Обзор Xamarin Forms

Илья Ефимов.

efimovilya86

skill.net.rude 18+

@ilyaefimov86

Занимаюсь программированием 15 лет.
Работал в больших и маленьких компаниях.
Сделал игру.
Работал в стартапе.
Делаю платформу для обучения.

О чем рассказ

- Что такое Xamarin Forms.
- Когда нужно использовать.
- Компиляция и рантайм.
- MVVM, code-behind, ReactiveUI, IoC DI, DependencyService
- Навигация, создание экземпляров страниц и передача в них параметров.
- CI, аналитика и баг-репорт - все это есть в AppCenter.
- Резюме.

История создания

1. .Net

История создания

1. .Net
2. Ximian Mono

История создания

1. .Net
2. Ximian Mono
3. Novell Xamarin.

История создания

1. .Net
2. Ximian Mono
3. Novell Xamarin.
4. Microsoft Xamarin.

История создания

1. .Net
2. Ximian Mono
3. Novell Xamarin.
4. Microsoft Xamarin.
5. Xamarin.Forms

Когда нужно использовать

- Есть команда C#.
- Особенно которая умеет WPF.
- Много готового кода.
- Когда важна скорость разработки.

Компиляция.

- Компиляция в нативные приложения осуществляет mono.

Компиляция.

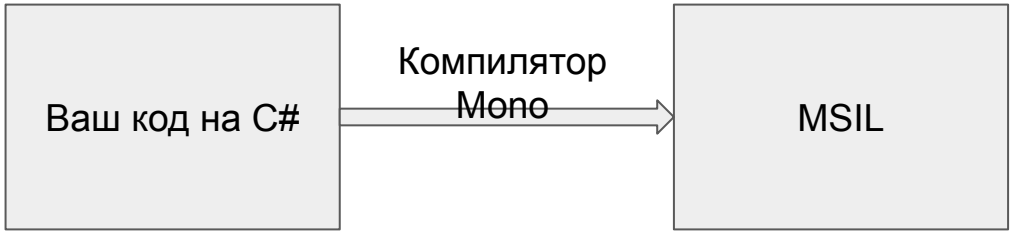
- Компиляцию в нативные приложения осуществляет mono.
- iOS - Ahead-Of-Time.

Компиляция.

- Компиляцию в нативные приложения осуществляет mono.
- iOS - Ahead-Of-Time.
- Android - JIT

