VK Карты.
Как жить с двумя провайдерами карт в крупном проекте

Konstantin Kulakov
iOS product team lead

Константин Кулаков

Youla
iOS Product Team Lead
Преподаватель VK Образования

VK

Telegram

юла

План

- Путь к внедрению VK Карт
- А что было до VK карт?
- Проблемы использования 2-х карт в проекте
- Основные подходы
- Корнер-кейсы во время внедрения
- Как внедрить к себе в проект? -
  Бесплатное использование на **iOS** и Android

Зачем нам карты?

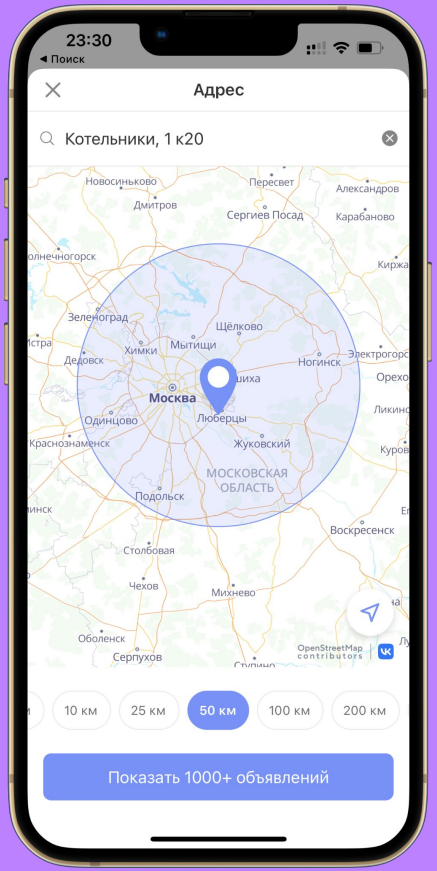
Сервис онлайн-объявлений «Юла» – работает на основе определения местоположения

А где используются карты?

- Выбор адреса на главной
- Местоположение пользователя в поиске
- Выбор метро и адреса
- Адрес в карточке продукта
- Местоположение магазина на карте
- Точки выдачи в доставке

Карты

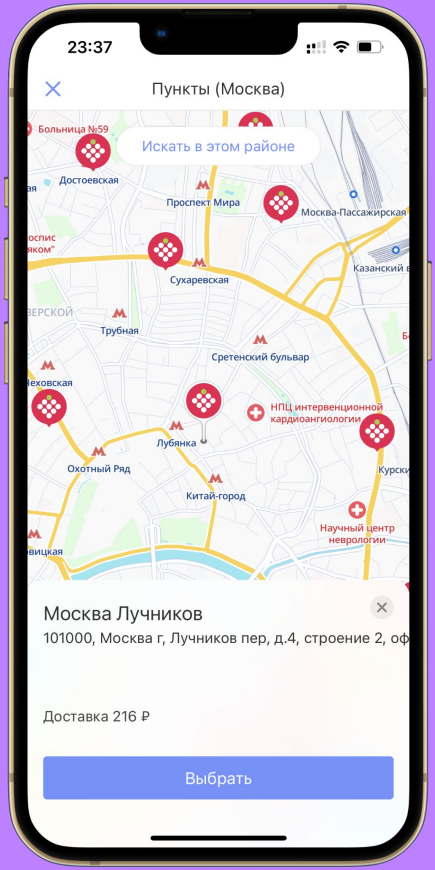
Выбор местоположения



юла

Карты

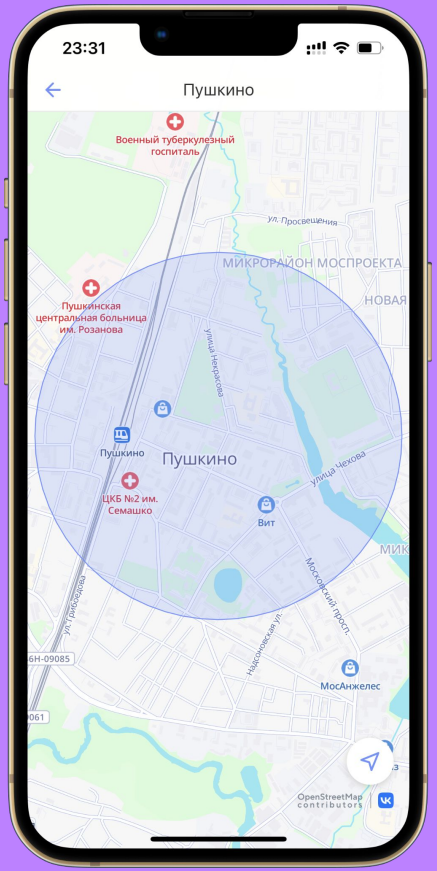
Выбор пунктов доставки



юла

Карты

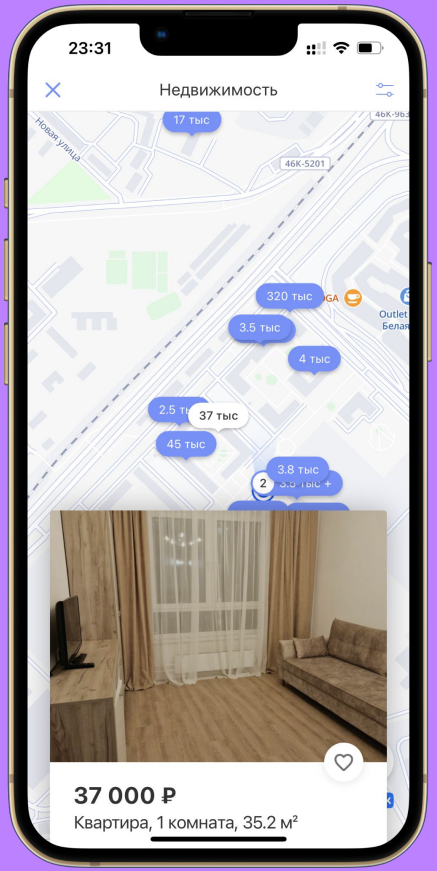
Просмотр
местоположения товара



юла

Карты

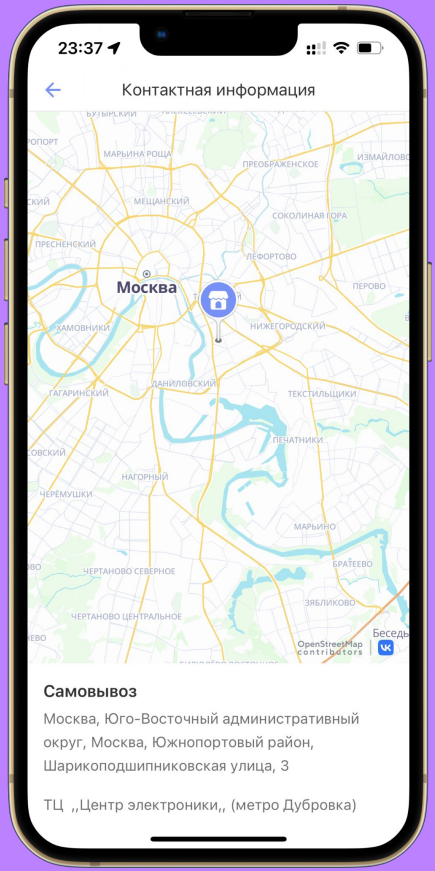
Недвижимость



юла

Карты

Местоположение магазина



юла

Задача по внедрению

VK Карты + Google карты

Задача по внедрению

Использование двух видов карт

«В идеале нам нужно инкапсулировать реализацию карт, чтобы при изменениях подобного плана мы подменяли карту внутри нашей обёртки без изменения клиентского кода. Обертка должна быть в отдельном модуле, а модули её использующие должны инжектить в себя эту зависимость» — **Техлид. 2022 г. н.э.**

Плюсы

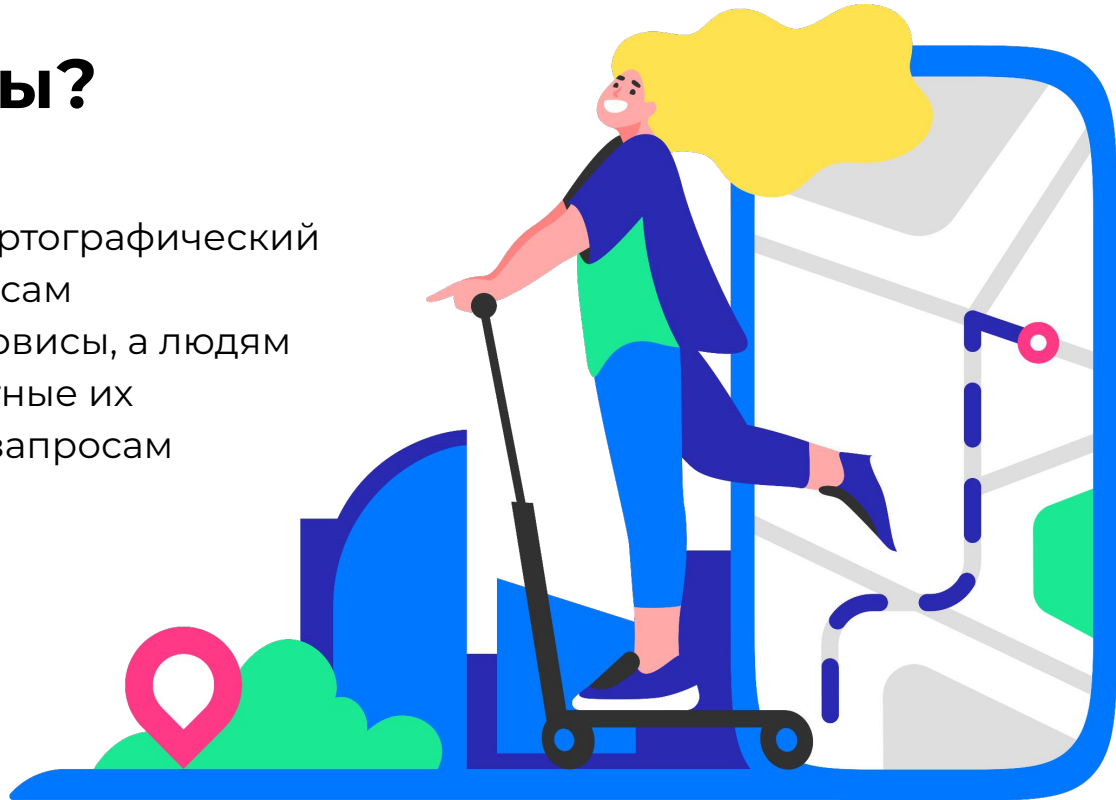
- Соответствуем законодательству
- Можем тестировать аб-гипотезы
- Можем добавлять других поставщиков карт без изменения кода проекта

Минусы

- Поддержка сразу 2-х провайдеров карт
- Отсутствует прозрачность

Ого, а что за карты?

