

Визуальное программир ование

ЛЕКЦИЯ 11

Содержание лекции

01

Xamarin

02

Архитектура Xamarin
приложений

03

Библиотеки Xamarin

04

XAML

XAMARIN

Xamarin

Xamarin — это платформа с открытым исходным кодом, предназначенная для разработки мобильных приложений.

С помощью Xamarin разработчик может написать всю бизнес-логику на одном языке, но при этом получить характеристики производительности, оформление и поведение, характерные для каждой отдельной платформы.

Приложения Xamarin можно писать на ПК или Mac и компилировать в собственные пакеты приложений, например в файлы с расширением .apk для Android или .ipa для iOS.

Полная привязка для базовых пакетов SDK

Xamarin содержит привязки практически для всех базовых пакетов SDK в iOS и Android.

Эти привязки являются строго типизированными, что означает, что они удобны в навигации и использовании, а также позволяют осуществлять качественную проверку типов во время компиляции и разработки.

Строго типизированные привязки позволяют сократить количество ошибок времени выполнения и повысить качество приложений.

Взаимодействие Objective-C, Java, C и C++

Хамарин позволяет напрямую вызывать библиотеки Objective-C, Java, C и C++ для более эффективного использования стороннего кода.

Это позволяет использовать существующие библиотеки iOS и Android, написанные на Objective-C, Java или C/C++. Кроме того, Хамарин позволяет использовать собственные библиотеки Objective-C и Java с помощью декларативного синтаксиса.

Современные конструкции языка

Приложения Xamarin написаны на современном языке C#, который характеризуется значительными улучшениями по сравнению с Objective-C и Java.

Сюда входят динамические функции языка, функциональные конструкции, например лямбда-выражения, LINQ, функции параллельного программирования, универсальные шаблоны и т. д.

Надежная библиотека базовых классов

Приложения Xamarin используют библиотеку BCL .NET, большую коллекцию классов со всеобъемлющими и упрощенными возможностями, включая поддержку XML, баз данных, сериализации, операций ввода-вывода, строк, сетевых функций и т. д.

Существующий код C# можно скомпилировать для использования в приложениях, обеспечивая доступ к тысячам библиотек, которые содержат дополнительные функции, выходящие за рамки BCL.

Кроссплатформенность

Xamarin предлагает усовершенствованную кроссплатформенную поддержку для трех основных платформ — iOS, Android и Windows.

Объем общего кода в созданных приложениях может достигать 90%, а библиотека Xamarin.Essentials предлагает универсальный API-интерфейс для доступа к общим ресурсам на всех трех платформах.

Это позволяет значительно сократить затраты на разработку и время выпуска продуктов на рынок для разработчиков, создающих мобильные приложения.

АРХИТЕКТУРА ХАМАРИН ПРИЛОЖЕНИЙ



Android Native UI



iOS Native UI

Your Shared Application Code (C#/XAML)

.NET
APIs

Android
Bindings

MCW

Android.*

Java.*

Mono

ACW

Android Runtime
(ART)

Linux Kernel

.NET APIs

Bindings

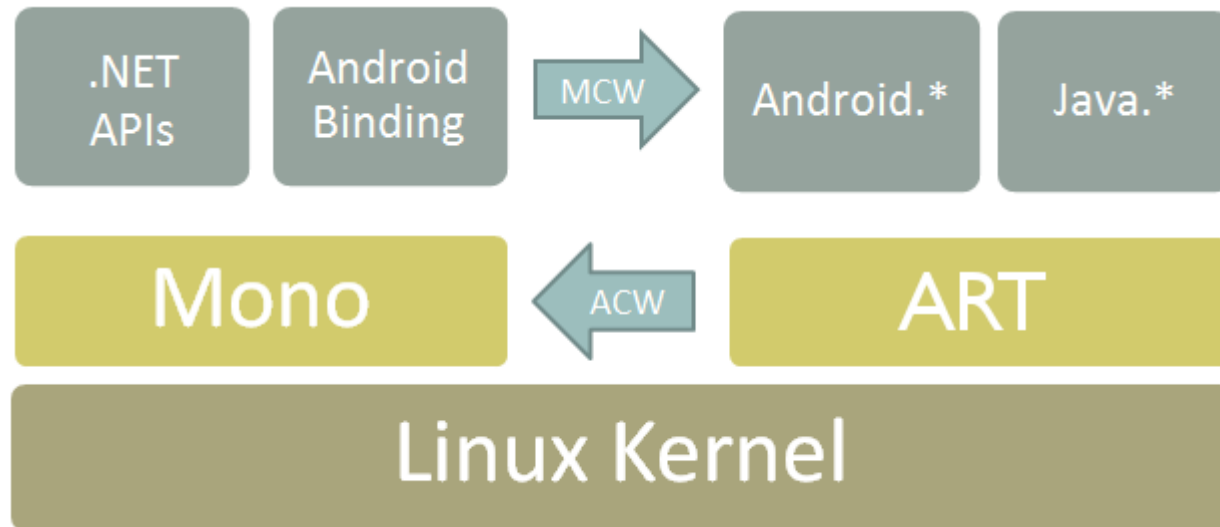
iOS APIs

Mono (Full AOT)