Компиляция в iOS

Компиляция в iOS



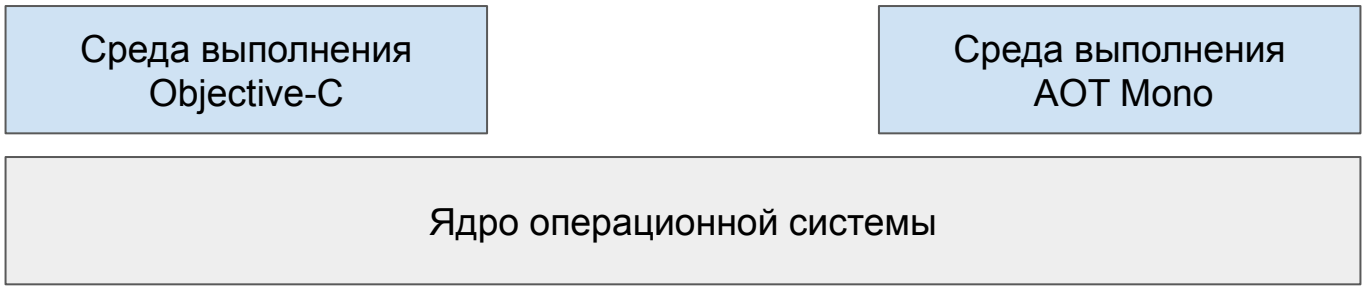
Компиляция в iOS

Исполнение в iOS

Среда выполнения
Objective-C

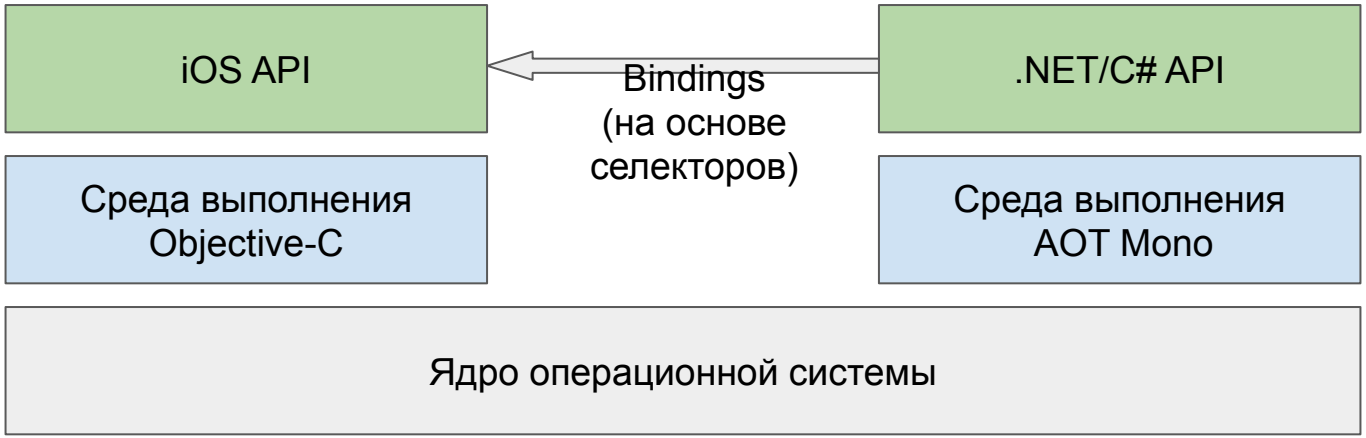
Среда выполнения
AOT Mono

Исполнение в iOS



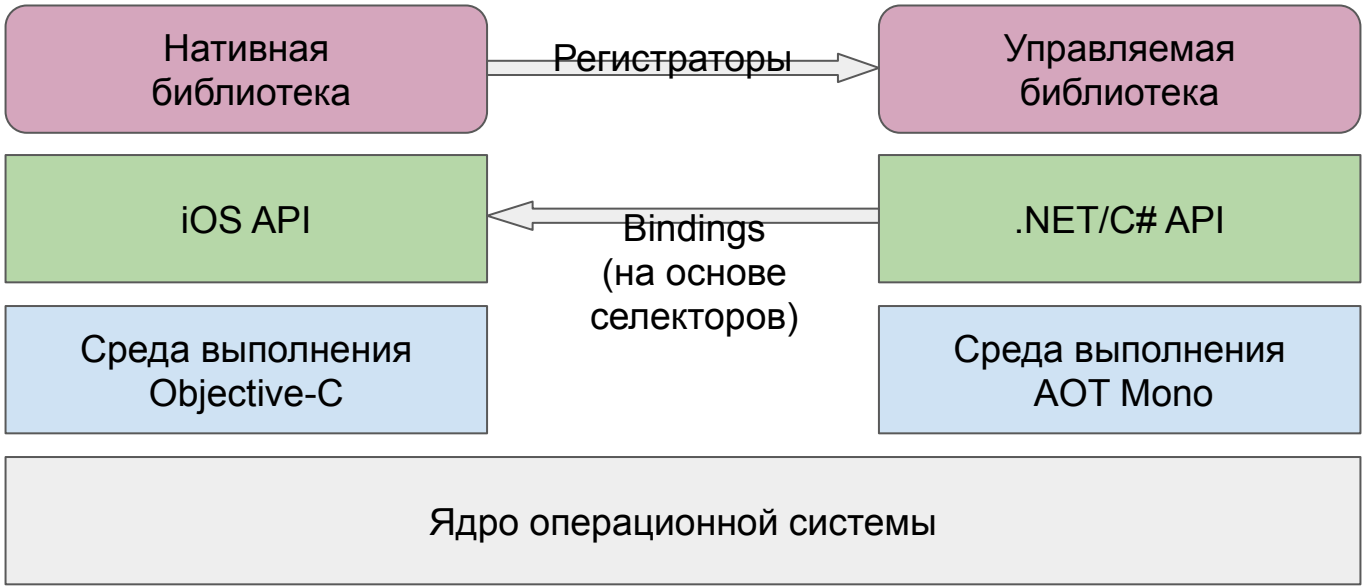
Исполнение в iOS

Исполнение в iOS



```
namespace Example.Binding {
    [Register("NSEnumerator")]
    class NSEnumerator : NSObject
    {
        static Selector selInit       = new Selector("init");
        static Selector selAllObjects = new Selector("allObjects");

        [Export("init")]
        public NSEnumerator() : base(NSObjectFlag.Empty)
        {
            Handle = Messaging.IntPtr_objc_msgSend(this.Handle, selInit.Handle);
        }
        public NSEnumerator(IntPtr handle) : base(handle) {}

        [Export("nextObject")]
        public virtual NSObject NextObject()
        {
            return Runtime.GetNSObject(
                Messaging.IntPtr_objc_msgSend(this.Handle,
                    selNextObject.Handle));
        }

        public virtual NSArray AllObjects {
            [Export("allObjects")]
            get {
                return (NSArray) Runtime.GetNSObject(
                    Messaging.IntPtr_objc_msgSend(this.Handle,
                        selAllObjects.Handle)); }
        }
    }
}
```

