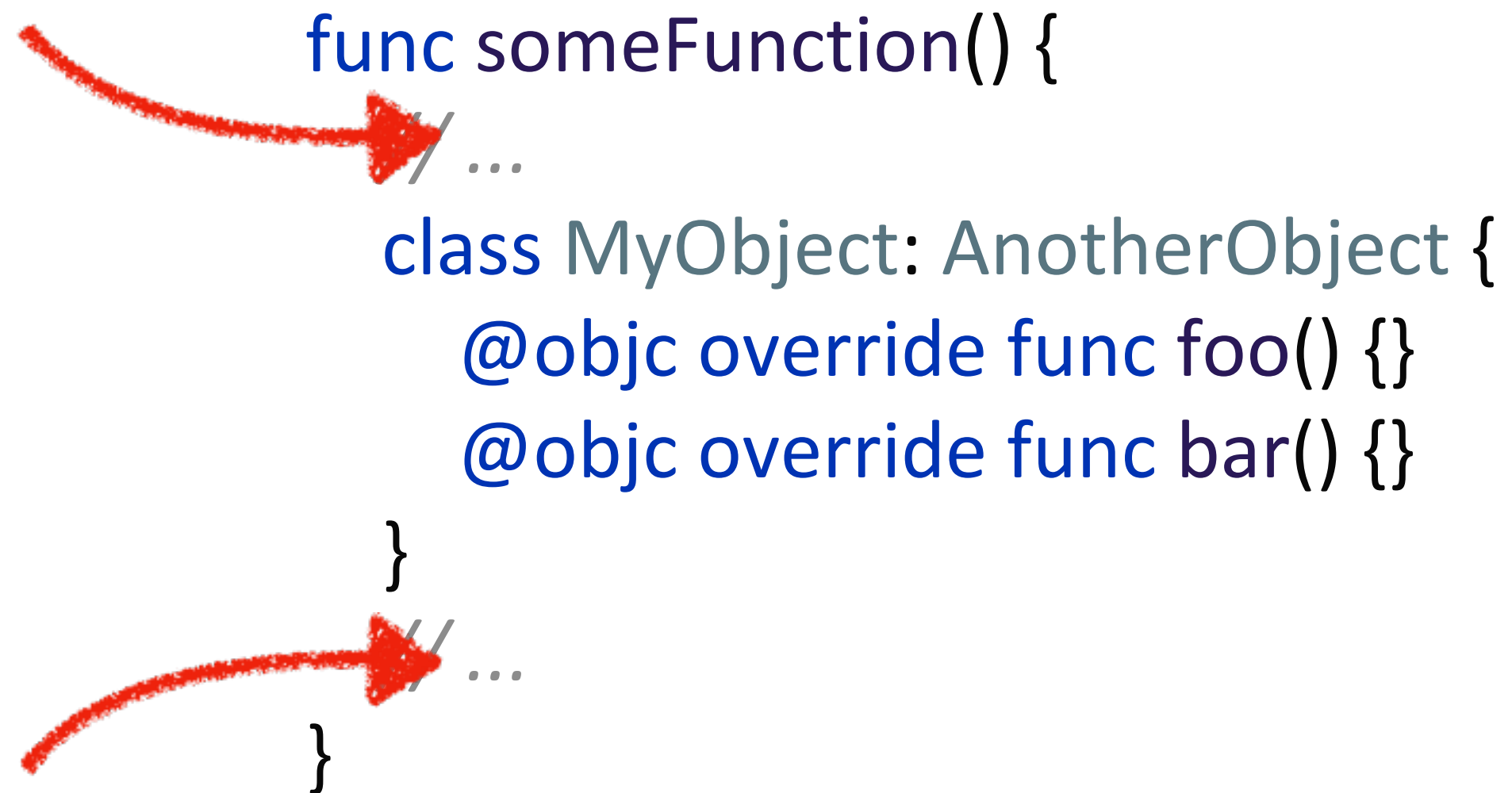
Поиск декларации типа

```
class MyObject: AnotherObject {  
    @objc override func foo() {}  
    @objc override func bar() {}  
}
```

Поиск декларации элемента

Поиск декларации типа

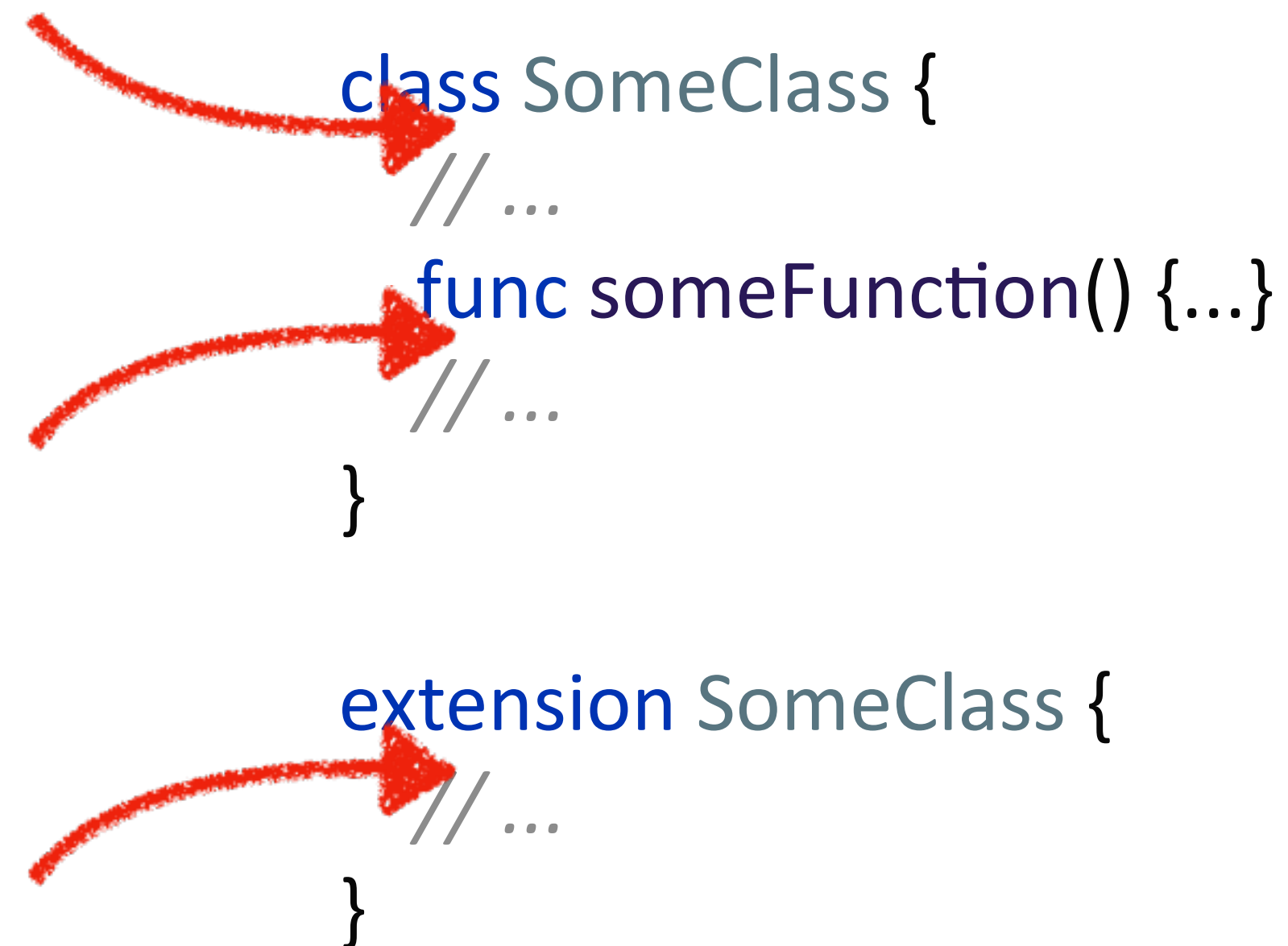
```
func someFunction() {  
  // ...  
  class MyObject: AnotherObject {  
    @objc override func foo() {}  
    @objc override func bar() {}  
  }  
  // ...  
}
```