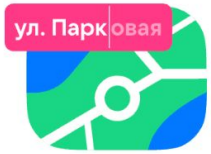
VK Карты — наш собственный картографический продукт, который позволят бизнесам использовать геоданные и геосервисы, а людям получать предложения, релевантные их местонахождению и поисковым запросам



VK Карты

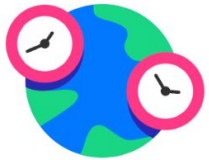
Поиск мест интереса

Сервис геокодирования



Подсказчик адреса

IP2GEO



Определение часового пояса

Поиск по почтовому индексу

Стек проекта

Стек проекта

Зависимости

- CocoaPods
- SPM

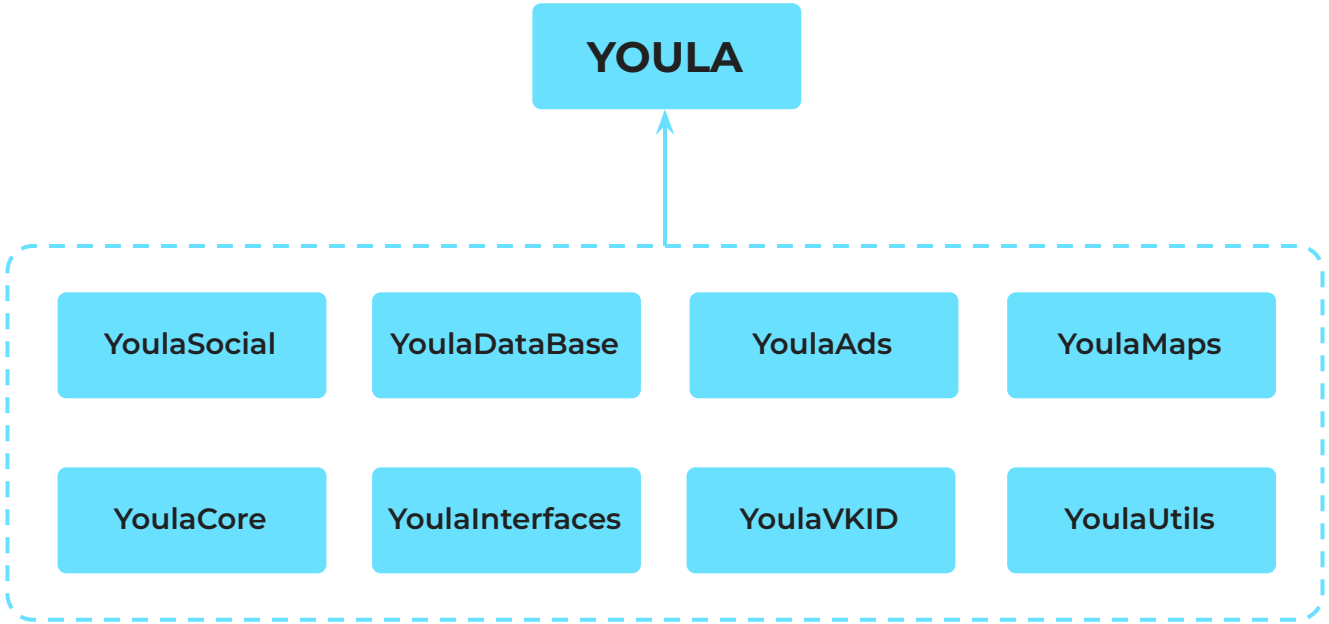
Карты

- Кластеризация
- Маркеры
- Расчёт расстояния
- Генерация баблов
- Местоположение пользователя

Стек проекта

юла

Наш путь. Модульность

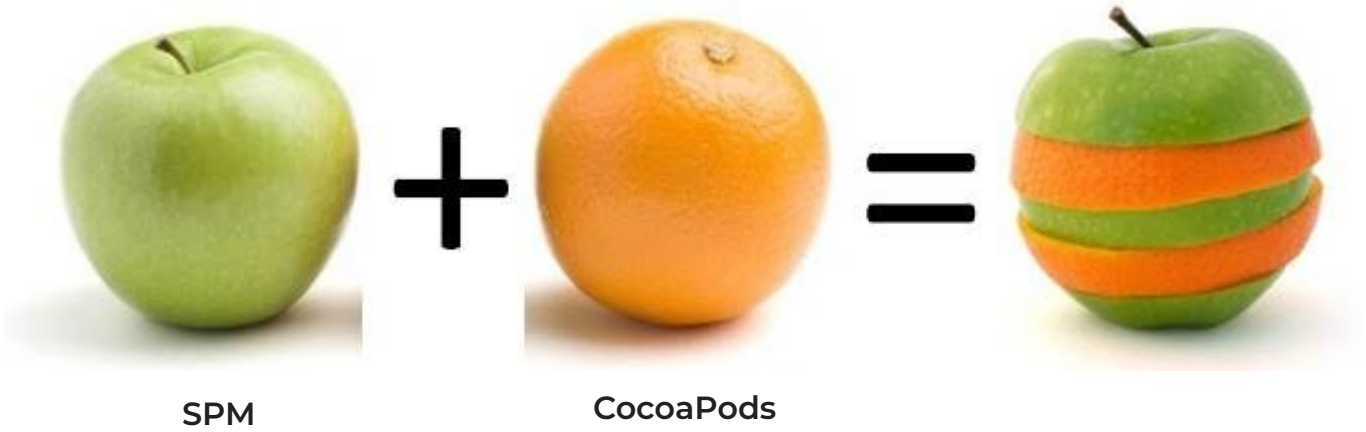


Структура модуля

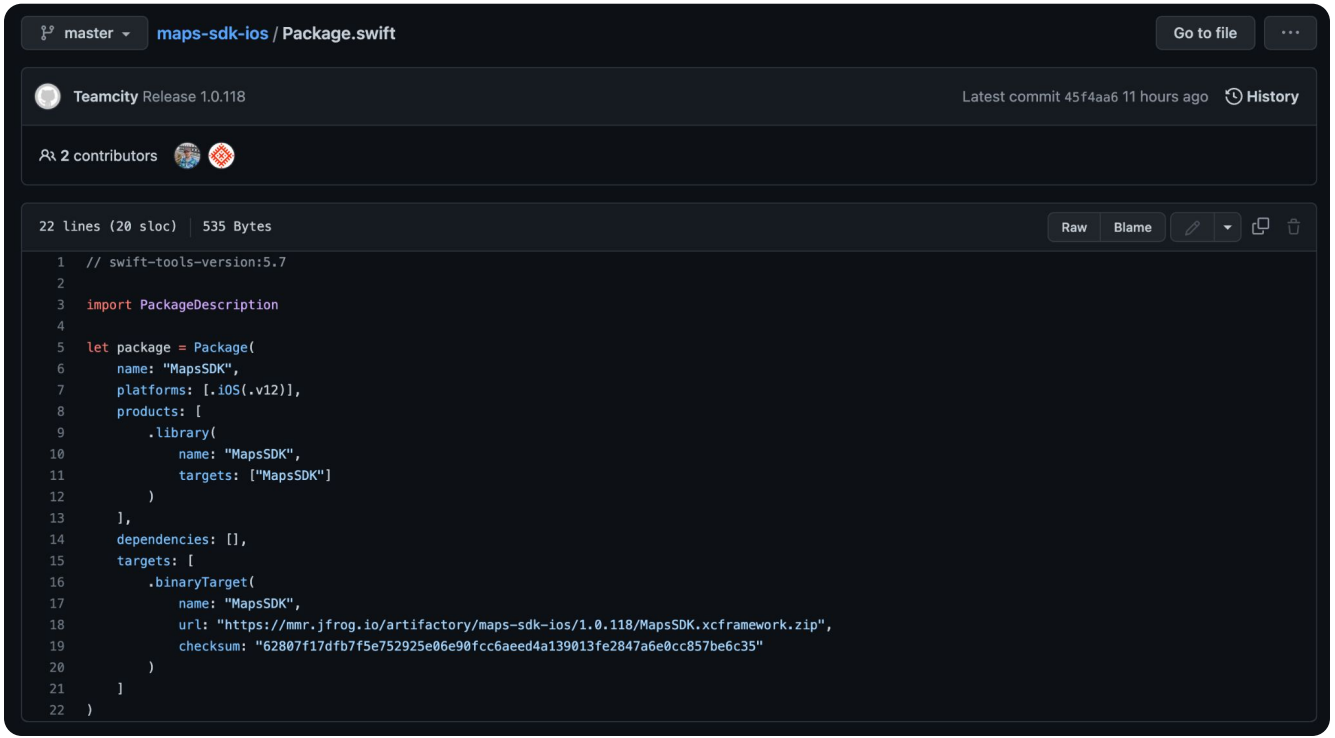
Первые проблемы

Google Maps — **CocoaPods**

VK Карты — **SPM**



Создание пакета CocoaPods



Создание пакета CocoaPods

VKMaps.podspec — Изменено

```
Pod::Spec.new do |s|
  s.name = 'VKMaps'
  s.version = '1.0.118'
  s.summary = 'VKMaps'
  s.authors = 'VK.com'

  s.homepage = 'https://github.com/maps-mailru/maps-sdk-ios/blob/master/Package.swift'
  s.license = { :type => 'Copyright (c) 2022 - present, LLC “V Kontakte”', :text => '' }

  s.ios.deployment_target = '12.4'
  s.swift_version = '5.5'
  s.cocoapods_version = '>= 1.9.0'

  s.vendored_frameworks = 'MapsSDK.xcframework'
  s.source = {
    :http => "https://mmr.jfrog.io/artifactory/maps-sdk-ios/1.0.118/MapsSDK.xcframework.zip",
    :sha256 => "62807f17dfb7f5e752925e06e90fcc6aeed4a139013fe2847a6e0cc857be6c35"
  }
end
```