Objective-C

Unix-Like Kernel

Xamarin.Android

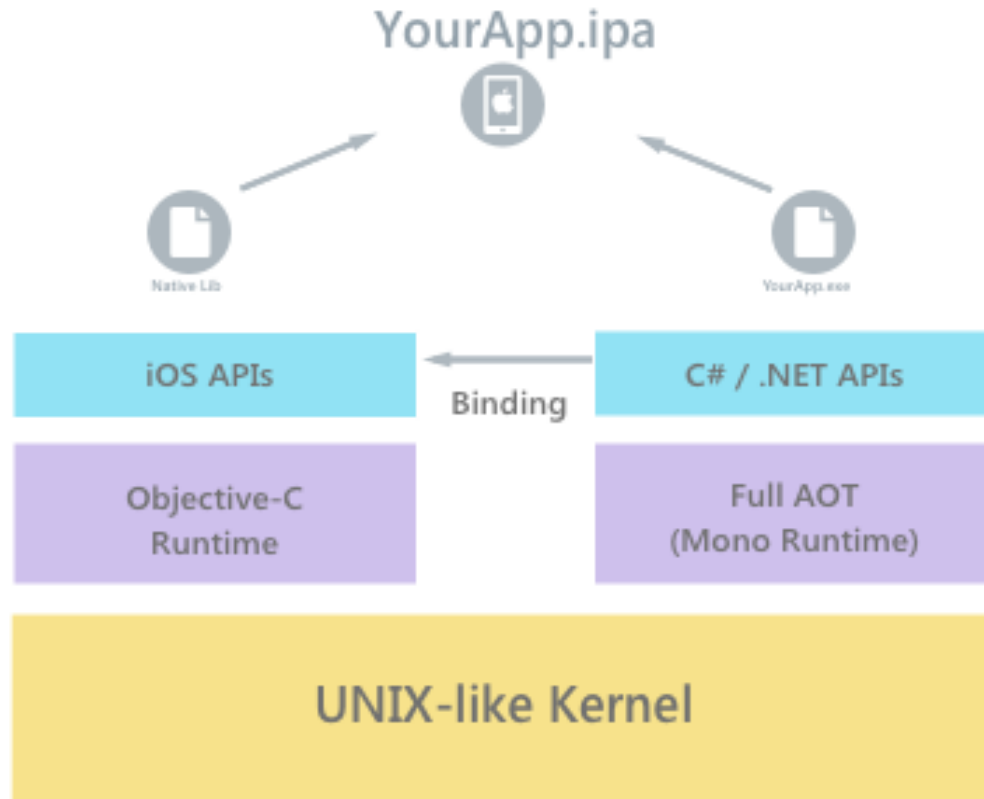


Приложения Xamarin.Android компилируются из языка C# в IL, который при запуске приложения претерпевает Just-in-Time-компиляцию в машинную сборку.

Приложения Xamarin.Android работают в среде выполнения Mono параллельно с виртуальной машиной среды выполнения Android (Android Runtime, ART). Xamarin предоставляет привязки (bindings) .NET к пространствам имен Android.* и Java.*.

Среда выполнения Mono обращается к этим пространствам имен с использованием управляемых вызываемых оболочек (Managed Callable Wrapper, MCW) и предоставляет среде выполнения ART вызываемые программы-оболочки Android (Android Callable Wrappers, ACW), благодаря чему обе среды могут вызывать код друг друга.

Xamarin.iOS



Приложения Xamarin.iOS проходят полную Ahead-of-Time-компиляцию (AOT) из языка C# в собственный код сборки ARM.

Xamarin использует селекторы (selectors) для предоставления кода Objective-C управляемому коду C# регистраторы (registrars) для предоставления управляемого кода C# коду Objective-C. Селекторы и регистраторы в совокупности называются привязками (bindings) и обеспечивают взаимодействие между Objective-C и C#.

БИБЛИОТЕКИ XAMARIN

Xamarin.Essentials

Это библиотека, предоставляющая кроссплатформенные API для собственных функций устройства.

Возможности:

- сведения об устройстве
- файловая система
- акселерометр
- телефон
- преобразование текста в речь
- блокировка экрана

Xamarin.Forms

Это платформа пользовательского интерфейса с открытым кодом.

Xamarin.Forms позволяет разработчикам создавать пользовательские интерфейсы в XAML с помощью кода программной части в C#.

Возможности:

- язык пользовательского интерфейса XAML
- привязка данных
- жесты
- задание стиля

XAML

XAML

Язык разметки приложений eXtensible (XAML) — это язык на основе XML, созданный корпорацией Майкрософт в качестве альтернативы программному коду для создания и инициализации объектов, а также организации этих объектов в иерархиях родительского дочернего элемента.

В XAML-файле Xamarin.Forms разработчик может определить пользовательские интерфейсы с помощью всех Xamarin.Forms представлений, макетов и страниц, а также пользовательских классов. XAML-файл можно скомпилировать или внедрить в исполняемый файл.

Преимущества

1. XAML часто является более кратким и удобочитаемым, чем эквивалентный код.
2. Иерархия «родители-потомки», присущая XML, позволяет XAML имитировать с большей визуальной ясностью иерархию объектов пользовательского интерфейса.
3. XAML может быть легко написан программистами, но также может быть инструментируемым и созданным средствами визуального проектирования.

Недостатки

1. XAML не может содержать код. Все обработчики событий должны быть определены в файле кода.
2. XAML не может содержать циклы для повторяющейся обработки.
3. XAML не может содержать условную обработку.
4. XAML обычно не может создавать экземпляры классов, которые не определяют конструктор без параметров.
5. XAML обычно не может вызывать методы.

ВОПРОСЫ ПО ЛЕКЦИИ