The diagram illustrates the search process for declarations in a code snippet. Two red arrows originate from the left side. The upper arrow starts at the level of the function declaration and points to the opening curly brace of the class definition. The lower arrow starts at the level of the class definition and points to the opening curly brace of the function body. This indicates that the search for a function declaration follows the class's superclass chain, while the search for a class declaration follows the function's enclosing scope.

Поиск декларации элемента

Поиск декларации типа

```
class SomeClass {  
    // ...  
    func someFunction() {...}  
    // ...  
}  
  
extension SomeClass {  
    // ...  
}
```



Поиск декларации элемента

Поиск декларации типа

```
// ...  
class SomeClass {}  
// ...  
extension SomeClass {}  
// ...
```

MyFile.swift

Поиск декларации элемента

Поиск декларации типа

MyFile.swift
ImportantFile.swift
Util.swift
Network.swift
Data.swift
etc...

MyModule

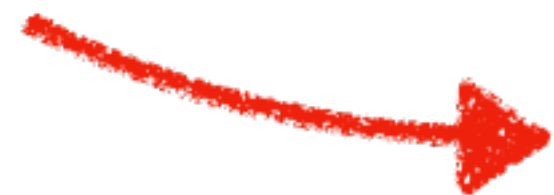
Поиск декларации элемента

Поиск декларации типа

```
import Foundation
import StoreKit
import CoreAudio
import Accelerate
import SwiftUI
import MapKit
import MyLibrary
```

Поиск декларации элемента

Поиск декларации функции



```
func bar() {}  
func bar(_ v: Int = 0) {}  
  
bar()
```

Поиск декларации элемента

Поиск декларации функции

```
func bar(_ v: String = "") {}
```

```
private func someFunction() {  
    func bar(_ v: Int) {}  
}
```

```
    bar("hello")  
}
```

```
extension Int: ExpressibleByStringLiteral {...}
```

Поиск декларации элемента

Swift и Objective-C

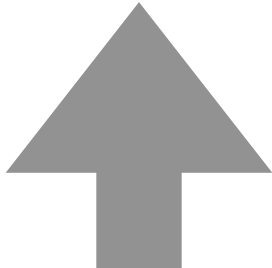
```
class MyObject: AnotherObject {}
```



```
class AnotherObject: NSObject
```



```
@interface AnotherObject : NSObject  
//...  
@end
```

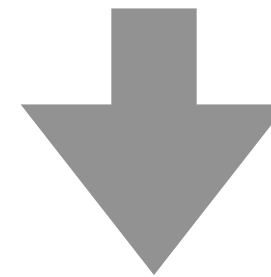


objc

Поиск декларации элемента

Swift и Objective-C

```
class MyObject: AnotherObject {}
```



```
@interface MyObject : NSObject
```

objc

```
@interface AnotherObject : NSObject  
//...  
@end
```

objc

Поиск декларации элемента

Поиск декларации типа

```
class MyObject: AnotherObject {  
    @objc override func foo() {}  
    @objc override func bar() {}  
}
```



Indexing...

Навигация

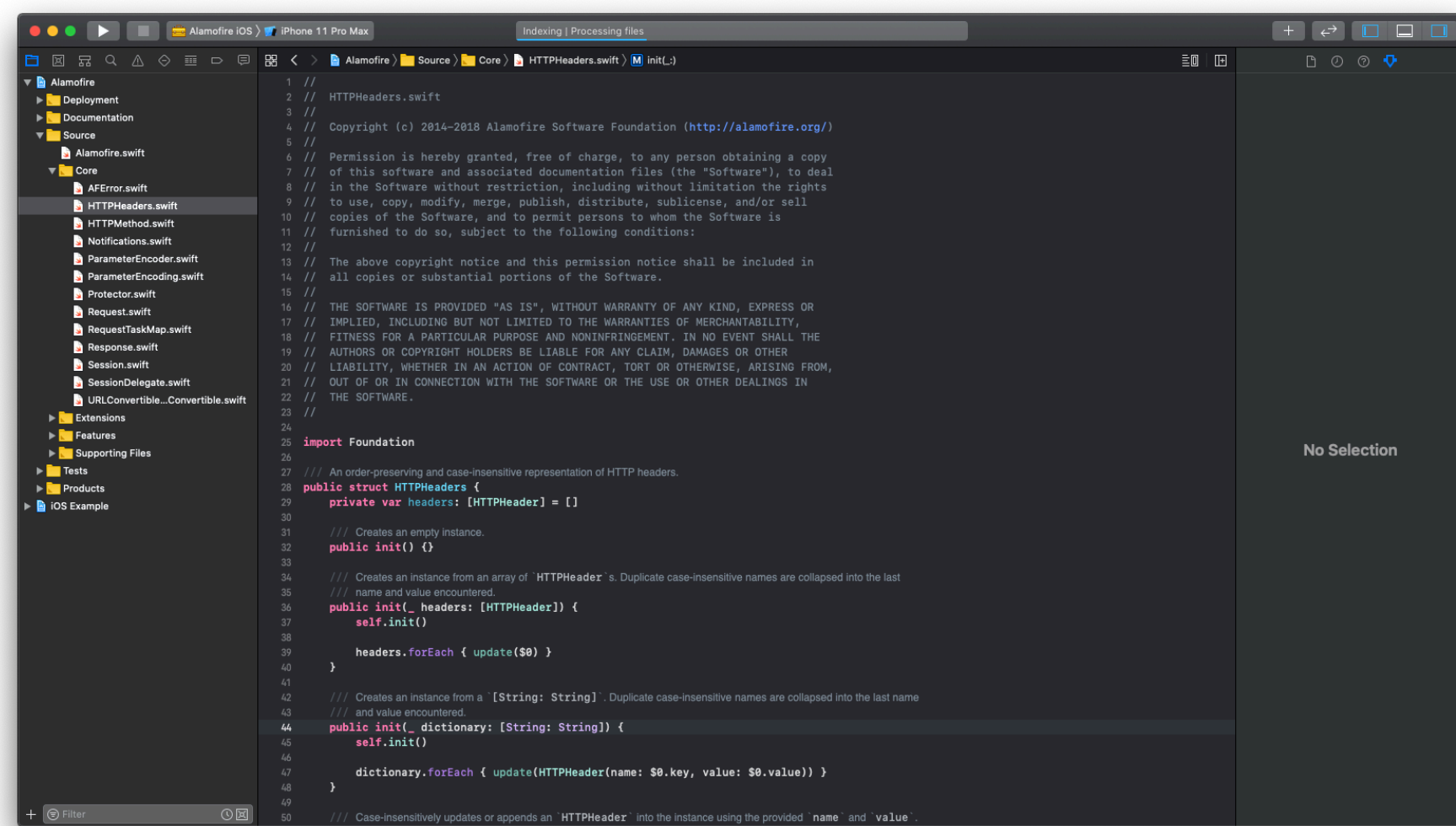
Подсветка (не вся)