Создание пакета CocoaPods

SPM

```
targets: [
    .binaryTarget(
        name: "MapsSDK",
        url: "https://mmr.jfrog.io/artifactory/maps-sdk-ios/1.0.118/MapsSDK.xcframework.zip",
        checksum: "62807f17dfb7f5e752925e06e90fcc6aeed4a139013fe2847a6e0cc857be6c35"
    )
]
```

CocoaPods

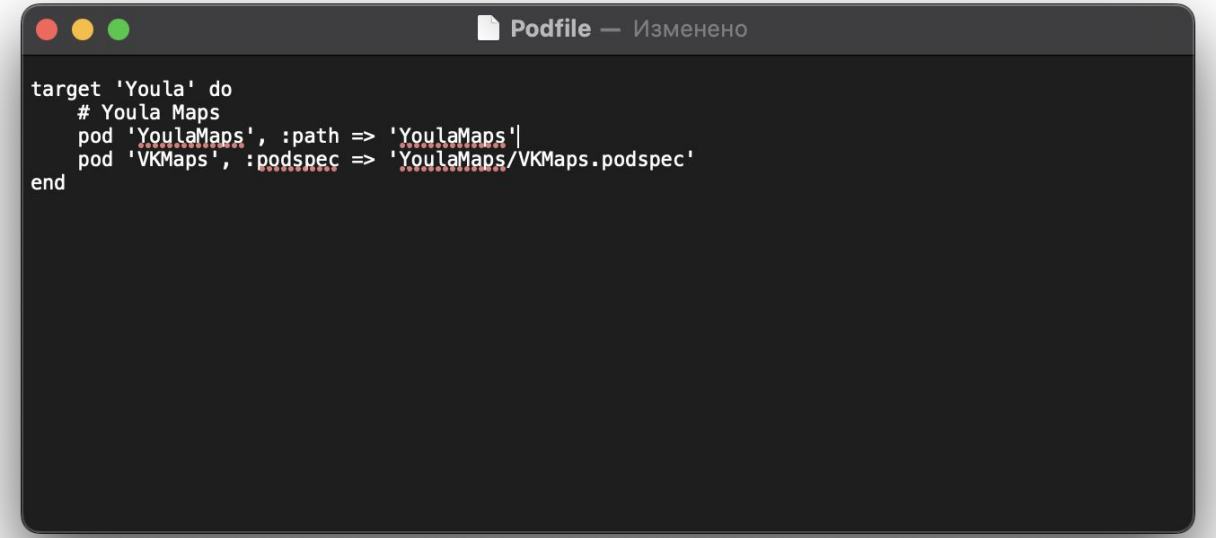
```
s.source = {
  :http => "https://mmr.jfrog.io/artifactory/maps-sdk-ios/1.0.118/MapsSDK.xcframework.zip",
  :sha256 => "62807f17dfb7f5e752925e06e90fcc6aeed4a139013fe2847a6e0cc857be6c35"
}
```

Создание модуля CocoaPods

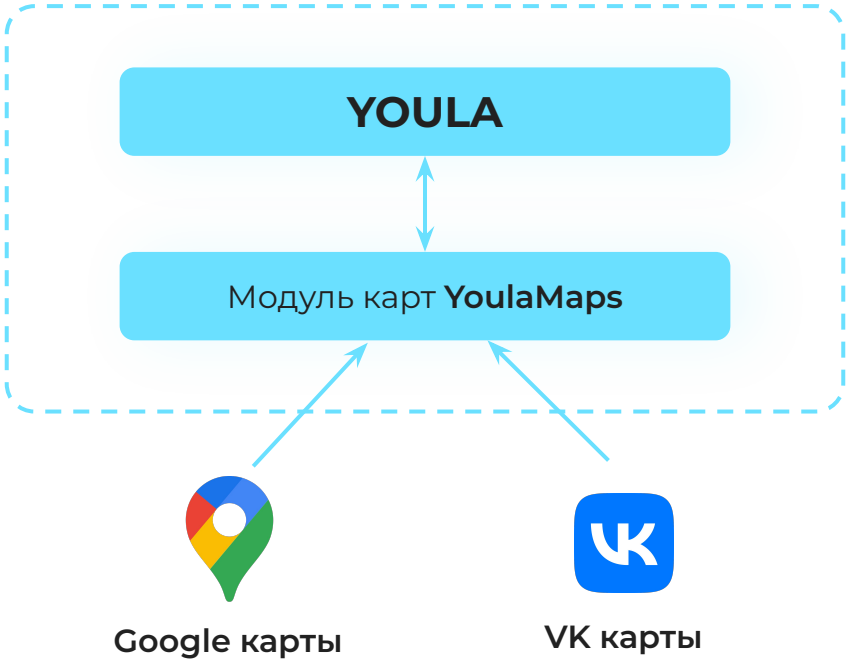
YoulaMaps.podspec — Изменено

```
Pod::Spec.new do |s|
  s.name             = "YoulaMaps"
  s.version          = "1.0.3"
  s.summary          = "Youla Maps Integration"
  s.description      = <<-DESC
  Youla Maps (Google + VK Maps Libs).
                       DESC
  s.license          = 'MIT'
  s.author           = { "Youla Team" => "k.kulakov@corp.mail.ru" }
  s.source           = { 'http' => '' }
  s.homepage         = "https://github.com/youla-dev"
  s.swift_versions   = '5.5'
  s.platform         = :ios, '12.4'

  s.ios.deployment_target = '12.4'

  s.ios.dependency 'GoogleMaps', '3.1.0'
  s.ios.dependency 'Google-Maps-iOS-Utils', '3.4.0'
  s.ios.dependency 'VKMaps'
  s.ios.dependency 'YInterfaceKit'

  s.source_files = 'YoulaMaps/Classes/**/*.{swift}'
  s.exclude_files = ['YoulaMaps.podspec']

  s.static_framework = true
end
```

Подключение к монолиту



Структура модуля



Что хотим?

Фабрику — на вход нужного провайдера,
на выход — view и ручки для управления

```
let mapInput = YoulaMapsFactory.map(with: provider, delegate: self)
self.mapInput = mapInput

view.addSubview(mapInput.view)
```

А что нужно для этого?

```
public protocol YoulaMapsFactoryDescription {
    static func map(with provider: YoulaMapsProvider, delegate: YoulaMapDelegate?) -> any YoulaMapDescription
}
```

```
public protocol YoulaMapDescription {
    associatedtype YoulaMapView: UIView

    var view: YoulaMapView { get }

    ....
}
```

Непрозрачные типы результатов (SE-0328)

Any придется писать везде для протоколов, где во время компиляции нельзя будет определить размер памяти под объект

Что получаем в итоге?

Maps Provider

- VK
- Google

Выбор провайдера для получения карты

YoulaMaps

Maps Provider

Public

Private

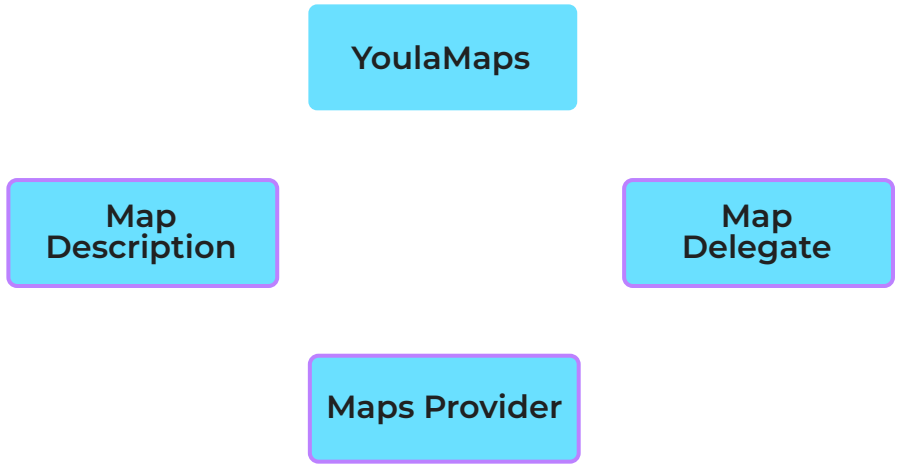
юла

Что получаем в итоге?

Protocols

- **Description** — управление (addMarker, ...)
- **Delegate** — события (didTapCoordinate, ...)