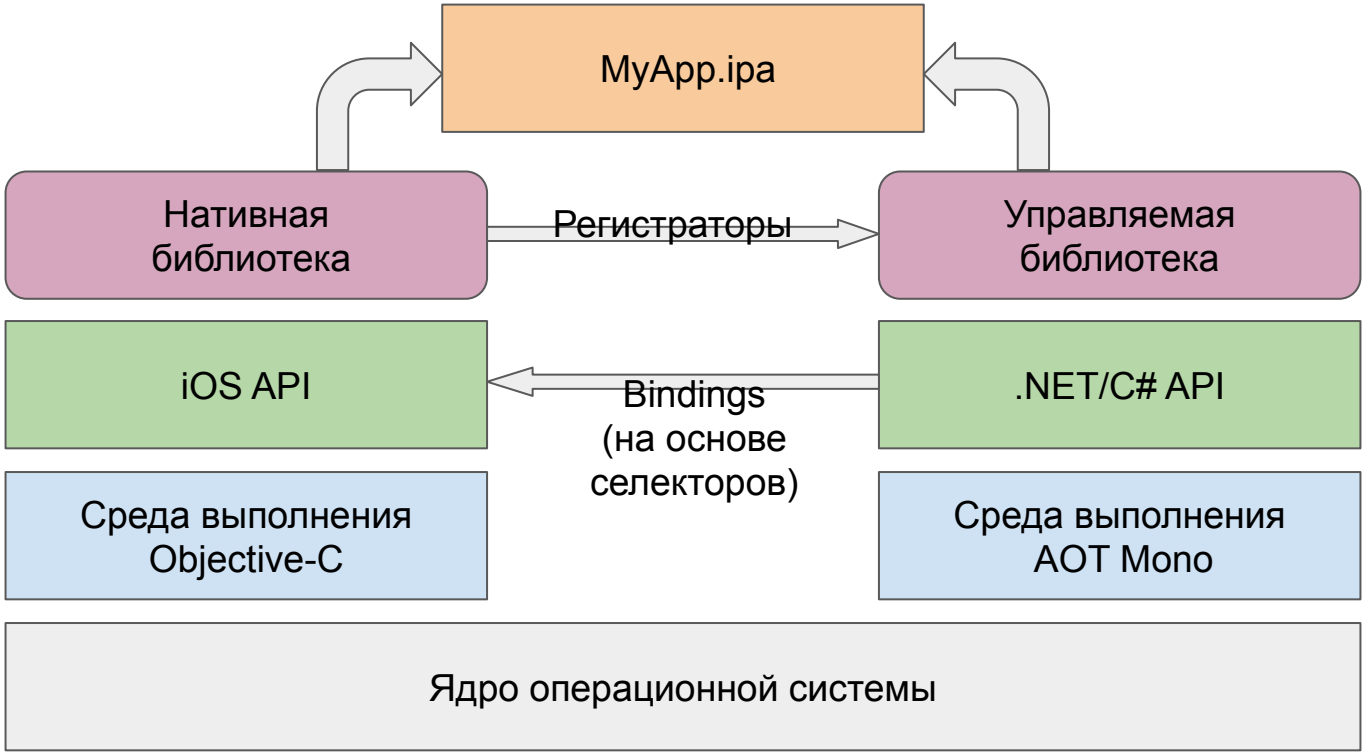
Исполнение в iOS



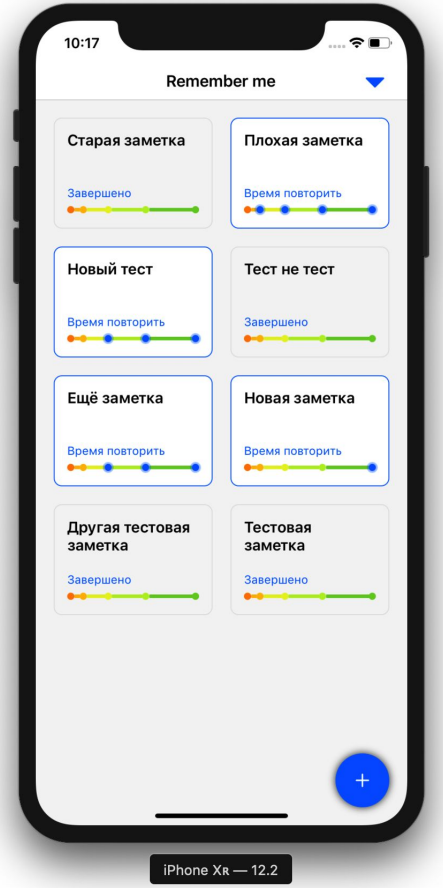
```
class MyViewController : UIViewController{
     [Export ("myFunc")]
     public void MyFunc ()
     {
     }
 }
```

```
@interface MyViewController : UIViewController { }

    -(void)myFunc;
@end

@implementation MyViewController {}

    -(void) myFunc
    {
        // код вызывающий управляемый метод MyViewController.MyFunc
    }
@end
```

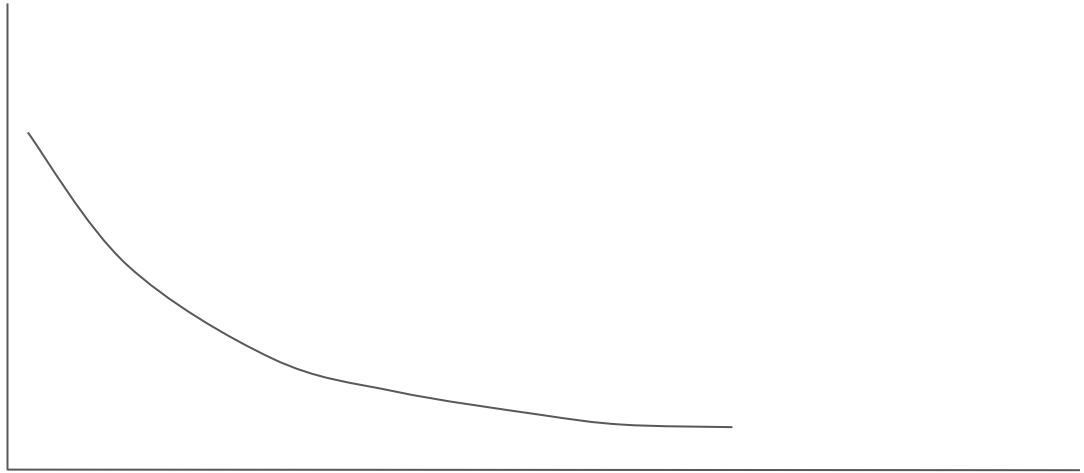
Исполнение в iOS



Сложное позади. Теперь смотрим код.



- Remember me - приложение для запоминания.
- Основано на кривой запоминания Эббингауза.
- Напоминает о том, что нужно повторить информацию, через определенные промежутки времени.



Архитектура приложения

- Проект с общей библиотекой .NetStandard.
- Xaml.
- MVVM.
- IoC DI, DependencyService.
- Навигация.
- ReactiveUI.
- TabbedPage.

App Center

- Сборка.
- Отправка билда на устройство.
- Тестирование.
- Диагностика неполадок.
- Аналитика.

Резюме

- Отличный инструмент для специалистов .Net/C#.
- Хорошая производительность.
- Стандартными средствами можно реализовать практически все. Для остального есть биндинги.
- При принятии решения оценивайте риски бизнеса.