Инспекции

Автодополнение

Использование SourceKit

Что такое SourceKit



request



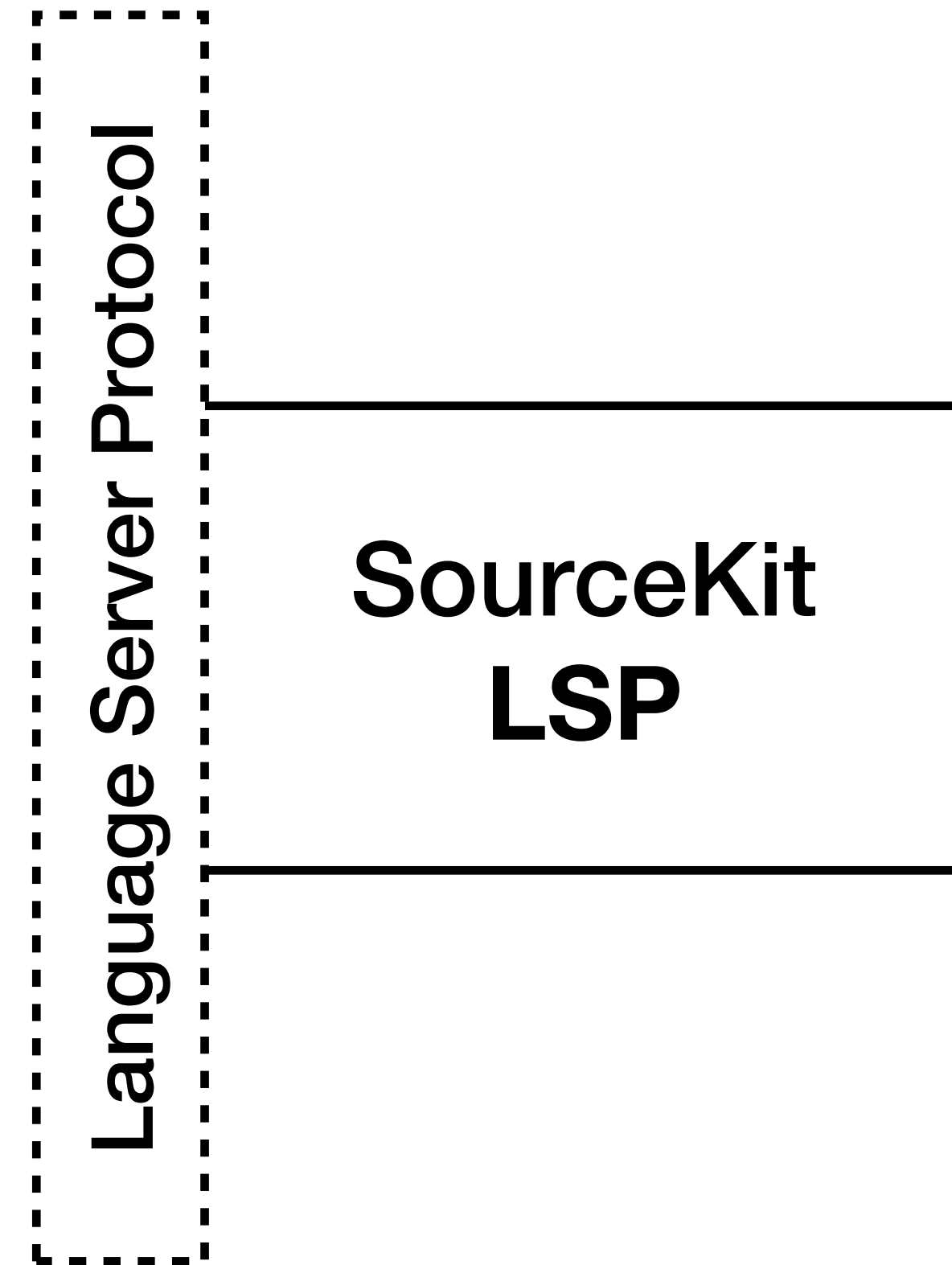
response

Навигация
Подсветка
Автодополнение

...