Обработка входящих событий и испускание исходящих



Public

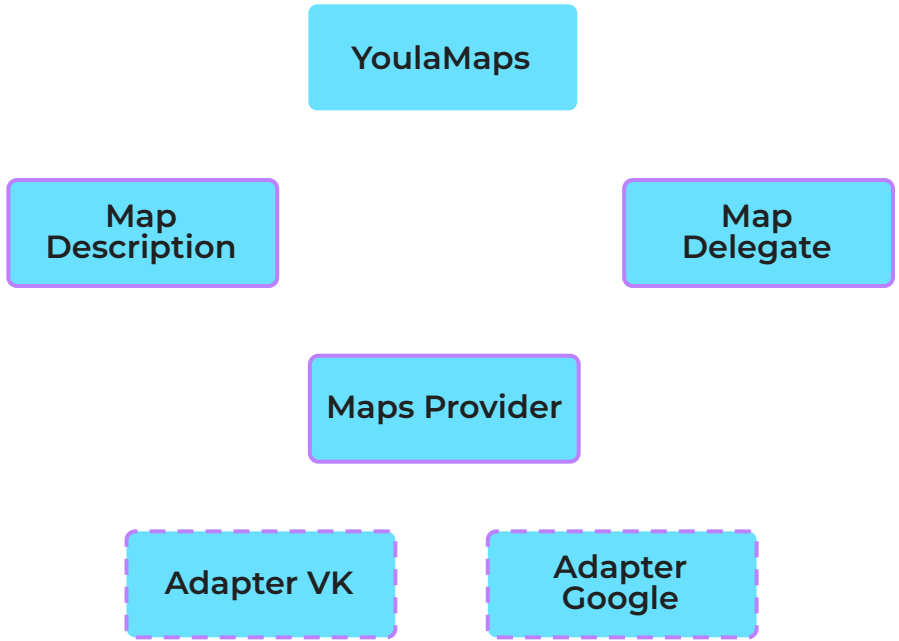
Private

Что получаем в итоге?

Adapters

- VK
- Google

Адаптеры конкретного провайдера



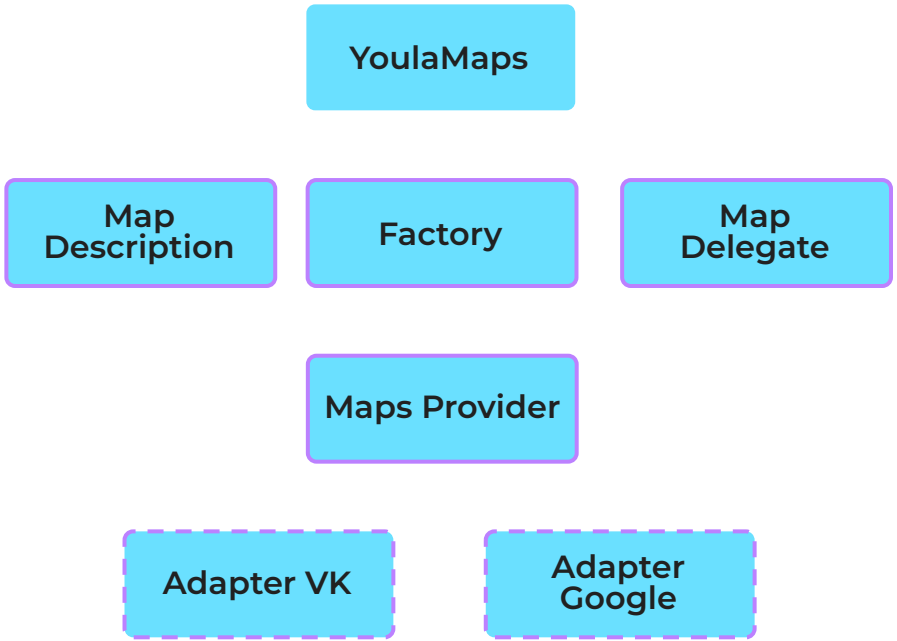
Public

Private

Что получаем в итоге?

Factory

Фабрика для создания карты
с конкретным адаптером



Public

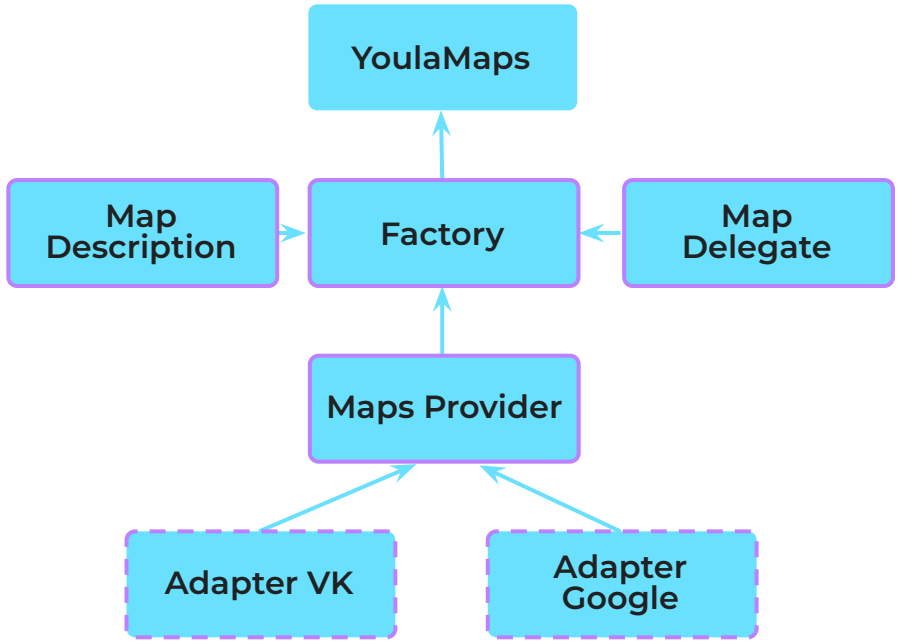
Private

Что получаем в итоге?

Youla Maps

Модуль для взаимодействия с картой, инкапсулированный единым протоколом.

Пользователь (программист) ничего не знает о провайдерах



Public

Private

Провайдеры

Сравнение и решение проблем

Сравним провайдеров

```
extension VKMapsProviderAdapter: MapViewDelegate {
    ...
}
```

```
extension GoogleMapsProviderAdapter: GMSMapViewDelegate {
    ...
}
```

Сравним провайдеров

```swift
public func mapViewSnapshotReady(_ mapView: GMSMapView) {
    guard !didLoadMap else {
        return
    }

    didLoadMap = true

    delegate?.didLoadMap()
}
```

```swift
public func mapViewDidLoad(_: MapView) {
    didLoadMap = true
    delegate?.didLoadMap()
}
```

А зачем didLoadMap?

Google — исключение повторных вызовов
VK — поздняя инициализация

```
public func configure(with config: YoulaMapConfig, animated: Bool) {
    guard config.location.horizontalAccuracy >= 0 else {
        return
    }

    let coordinates = Coordinates(lng: config.location.coordinate.longitude,
                                  lat: config.location.coordinate.latitude)

    let zoom: Double = Double(config.zoom)

    if !didLoadMap {
        loadMap(with: coordinates, zoomLevel: zoom)
    } else {
        updateLocation(with: coordinates, zoomLevel: zoom, animated: animated)
    }
}
```

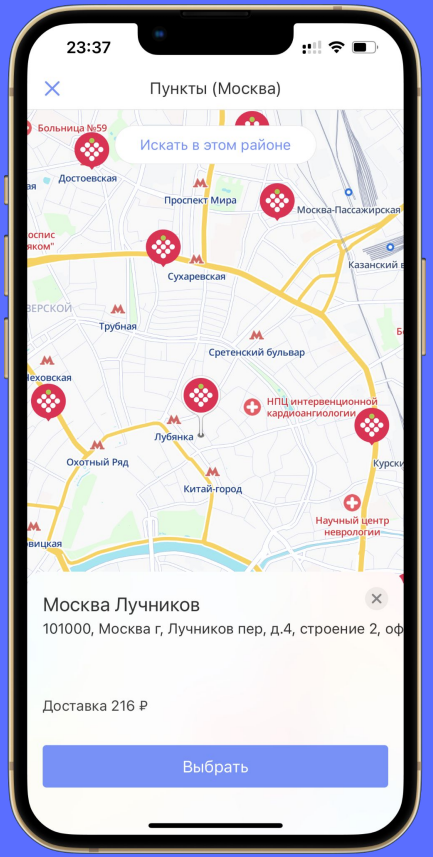
Сравним провайдеров

```
public func mapView(_ mapView: GMSMapView, didTap marker: GMSMarker) -> Bool {
    delegate?.didTapMarker(at: marker.position)

    return true
}
```

```
public func mapView(_: MapView, didSelectMarkerID id: String) {
    guard let marker = disposedBag.get(markerId: id) else {
        return
    }

    let coordinate = CLLocationCoordinate2D(latitude: marker.coords.lat,
                                            longitude: marker.coords.lng)

    delegate?.didTapMarker(at: coordinate)
}
```

