



6/7 37.6. Итоги и практика

текущий урок

Друзьям не жалко: бесплатные тренажёры, пробный урок и [скидка на занятия](#) в онлайн-школе английского для своих



Xamarin — уникальная технология, позволяющая вам охватить сразу три платформы при разработке.

Несмотря на то, что основная часть кода в вашем проекте будет переносимой, вы должны быть готовы к тому, что в процессе разработки у вас могут возникнуть нетривиальные ситуации, требующие изучения проблемы, а возможно и конкретных доработок под ту или иную платформу.

Если вы столкнетесь с проблемами при запуске/сборке учебного проекта, рассматривайте это как реальный рабочий кейс и возможность улучшить свои навыки самостоятельного решения технических проблем и поиска необходимой информации в Интернете.

Все эти навыки необходимы программисту, так как заранее знать и предусмотреть все возможные проблемы невозможно.

При самостоятельном изучении документации и поиске информации в Интернете настоятельно рекомендуем вам пользоваться английским языком. И вы всегда можете рассчитывать на нашу помощь как с вопросами по материалу, так и с настройкой окружения для разработки.

Что же мы узнали нового в этом модуле:

1

Основа Xamarin — фреймворк Mono

Мы немного погрузились в историю технологии и выяснили, как появился Xamarin на основе фреймворка Mono, узнали, как эта технология появилась у Microsoft, что в итоге привело к тому, что Xamarin SDK стал поставляться вместе с IDE Visual Studio.

2

Под что пишут на Xamarin и как он работает

Мы узнали, что Xamarin предоставляет нам возможность писать сразу под iOS, Android и Windows (UWP), при этом основная логика на C# составляет общий для всех платформ код. Мы также детально рассмотрели, что именно позволяет нам запускать C#-код сразу на всех платформах.

3

Из чего состоит Xamarin-проект и как его запускать

Мы познакомились со структурой мультиплатформенного Xamarin-проекта, поняли, где располагается общая логика а где — платформо-зависимый код. Мы также выяснили, какие бывают способы запуска проекта на Xamarin, как это сделать с помощью эмулятора, если у вас нет под рукой соответствующих мобильных устройств.

4

Как строится интерфейс Xamarin-приложения

Мы узнали два способа построения интерфейса — C# и XAML, а также способ связи страницы XAML со своим C#-обработчиком. Изучили то, как в Xamarin атрибуты XAML-разметки связаны с полями класса в C#. Также мы изучили основные элементы интерфейса, такие как Label, Button и другие.

5

Как самостоятельно создавать интерфейс из XML

Мы попробовали самостоятельно генерировать интерфейс «на лету» из XML-кода. Написали свое расширение XAML.

6

Что определяет вид интерфейса и расположение элементов на экране

Мы выяснили, что конечный вид и расположение элементов на экране зависит от того, внутри какого контейнера они скомпонованы, и изучили основные контейнеры: StackLayout, ScrollView, AbsoluteLayout, RelativeLayout и Grid.

В следующем модуле мы продолжим рассматривать некоторые элементы интерфейса Xamarin, которые не успели охватить здесь, а также будем рассматривать другие важные темы, без которых ваше приложение не будет работать должным образом: навигация, локальная БД, взаимодействие с веб-сервисом на Backend и многое другое.