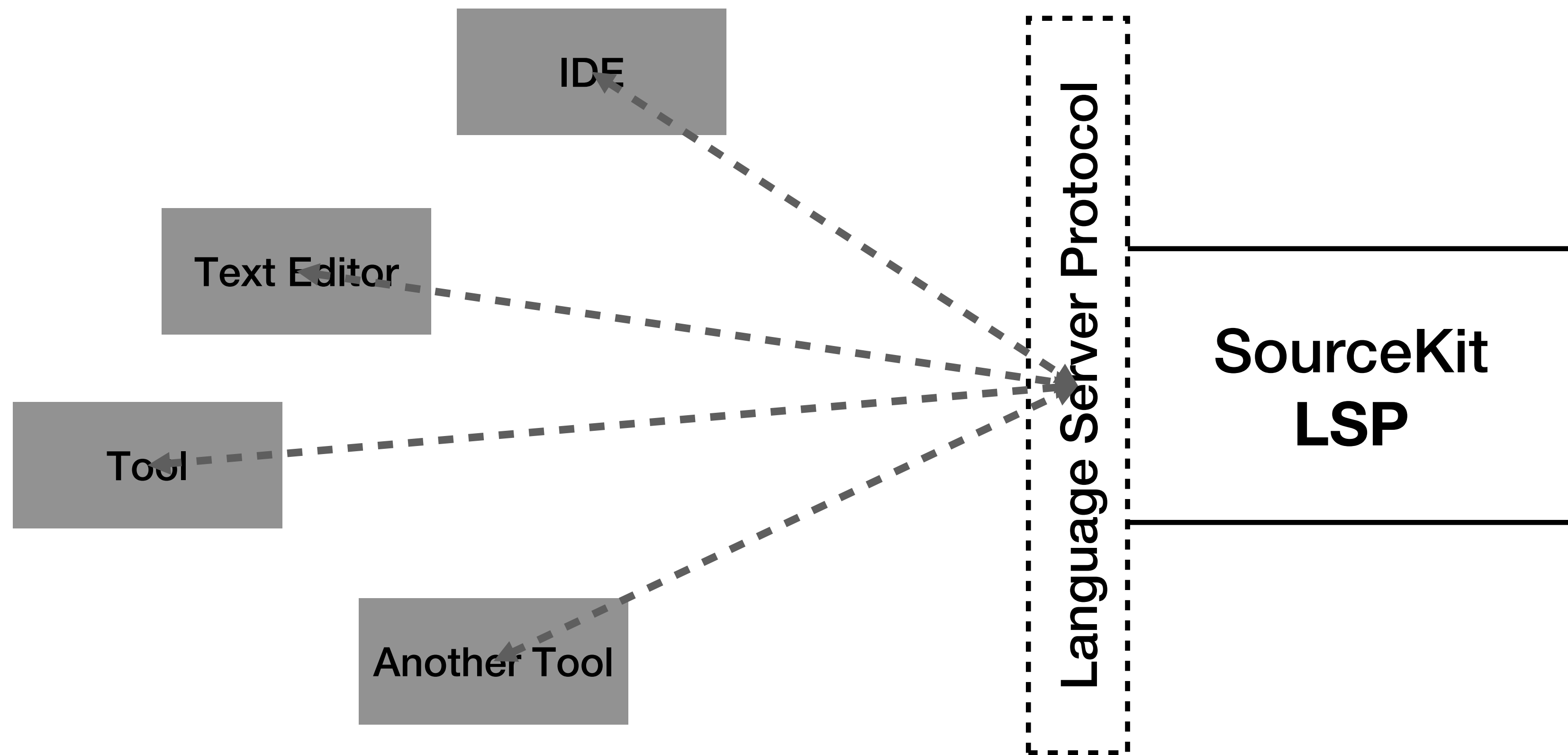
Использование SourceKit

Что такое SourceKit



Использование SourceKit

Что такое SourceKit



Использование SourceKit

AppCode



```
var a: Int? = null
```

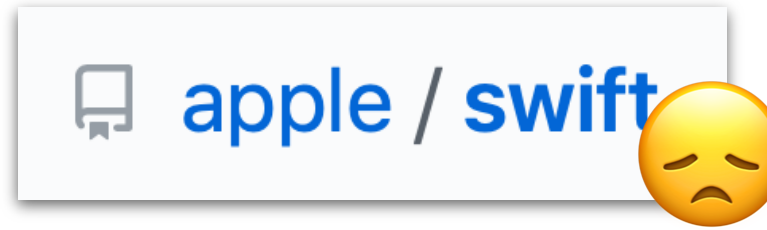
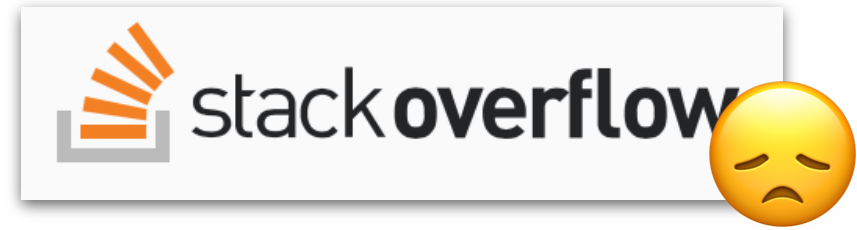
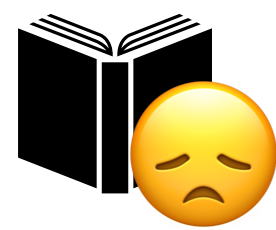
Use of unresolved identifier 'null' ⋮
Create global variable 'null' ⌘⇧⌘ More actions... ⌘⇧

```
1.bitW
```

F bitWidth Int
Press ⌘ to insert, → to replace ⋮

Использование SourceKit

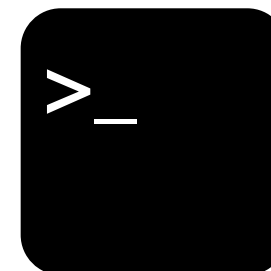
Разработка



Использование SourceKit

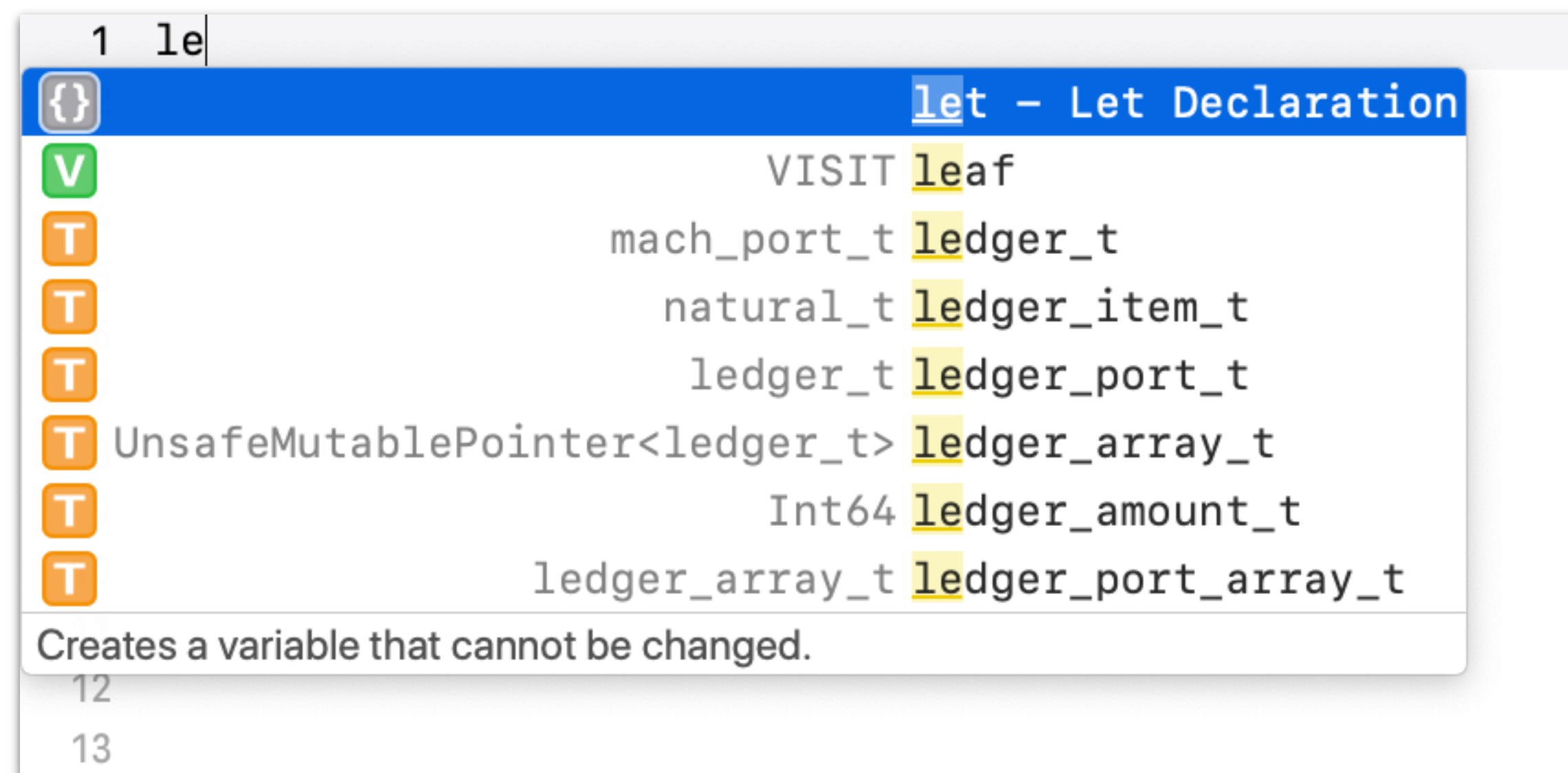
Разработка

```
$ SOURCEKIT_LOGGING=3 /Applications/Xcode.app/Contents/MacOS/Xcode
```



Использование SourceKit

Автодополнение



The screenshot shows a code editor with the text '1 le' on the first line. A completion menu is open, listing several options for the 'let' keyword. The top option is '{ }' with the label 'let - Let Declaration'. Below it are several type-related options, each with a colored icon (green 'V' for visit, orange 'T' for type) and a yellow highlight on the keyword 'let':

- V VISIT let
- T mach_port_t let
- T natural_t let
- T ledger_t let
- T UnsafeMutablePointer<ledger_t> let
- T Int64 let
- T ledger_array_t let

At the bottom of the menu, there is a description: 'Creates a variable that cannot be changed.' Below the menu, the line numbers 12 and 13 are visible.