Предыстория

Маркеры и DisposedBag

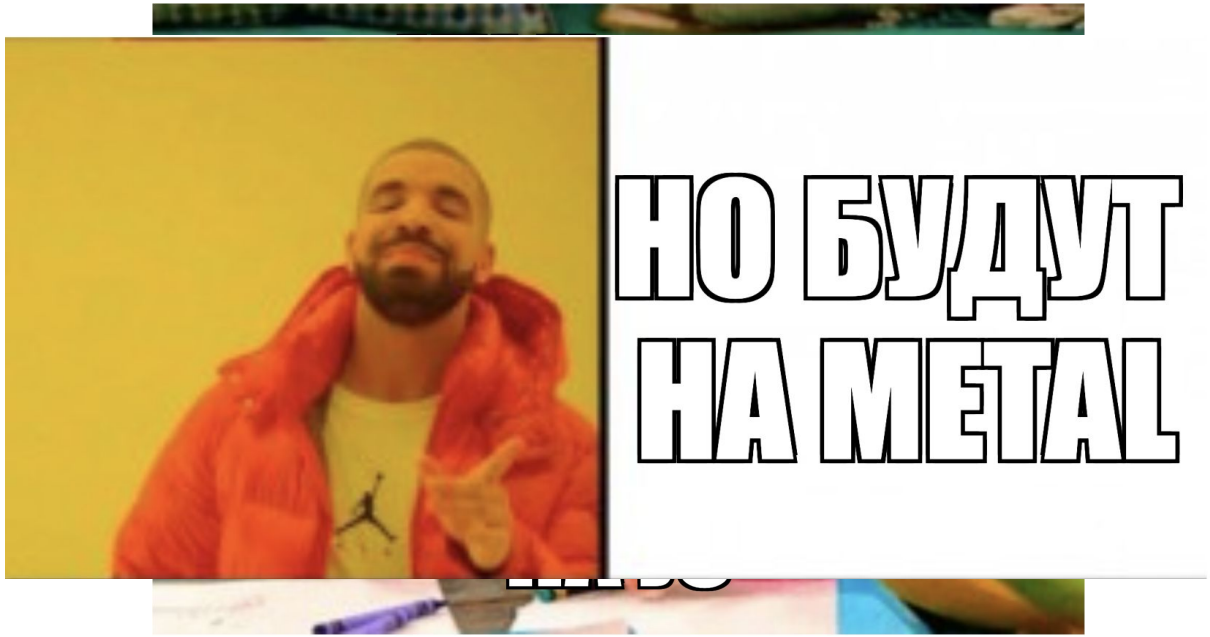


юла

DisposedBag и очистка карты

DisposedBag и очистка карты

Очистка карты и добавление новых маркеров — **асинхронны**



DisposedBag и очистка карты

```
protocol VKMapsDisposedBagDescription {
    func removeAll()
    func remove(markerIds: [String])
    func append(marker: Marker)
    func get(markerId: String) -> Marker?
    func getMarkers() -> [Marker]
    func append(layerId: String)
    func append(cluster: String)
}
```

DisposedBag и очистка карты

```
final class VKMapsDisposedBag: VKMapsDisposedBagDescription {
    private var markers: [Marker] = []
    private var layersIds: [String] = []
    private var clusters: [String] = []
    private weak var mapView: MapView?

    init(mapView: MapView) {
        self.mapView = mapView
    }
```

DisposedBag и очистка карты

```
func removeAll() {
    markers.forEach { marker in
        mapView?.removeMarker(id: marker.id)
    }

    layersIds.forEach { id in
        mapView?.removeLayer(id: id)
    }

    clusters.forEach { id in
        mapView?.removeCluster(id: id)
    }

    markers.removeAll()
    layersIds.removeAll()
    clusters.removeAll()
}
```

```
func remove(markerIds: [String]) {
    markerIds.forEach { markerId in
        mapView?.removeMarker(id: markerId)
        markers.removeAll(where: { $0.id == markerId })
    }
}

func append(marker: Marker) {
    markers.append(marker)
}

func get(markerId: String) -> Marker? {
    return markers.first { marker in
        marker.id == markerId
    }
}
```

Функциональность карт

Оба провайдера покрывают наши задачи, но имеются различия

VK Карты

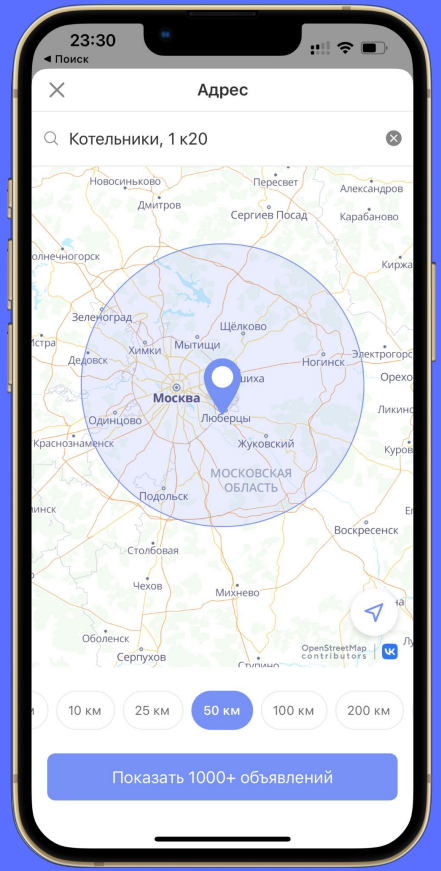
- Кластеризация
- Маркеры
- **Нет расчёта расстояния**
- **Северо-восток и юго-запад**

Google Maps

- Кластеризация
- Маркеры
- Расчёт расстояния
- Любое направления координат

Расчет расстояния

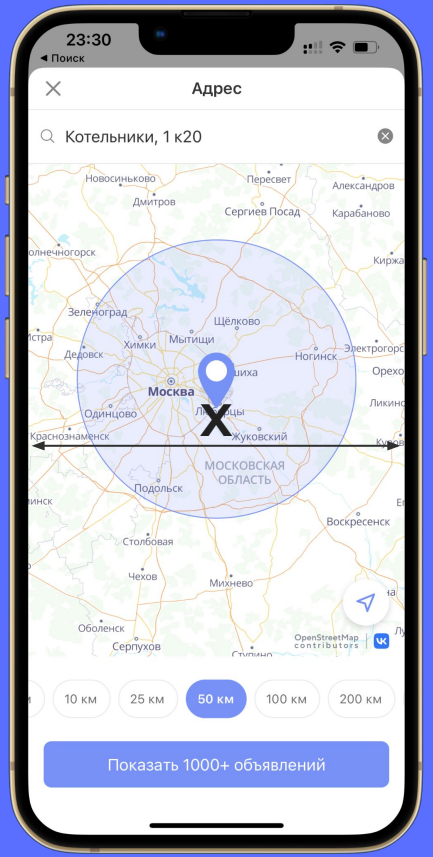
Выбор радиуса поиска
товаров от 10 до 200 км



юла

А как считать?

- **X** — метров



юла

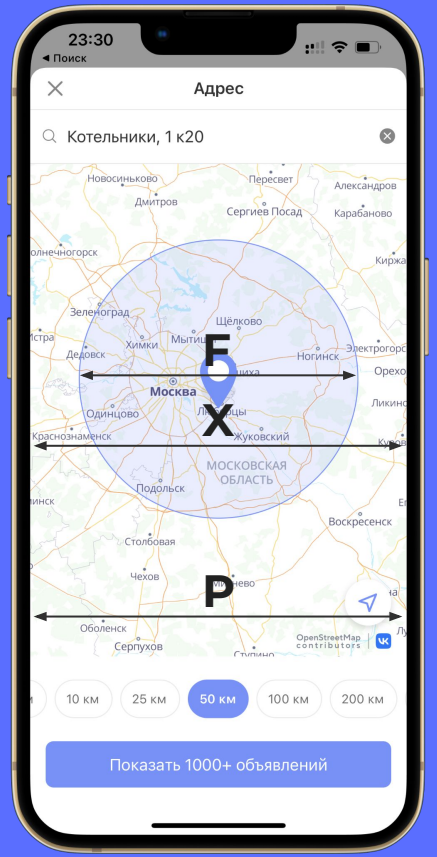
А как считать?

- **X** — метров
- **P** — пикселей

X / P = L — кол-во метров в пикселе

D — выбранное число метров поиска товаров

F = D / L — Круг в пикселях



юла

Расчет расстояния. Google

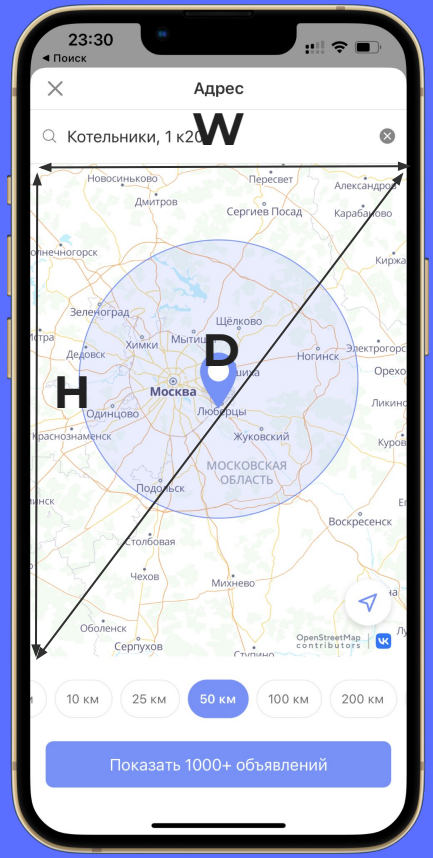
```
public var visibleDiameter: CLLocationDistance {
    let centerLeft = CGPoint(x: 0, y: view.bounds.height / 2)
    let centerLeftCoordinate = view.projection.coordinate(for: centerLeft)

    let centerRight = CGPoint(x: view.bounds.width, y: view.bounds.height / 2)
    let centeRightCoordinate = view.projection.coordinate(for: centerRight)

    return GMSGeometryDistance(centerLeftCoordinate, centeRightCoordinate)
}
```

Расчет расстояния. VK Карты

```swift
public var visibleDiameter: CLLocationDistance {
    guard
        let southwest = view.mapBounds?.southwest,
        let northeast = view.mapBounds?.northeast
    else {
        return .zero
    }

    let bottomLeftLocation = CLLocation(latitude: southwest.lat, longitude: southwest.lng)
    let topRightLocation = CLLocation(latitude: northeast.lat, longitude: northeast.lng)

    let diagonal = bottomLeftLocation.distance(from: topRightLocation)

    let width = view.bounds.width
    let height = view.bounds.height
    let diagonalPexels = sqrt(pow(width, 2) + pow(height, 2))

    let kmInPexel = diagonal / diagonalPexels

    let diameter = kmInPexel * width

    return diameter
}
```