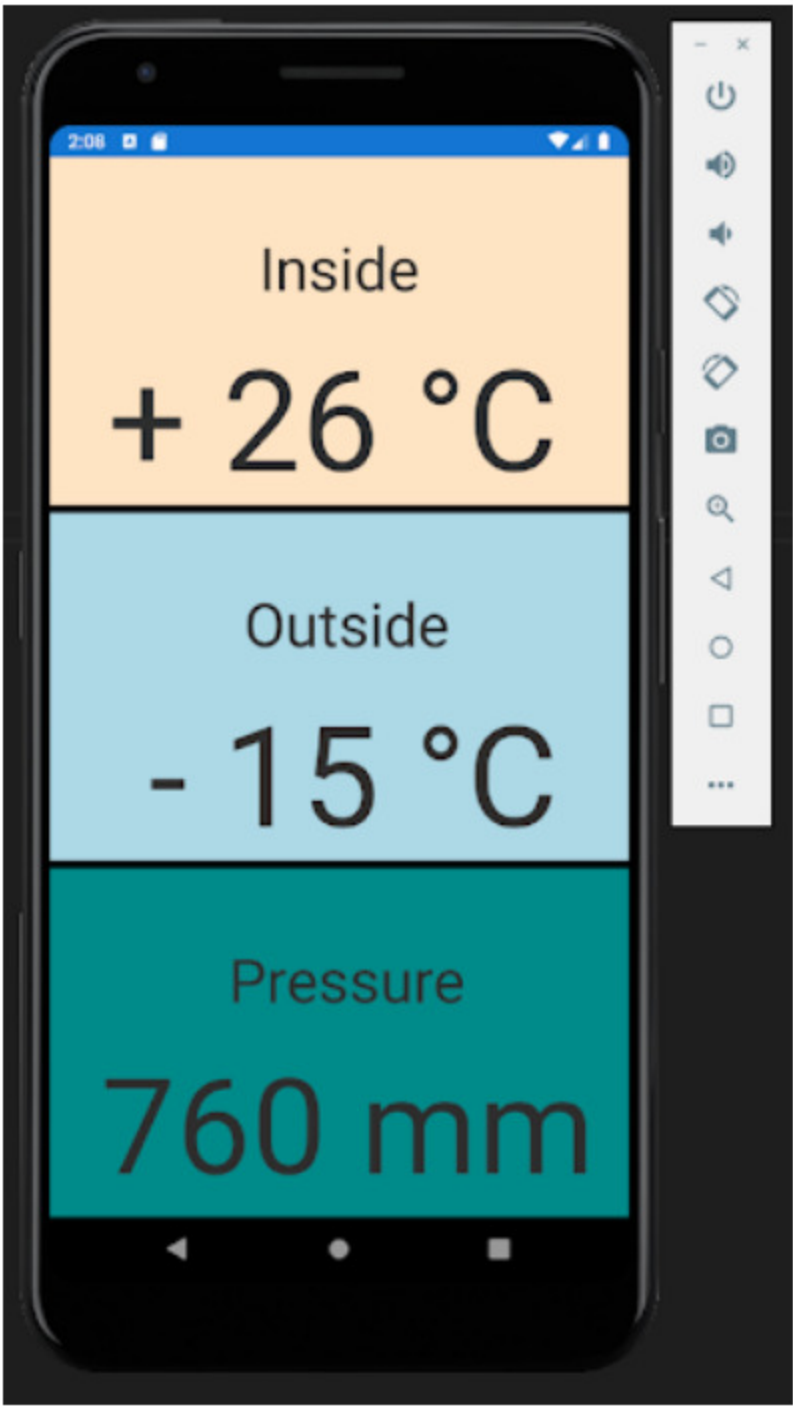
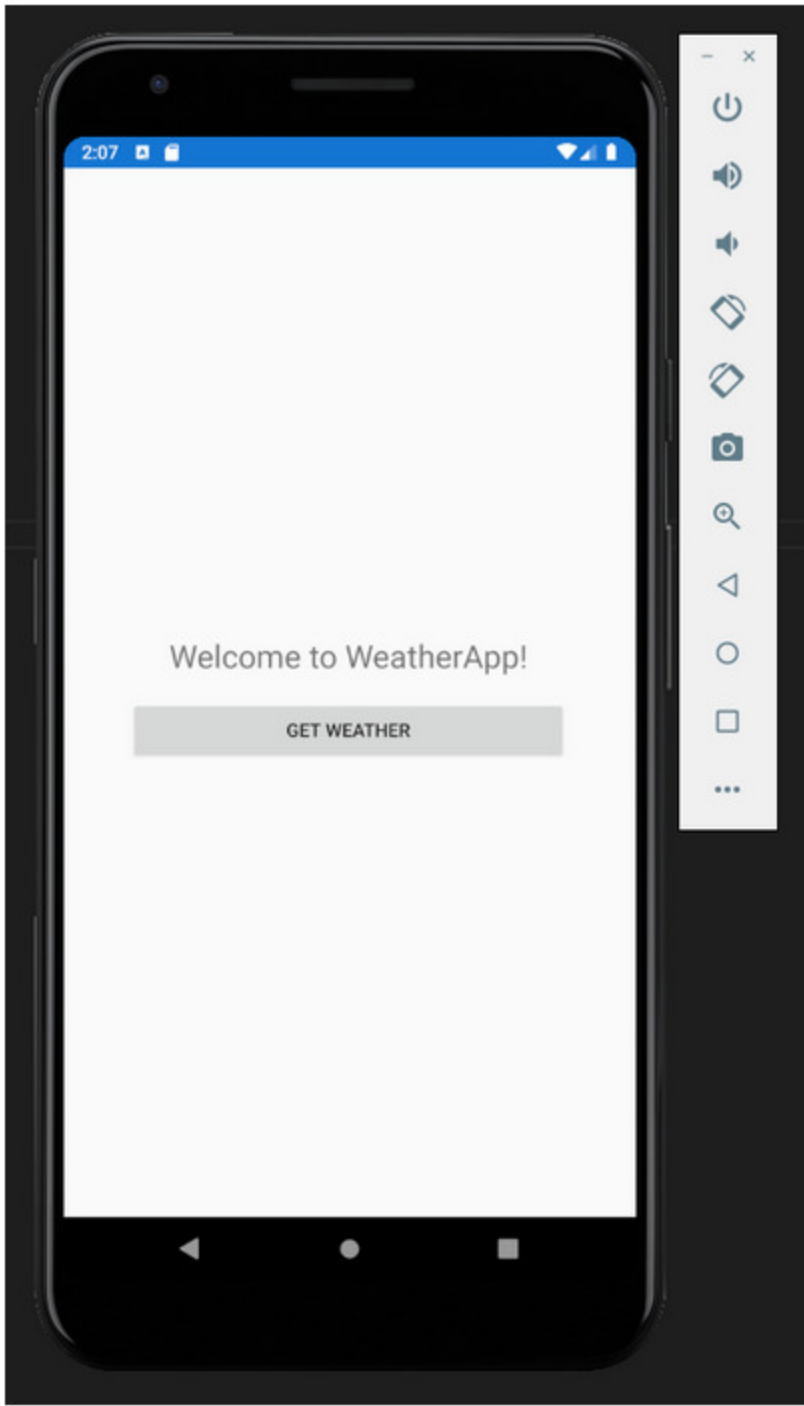
А сейчас переходим к практической части!

Практика

Ваша задача: создать простое Xamarin-приложение, показывающее погоду. В приложении будет один экран, но его содержимое будет меняться.

При запуске на экране будет название и кнопка «Узнать погоду». По нажатию на кнопку приложение показывает три блока с температурой и атмосферным давлением.

Пример того, что у вас может получиться:



Пожалуйста, не старайтесь копировать интерфейс, цвета, шрифты и прочее в точности. Для нас главное понять, что вы в принципе знаете, как работать с разметкой и блоками в Xamarin и можете построить несложный интерфейс. Тип используемого контейнера компоновки, писать код C# или XAML, — всё это на ваше усмотрение.

Пример решения от автора можно будет увидеть в следующем модуле.

Удачи!

< Предыдущий урок

Следующий урок >