SOURCEKIT_LOGGING=3 /Applications/Xcode.app/Contents/MacOS/Xcode 2> logs.txt

Использование SourceKit

Автодополнение

```
2020-04-02 15:46:10.180 Xcode[94422:5859798] SourceKit-client: [2:request:112131: 7.4167] [31] {
  key.request: source.request.codecomplete,
  key.compilerargs: [<...>],
  key.offset: 0,
  key.sourcefile: "/Users/aydar.mukhametzhanov/Documents/iOS/EmptyMac/EmptyMac/main.swift",
  key.sourcetext: "le",
  key.toolchains: [
    "com.apple.dt.toolchain.XcodeDefault"
  ]
}
```

Использование SourceKit

Автодополнение

```
2020-04-02 15:46:11.664 Xcode[94422:5859799] SourceKit-client: [2:response:112131: 8.7030] [31] {
  key.results: [
    {
      key.kind: source.lang.swift.keyword,
      key.name: "let",
      key.sourcetext: "let",
      key.description: "let",
      key.typename: "",
      key.context: source.codecompletion.context.none,
      key.num_bytes_to_erase: 0,
      key.substructure: {
        key.nameoffset: 0,
        key.namelength: 3
      }
    },
    <...>
  ]
}
```

Использование SourceKit

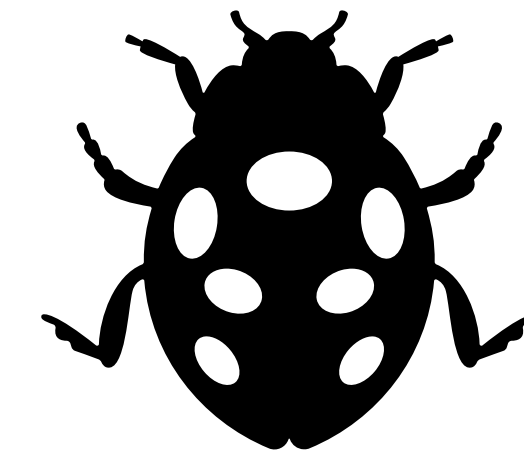
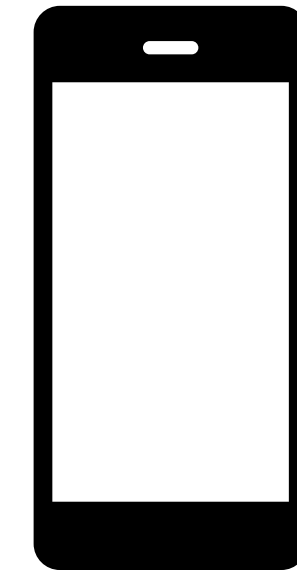
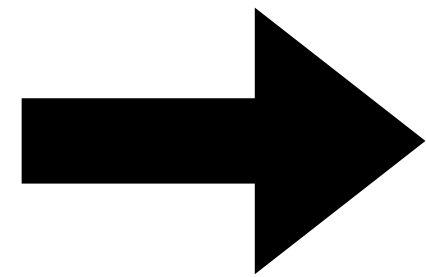
Автодополнение

```
3 func foo() {  
4   let vc: UIViewController? = nil  
5   vc.  
Void addChild(childController: UIViewController)
```

Immutable value 'vc' was never used; consider replacing with '_' or removing it

```
key.kind: source.lang.swift.decl.function.method.instance,  
key.name: "addChild(:)",  
key.sourcetext: "?.addChild(<#T##childController: UIViewController##UIViewController#>)",  
key.description: "addChild(childController: UIViewController)",  
key.typename: "void",  
key.modulename: "UIKit.UIViewController",  
key.context: source.codecompletion.context.thisclass,  
key.num_bytes_to_erase: 1,  
key.associated_usrs: "c:objc(cs)UIViewController(im)addChildViewController:"
```