VK Карты алгоритм

- **D** — диагональ в метрах
- **W, H** — ширина и высота в пикселях
- **Dp = sqrt(w^2 + h^2)** — диагональ в пикселях (по теореме пифагора)
- **Mp = D / Dp** — метров в пикселе
- **Di = Mp * W** — диаметр в метрах

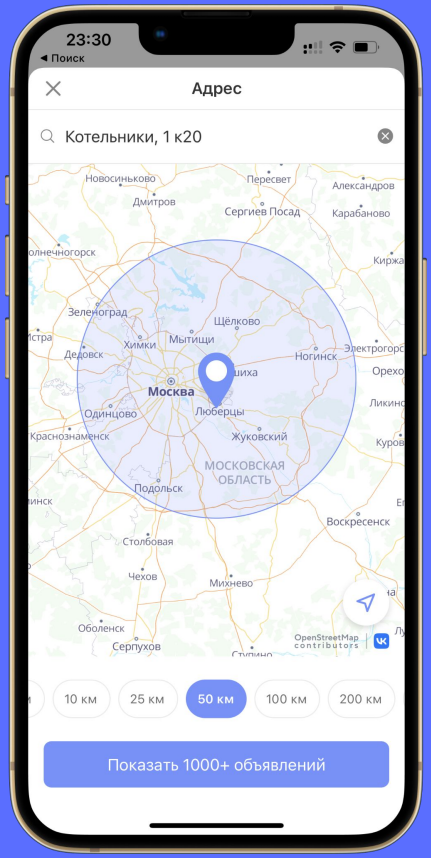


юла

VK Карты.
Как можно лучше?

Есть метод асинхронный —
можно использовать его.

Выпускают новый синхронный
метод через viewPoint



юла

Внедрение в objc экраны

Чтобы добавить новые методы objc-классу:

- Файл ObjcClassName+Extension.swift

```swift
extension AddressSelectMapVC: YoulaMapDelegate {

    public func didMoveMap(to coordinate: CLLocationCoordinate2D) {
        changeMapPosition(to: coordinate)
    }

    public func willMoveMap(isGesture: Bool) {
        mapWillMove(isGesture)
    }

    public func didTapCoordinate(at coordinate: CLLocationCoordinate2D) {
        mapDidTapCoordinate(at: coordinate)
    }

    public func didLoadMap() {
        setupRadiusCircleIfNeeded(animated: false)
    }
}
```

Внедрение в objc экраны

Чтобы добавить новые свойства objc-классу:

- Создайте файл **ObjcClassNameProperties.swift** и наследуйте класс ObjcClassNameProperties от NSObject;
- Внутри данного **ObjcClassNameProperties** инициализируйте новые свойства. Не забывайте помечать их атрибутом **@objc**, если они должны быть видны из Objective-C.
- Добавьте в **ObjcClassName.h** новое свойство **@property (nonatomic, strong) ObjcClassNameProperties *properties;**
- Инициализируйте properties в **ObjcClassName.m**

Внедрение в objc экраны

```
final class AddressSelectMapVCProperties: NSObject {
    var mapInput: (any YoulaMapDescription)?
}
```

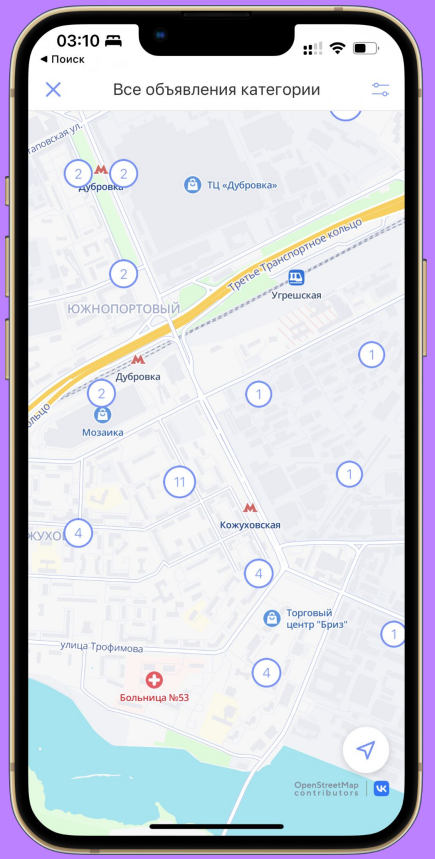
Внедрение в objc экраны

```swift
private func loadMap() {
    let provider: YoulaMapsProvider

    switch properties.youlaMapsHelper.provider {
    case .vk:
        let context: YoulaVKMapProviderContext = .init(apiKey: YoulaDefines.vkMapsApiKey(), options: [])
        provider = .vk(context: context)
    case .google:
        provider = .google
    }

    let mapInput = YoulaMapsFactory.map(with: provider, delegate: self)
    properties.mapInput = mapInput

    mapContainerView.addSubview(mapInput.view)
}
```

Кластеризация

Кластеризация

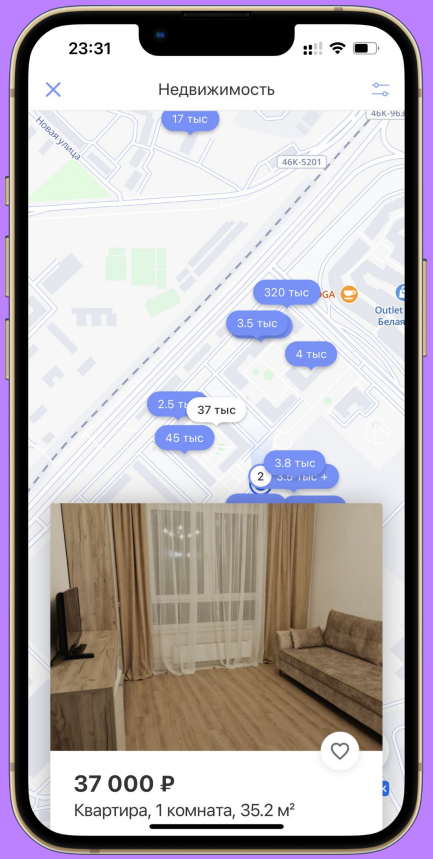
Закрытый кластер



юла

Кластеризация

Открытый кластер



юла

Генерация индивидуальных сносок

```swift
func icon(forSize size: UInt) -> UIImage! {
    guard let clusterImage = UIImage(named: "someName")  else {
        return UIImage()
    }

    let countText: String = size > 99 ? "99" : "\(size)"

    let targetSize = clusterImage.size

    UIGraphicsBeginImageContextWithOptions(targetSize, false, .zero)

    clusterImage.draw(in: CGRect(x: .zero,
                                 y: .zero,
                                 width: targetSize.width,
                                 height: targetSize.height))

    let font = UIFont.boldApplicationFont(withSize: Constants.fontSize)
    let attributes: [NSAttributedString.Key: Any] = [.font: font,
                                                     .foregroundColor: UIColor.red]
    let textRect: CGRect = countText.boundingRect(with: .zero,
                                                  options: .usesLineFragmentOrigin,
                                                  attributes: attributes,
                                                  context: nil)

    let countAttributedText = NSAttributedString(string: countText, attributes: attributes)

    countAttributedText.draw(at: CGPoint(x: targetSize.width / 2 - textRect.size.width / 2, y: 17))

    let resultClusterImage = UIGraphicsGetImageFromCurrentImageContext()
    UIGraphicsEndImageContext()

    return resultClusterImage
}
```

Генерация с помощью render

```swift
let myView = MyView(frame: .init(origin: .zero, size: .init(width: 100, height: 100)))

let format = UIGraphicsImageRendererFormat.default()
format.scale = UIScreen.main.scale

let image = UIGraphicsImageRenderer(size: myView.bounds.size, format: format).image { context in
    myView.layer.render(in: context.cgContext)
}
```

UIGraphicsImageRender.image

Возвращает **UIImage**, после выполнения **imageRenderBlock (actions)**

CALayer.render

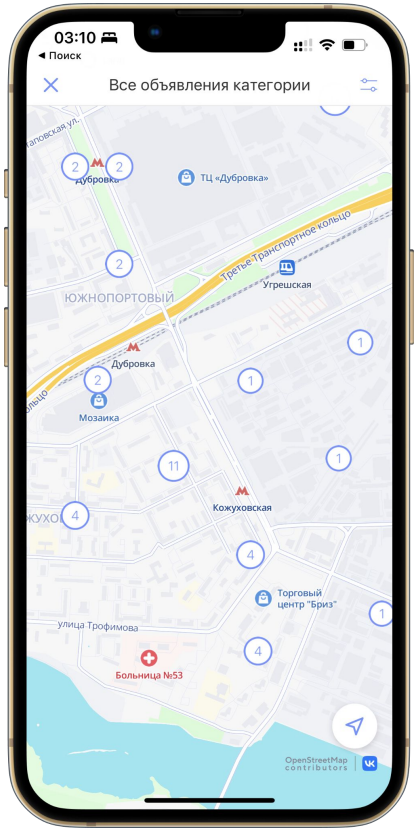
Отрисовывает слой и его дочернии слои в указанном контексте

Генерация индивидуальных сносок

```swift
@objc
public extension UIView {
    func toSnapshot() -> UIImage {
        UIImage.image(view: self)
    }
```

```swift
let iconView: UIView = properties.markerBuilder.markerView(forItem: cluster, with: type)
let icon: UIImage = iconView.toSnapshot()
```

А что на бекенде?