Swift
Parsing
PSI
Intentions
Inspections
SourceKit





Знакомство и история AppCode



Поддержка языков программирования



Сборка, запуск и отладка iOS приложения

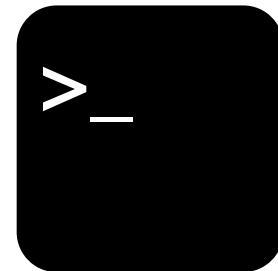


Команда и процессы

Сборка

xcodebuild

```
$ xcodebuild build -workspace MyApp.xcworkspace -scheme prod
```



Сборка

xcodebuild

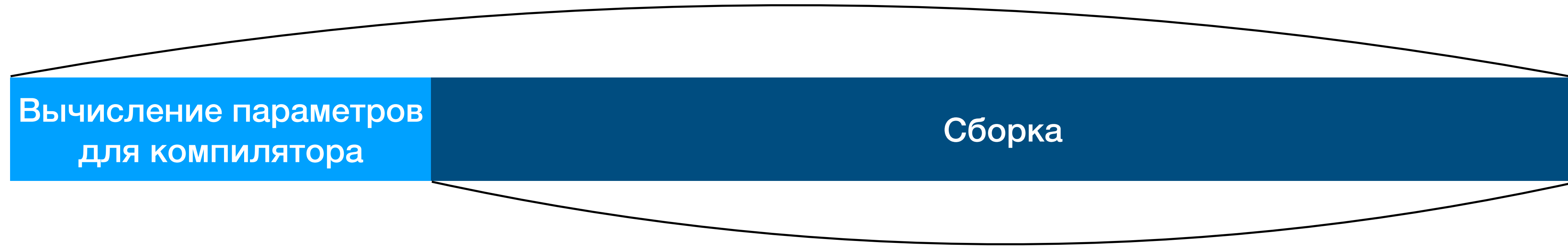
xcodebuild

Вычисление параметров
для компилятора

Сборка

Xcode

AppCode



Сборка

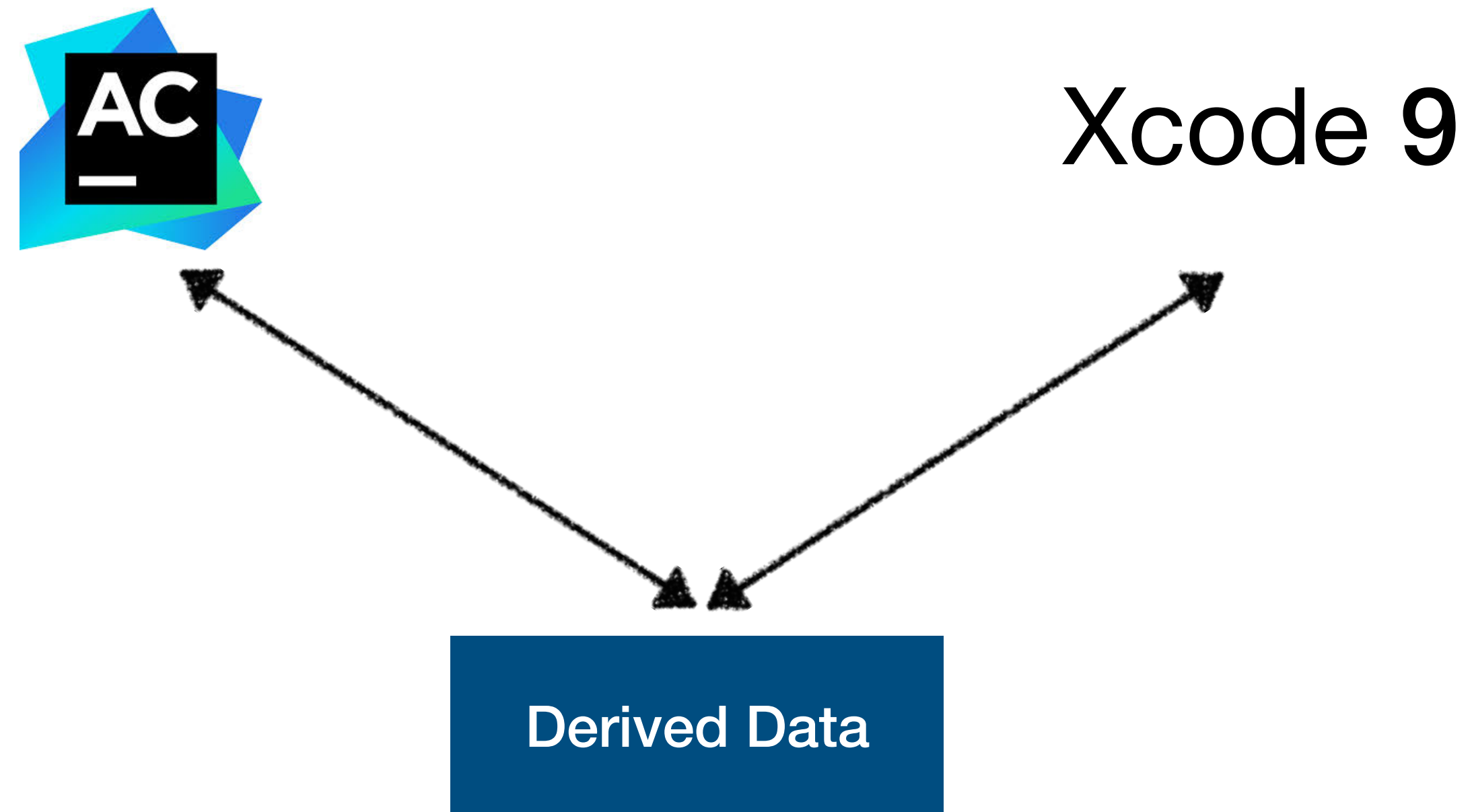
Derived Data



Derived Data

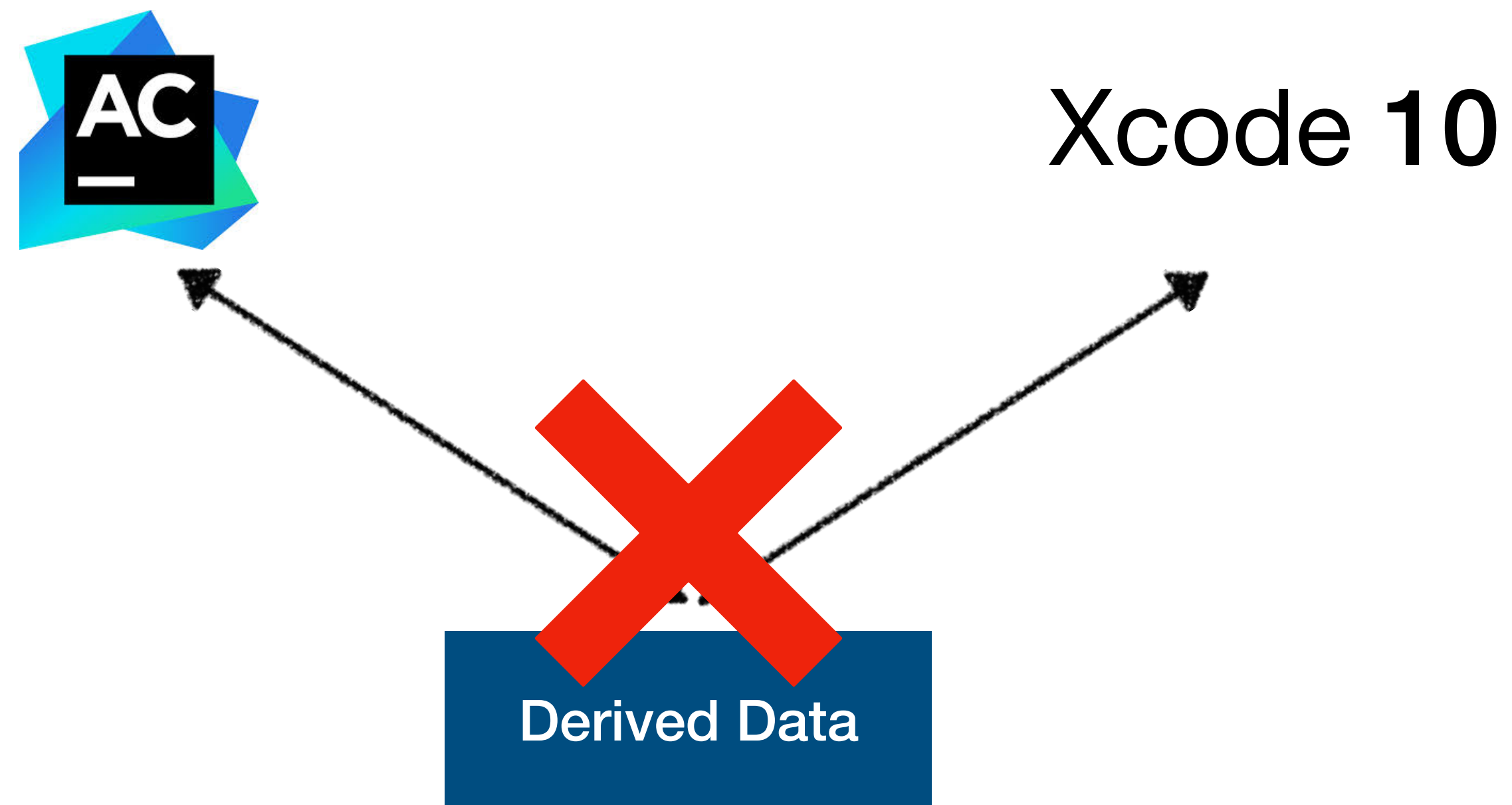
Сборка

Derived Data



Сборка

Derived Data



“This is prohibited to prevent build state inconsistencies.”

Сборка

Derived Data



Запуск на симуляторе

`simctl`

```
$ xcrun simctl list
```



Запуск на симуляторе

Подход AppCode

CoreSimulator.framework

CoreSimulator.framework/Resources/bin/**simctl**

Запуск на симуляторе

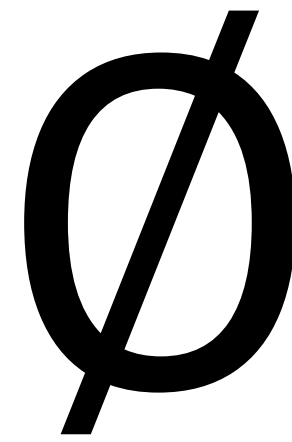
idb



FB SimulatorControl

<https://github.com/facebook/idb>

Запуск на устройстве



Запуск на устройстве

`MobileDevice.framework`

Запуск на устройстве

idb



FBDeviceControl

<https://github.com/facebook/idb>

Отладка

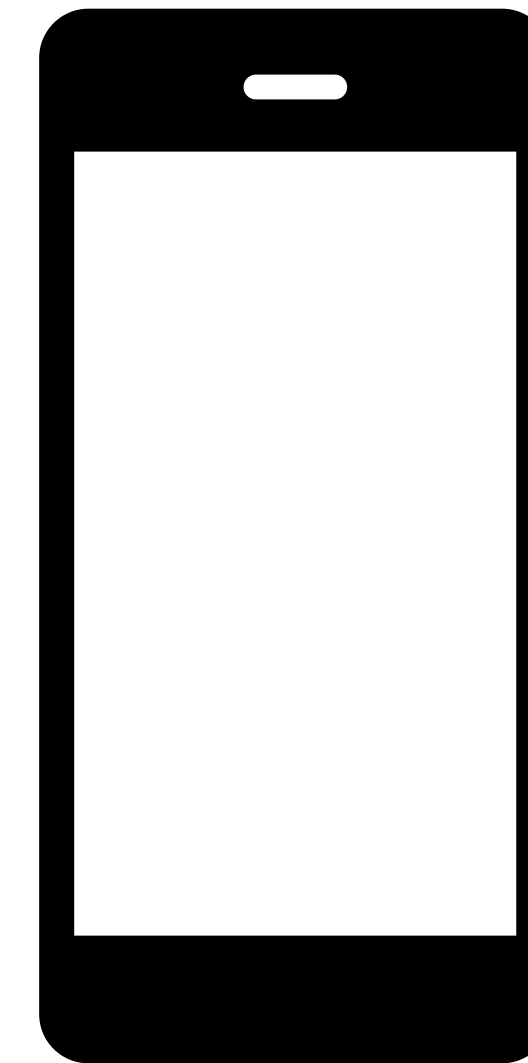
На симуляторе

LLDB

Отладка

На устройстве

LLDB



Отладка

На устройстве

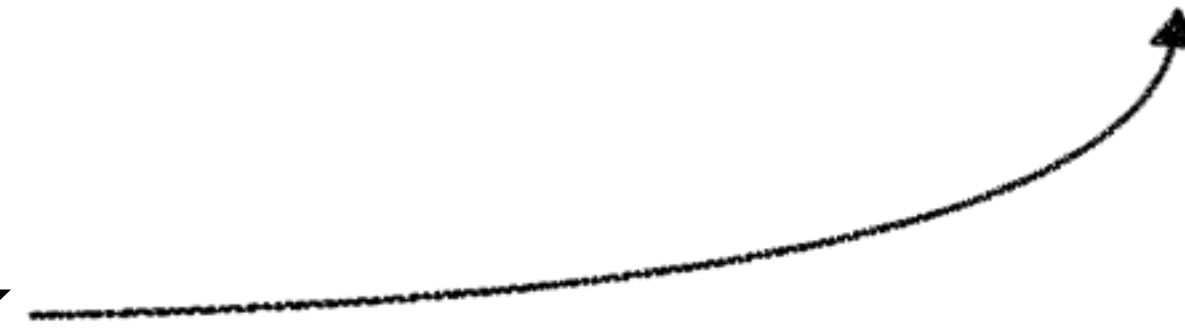
`/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/DeviceSupport/13.4/`