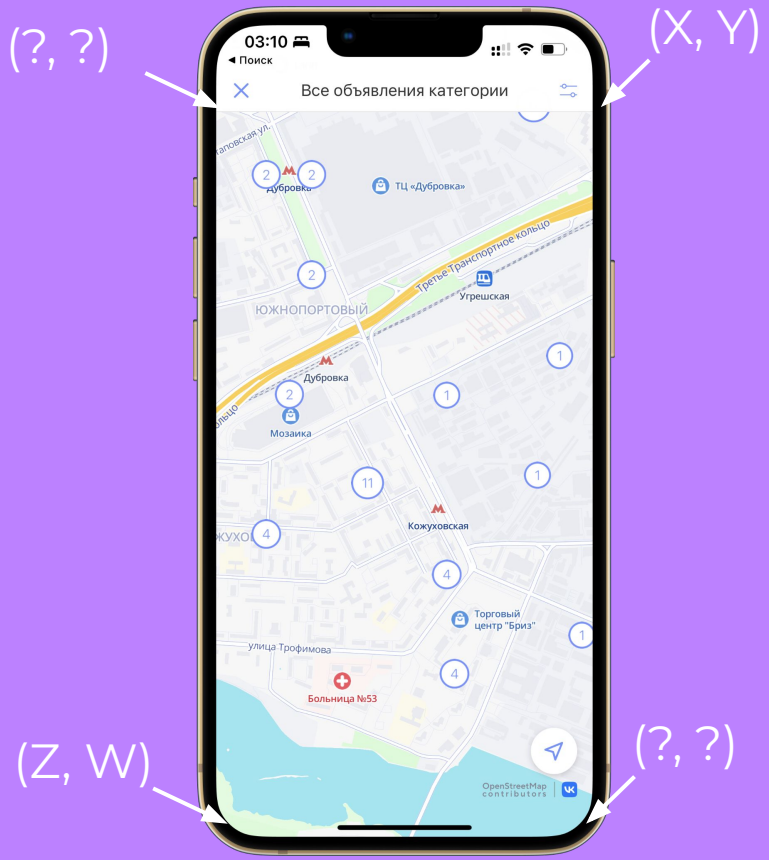


Новые координаты

Имеем: Северо-Восток и Юго-Запад

Хотим: Северо-Запад и Юго-Восток

```swift
public var topLeft: CLLocationCoordinate2D {
    guard
        let southwest = view.mapBounds?.southwest,
        let northeast = view.mapBounds?.northeast
    else {
        return CLLocationCoordinate2D()
    }

    return CLLocationCoordinate2D(latitude: northeast.lat,
                                  longitude: southwest.lng)
}

public var bottomRight: CLLocationCoordinate2D {
    guard
        let southwest = view.mapBounds?.southwest,
        let northeast = view.mapBounds?.northeast
    else {
        return CLLocationCoordinate2D()
    }

    return CLLocationCoordinate2D(latitude: southwest.lat,
                                  longitude: northeast.lng)
}
```



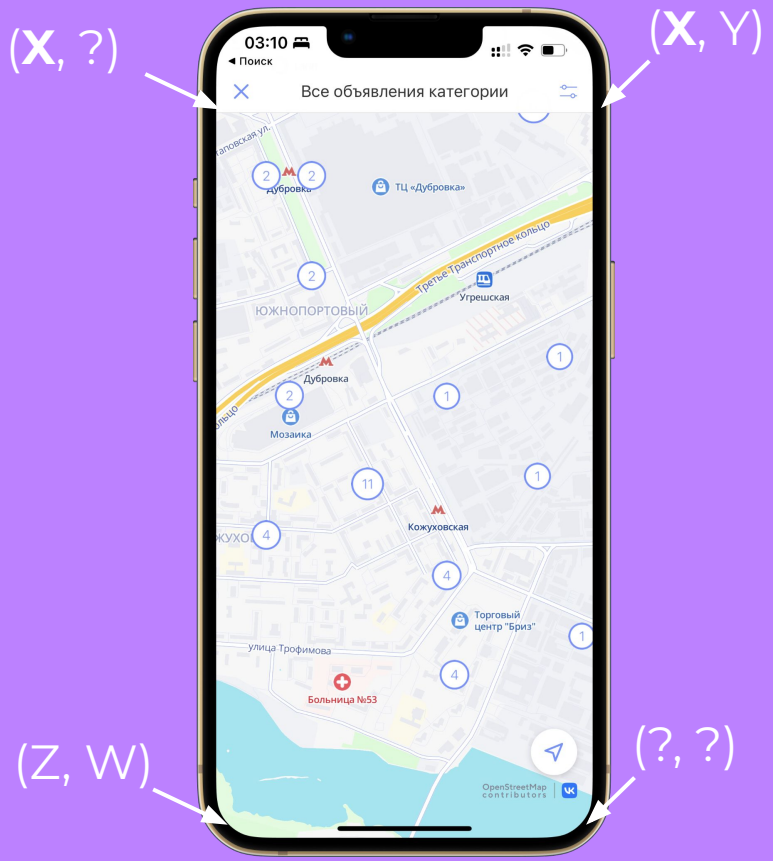
юла

Новые координаты

Имеем: Северо-Восток и Юго-Запад

Хотим: Северо-Запад и Юго-Восток

```swift
public var topLeft: CLLocationCoordinate2D {
    guard
        let southwest = view.mapBounds?.southwest,
        let northeast = view.mapBounds?.northeast
    else {
        return CLLocationCoordinate2D()
    }

    return CLLocationCoordinate2D(latitude: northeast.lat,
                                  longitude: southwest.lng)
}

public var bottomRight: CLLocationCoordinate2D {
    guard
        let southwest = view.mapBounds?.southwest,
        let northeast = view.mapBounds?.northeast
    else {
        return CLLocationCoordinate2D()
    }

    return CLLocationCoordinate2D(latitude: southwest.lat,
                                  longitude: northeast.lng)
}
```



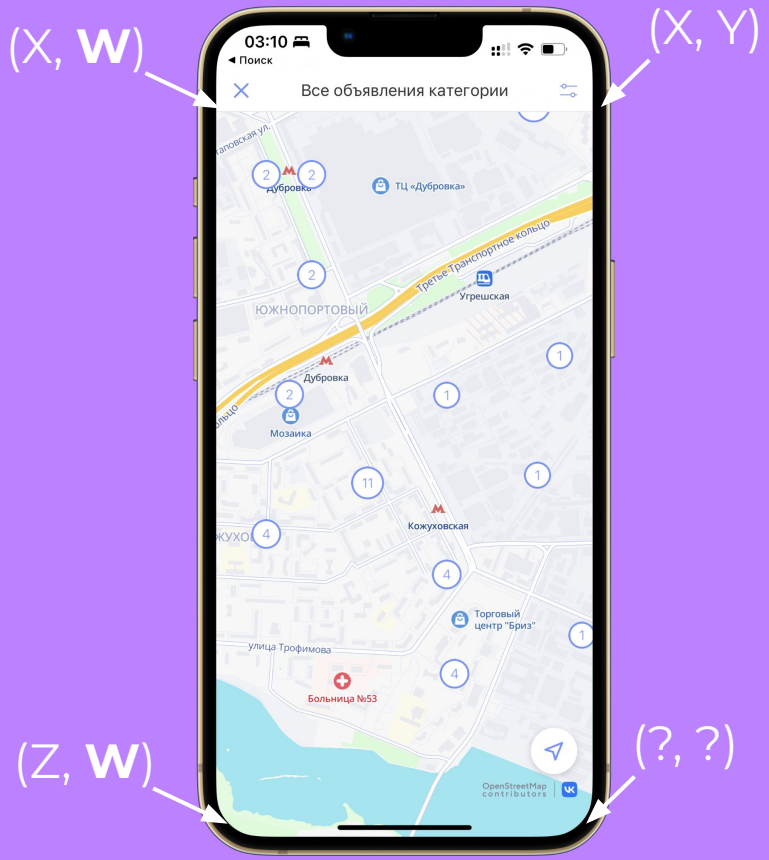
юла

Новые координаты

Имеем: Северо-Восток и Юго-Запад

Хотим: Северо-Запад и Юго-Восток

```swift
public var topLeft: CLLocationCoordinate2D {
    guard
        let southwest = view.mapBounds?.southwest,
        let northeast = view.mapBounds?.northeast
    else {
        return CLLocationCoordinate2D()
    }

    return CLLocationCoordinate2D(latitude: northeast.lat,
                                  longitude: southwest.lng)
}

public var bottomRight: CLLocationCoordinate2D {
    guard
        let southwest = view.mapBounds?.southwest,
        let northeast = view.mapBounds?.northeast
    else {
        return CLLocationCoordinate2D()
    }

    return CLLocationCoordinate2D(latitude: southwest.lat,
                                  longitude: northeast.lng)
}
```