DeveloperDiskImage.dmg

Use for Development

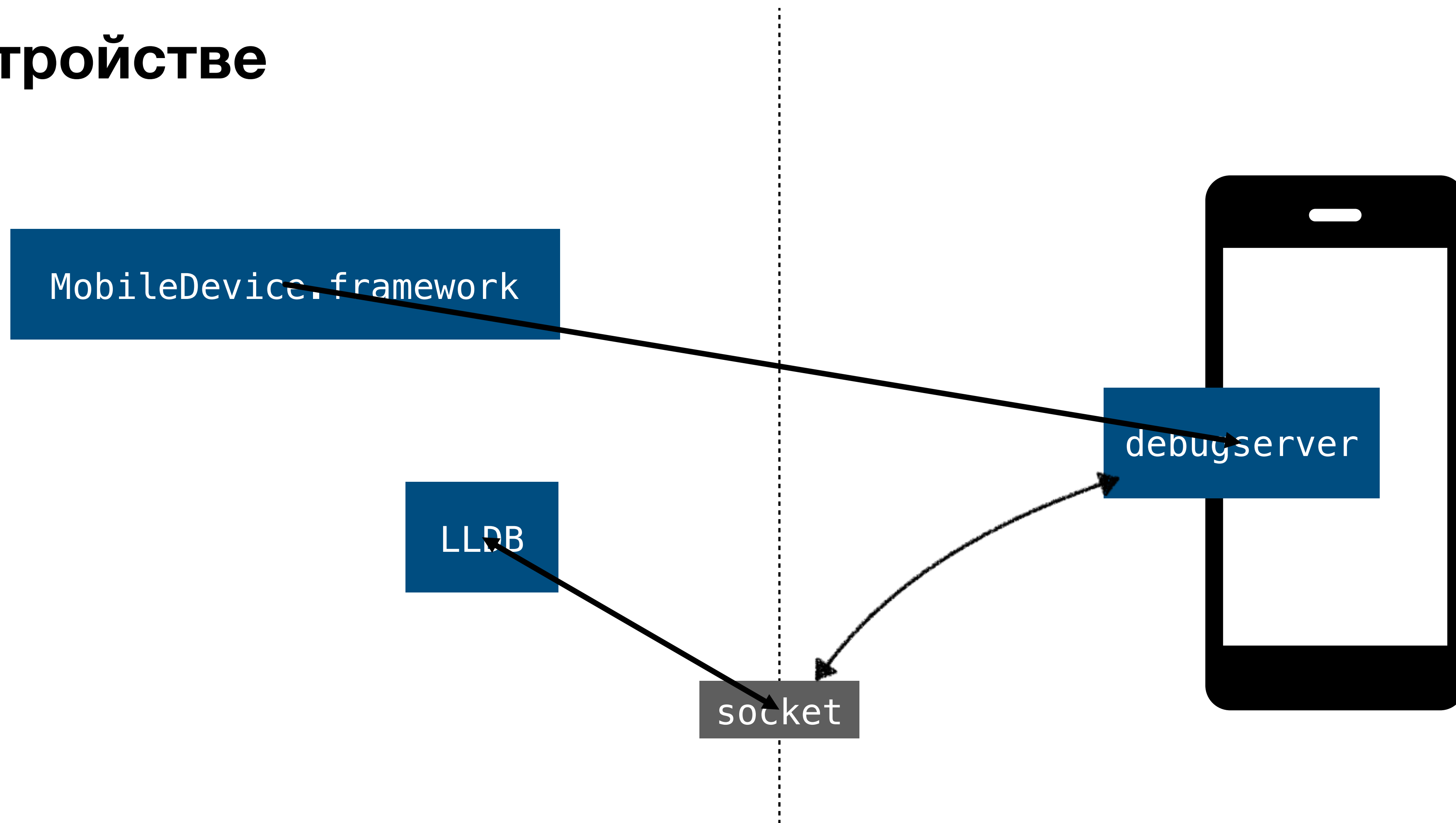
`/usr/bin/debugserver`

`MobileDevice.framework`



Отладка

На устройстве





Знакомство и история AppCode



Поддержка языков программирования



Сборка, запуск и отладка iOS приложения



Команда и процессы

Как разрабатывают AppCode

Kotlin, Java, C++, Objective-C

IDEA, AppCode, CLion

Monorepo

3 релиза в год

YouTrack

UpSource

TeamCity

Как разрабатывают AppCode

Тесты, тесты, тесты

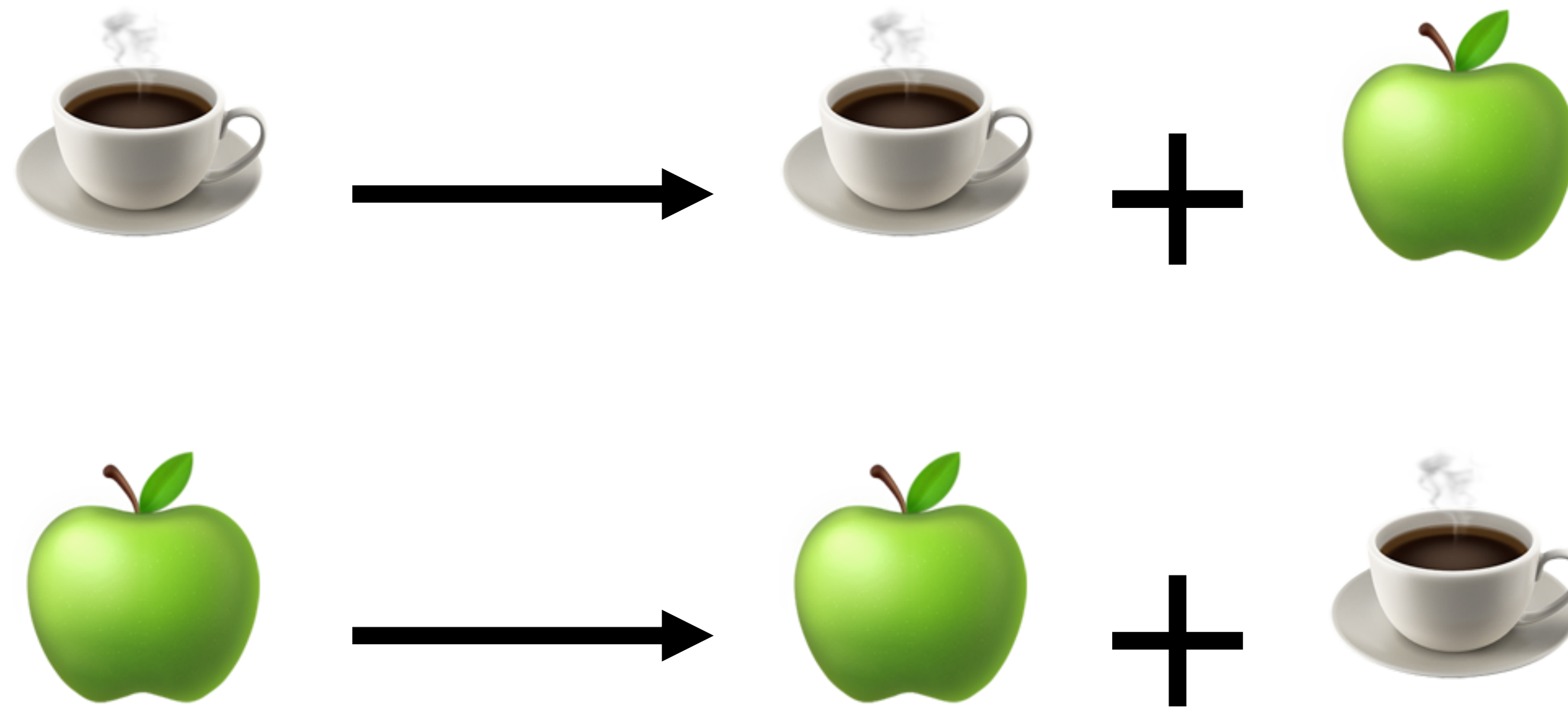
TeamCity duty

Push to master

Релизные ветки для EAP (Early Access Program)

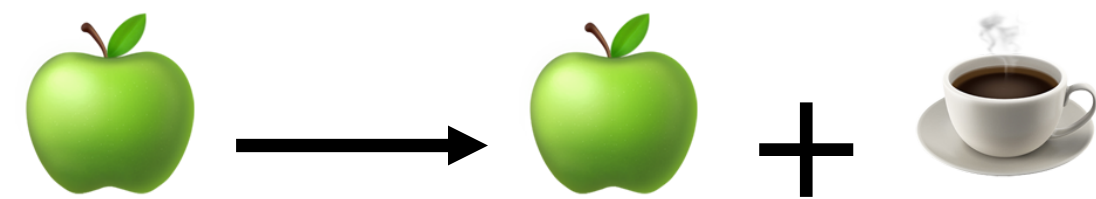
Кто разрабатывает AppCode

Пути некоторых разработчиков



Кто разрабатывает AppCode

Переход из iOS-разработки: личный опыт



Большой проект

Длиннее релизный цикл

Большие исследовательские задачи

`null` 🤔 `nil`

Кто разрабатывает AppCode

Пути некоторых тестировщиков

Опыт в ручном тестировании

Способность вести проект самостоятельно

Кто разрабатывает AppCode

Как с нами общаться

<https://youtrack.jetbrains.com/issues/OC>

cocoa-developers.slack.com —> #jetbrains

Twitter: **@appcode**

Знакомство и история AppCode

- Знакомство
- История AppCode

Поддержка языков программирования

- Работа с кодом
- Поиск декларации элемента
- Использование SourceKit

Сборка, запуск и отладка iOS приложения

- Сборка
- Запуск на симуляторе
- Запуск на устройстве
- Отладка

Команда и процессы

- Как разрабатывают AppCode
- Кто разрабатывает AppCode

Заключение

Делать IDE сложно, но можно

В AppCode нет черной магии

Альтернативный путь развития iOS разработчика

Каждый может помочь сделать AppCode лучше

Всем спасибо!

<https://youtrack.jetbrains.com/issues/OC>

[#jetbrains](https://cocoa-developers.slack.com)

Twitter: [@appcode](#), [@aydarmukh](#)