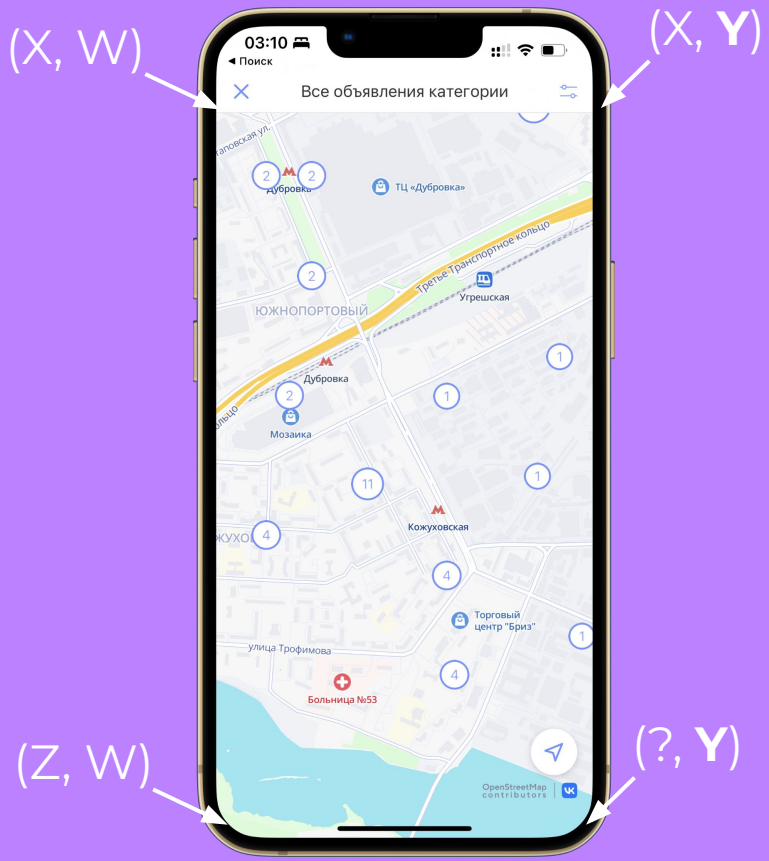
юла

Новые координаты

Имеем: Северо-Восток и Юго-Запад

Хотим: Северо-Запад и Юго-Восток

```
public var topLeft: CLLocationCoordinate2D {
    guard
        let southwest = view.mapBounds?.southwest,
        let northeast = view.mapBounds?.northeast
    else {
        return CLLocationCoordinate2D()
    }

    return CLLocationCoordinate2D(latitude: northeast.lat,
                                  longitude: southwest.lng)
}

public var bottomRight: CLLocationCoordinate2D {
    guard
        let southwest = view.mapBounds?.southwest,
        let northeast = view.mapBounds?.northeast
    else {
        return CLLocationCoordinate2D()
    }

    return CLLocationCoordinate2D(latitude: southwest.lat,
                                  longitude: northeast.lng)
}
```



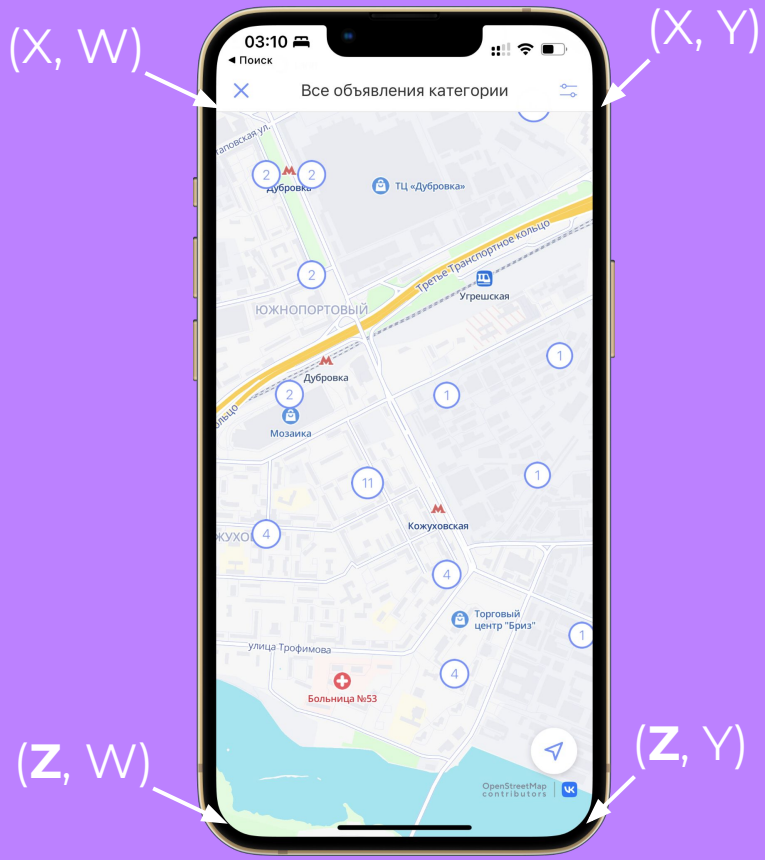
юла

Новые координаты

Имеем: Северо-Восток и Юго-Запад

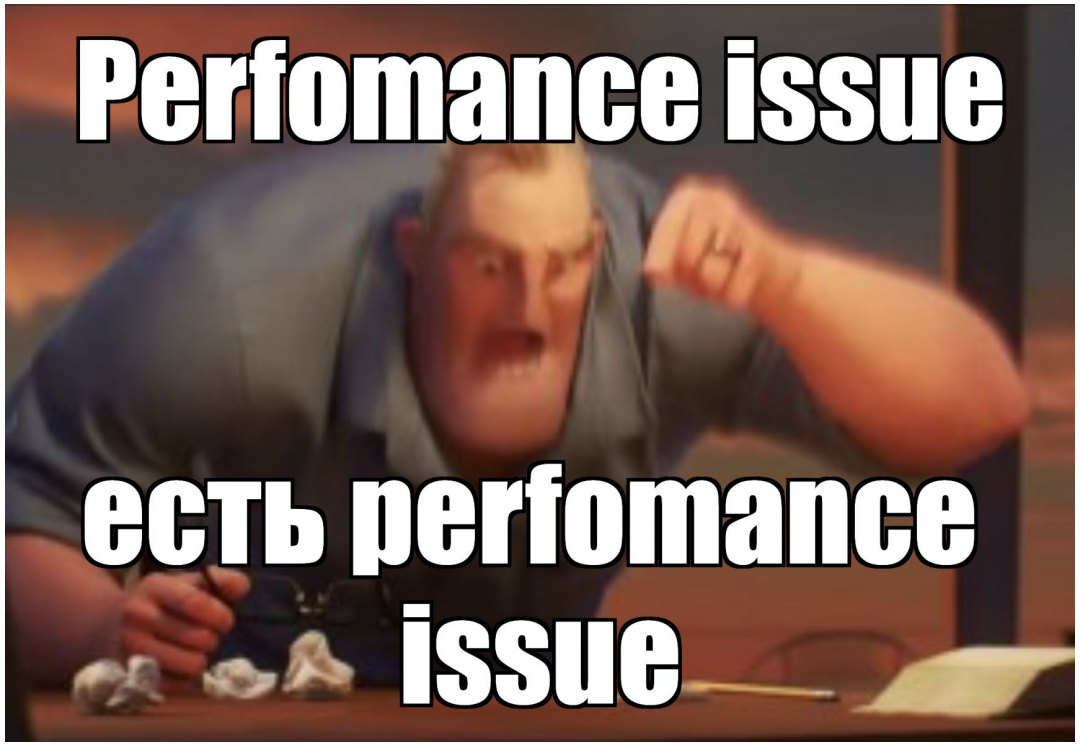
Хотим: Северо-Запад и Юго-Восток

```swift
public var topLeft: CLLocationCoordinate2D {
    guard
        let southwest = view.mapBounds?.southwest,
        let northeast = view.mapBounds?.northeast
    else {
        return CLLocationCoordinate2D()
    }

    return CLLocationCoordinate2D(latitude: northeast.lat,
                                  longitude: southwest.lng)
}

public var bottomRight: CLLocationCoordinate2D {
    guard
        let southwest = view.mapBounds?.southwest,
        let northeast = view.mapBounds?.northeast
    else {
        return CLLocationCoordinate2D()
    }

    return CLLocationCoordinate2D(latitude: southwest.lat,
                                  longitude: northeast.lng)
}
```



юла

Кластеризация. Диффер



Кластеризация. Диффер

```
struct VKMapsMarkerBox: Equatable {
    let marker: Marker

    static func == (lhs: VKMapsMarkerBox, rhs: VKMapsMarkerBox) -> Bool {
        let isEqualPin: Bool
        let isEqualCoords: Bool = lhs.marker.coords == rhs.marker.coords

        switch (lhs.marker.pin, rhs.marker.pin) {
        case (.custom(let lhsImage), .custom(let rhsImage)):
            isEqualPin = lhsImage.size == rhsImage.size && lhsImage.pngData() == rhsImage.pngData()
        default:
            isEqualPin = lhs.marker.pin == rhs.marker.pin
        }

        return isEqualPin && isEqualCoords
    }
}
```

Кластеризация. Диффер

```swift
final class VKMapsMarkerDiffer {

    struct Changeset {
        let removeList: [String]
        let insertList: [String]
    }

    static func diff(source: [Marker], target: [Marker]) -> Changeset {
        let sourceBoxed = source.map({ VKMapsMarkerBox(marker: $0) })
        let targetBoxed = target.map({ VKMapsMarkerBox(marker: $0) })

        let insertList: [String] = targetBoxed.filter({ !sourceBoxed.contains($0) }).map({ $0.marker.id })
        let removeList: [String] = sourceBoxed.filter({ !targetBoxed.contains($0) }).map({ $0.marker.id })

        return Changeset(removeList: removeList, insertList: insertList)
    }
}
```

Кластеризация. Диффер

```
public func addMarkers(with input: [YoulaMapMarkerInput], shouldClear: Bool) {
    if !shouldClear {
        input.forEach { addMarker(with: $0) }
        return
    }

    let currentMarkers = disposedBag.getMarkers()
    let newMarkers = input.map({ createMarker(with: $0) })

    let changeset = VKMapsMarkerDiffer.diff(source: currentMarkers, target: newMarkers)

    disposedBag.remove(markerIds: changeset.removeList)

    let insertMarkers = newMarkers.filter({ changeset.insertList.contains($0.id) })

    insertMarkers.forEach {
        disposedBag.append(marker: $0)
        view.addMarker($0)
    }
}
```

Кластеризация. Диффер

```swift
public func addMarkers(with input: [YoulaMapMarkerInput], shouldClear: Bool) {
    if shouldClear {
        clear()
    }

    input.forEach { self.addMarker(with: $0) }
}
```

Внедрение к себе в проект

Для разработчиков, публикующихся в RuStore - **бесплатно.** И на iOS тоже.

Спасибо за внимание

Telegram

VK

юла