

## **ПРАКТИЧНА РОБОТА № 8. СТВОРЕННЯ ТА ЗАПУСК КРОСМОБІЛЬНОГО ДОДАТКУ ЗАСОБАМИ XAMARIN.FORMS**

**Мета:** набуття здобувачами освіти навичок створення та запуску кросмобільного додатку засобами Xamarin.Forms у середовищі Visual Studio 2019; створення ключа для підписування додатку; створення інсталяційного файлу мобільного додатку.

### **Короткі теоретичні відомості**

Xamarin.Forms представляє платформу, яка націлена на створення кросплатформових додатків під Android, iOS та Windows 10. Навіщо використовувати саме цю платформу, які переваги вона несе? Є певні статистичні дані, що значна частина мобільних програм створюється більш ніж для однієї платформи, наприклад, для Android та iOS. Однак неминуче розробники стикаються з такими труднощами:

- відмінність у підходах побудова графічного інтерфейсу однак впливає розробку. Розробники змушені підлаштовувати програму під вимоги до інтерфейсу на конкретній платформі;
- різні API - відмінність у програмних інтерфейсах та реалізаціях тих чи інших різного функціоналу також вимагає від програміста облік цих специфічних особливостей;
- різні платформи для розробки. Наприклад, щоб створювати програми для iOS необхідне відповідне середовище – macOS і ряд спеціальних інструментів типу XCode, як мову програмування вибирається Objective-C або Swift; для Android використовуються середовища - Android Studio, Eclipse, як мова програмування застосовується Java або Kotlin; для створення програм під Windows використовується Visual Studio та мови - C#, F#, VB.NET, C++.

Такий діапазон платформ, засобів розробки та мов програмування не може позитивно позначатися на термінах створення додатків, і, зрештою, на коштах, що виділяються на розробку. Було б дуже ефективно мати один інструмент, який дозволяв легко і просто створювати програми відразу для всіх платформ. Таким інструментом є платформа Xamarin (вимовляється як "земарин").

Xamarin дозволяє створювати одну єдину логіку програми із застосуванням C# і .NET відразу для всіх трьох платформ – Android, iOS, UWP.

Переваги використання Xamarin.Forms:

- у процесі розробки створюється єдиний код для всіх платформ;
- Xamarin надає прямий доступ до нативних API кожної платформи;
- Xamarin Forms підтримує декілька платформ: Android, iOS, UWP, Tizen.

Найбільш важливою особливістю Xamarin є можливість створювати кросплатформові додатки – тобто одна логіка для всіх платформ. Дана можливість представлена технологією Xamarin.Forms і яка працює рівнем вище Xamarin.Android і Xamarin.iOS. Тобто за допомогою Xamarin.Forms ми один раз можемо визначити візуальний інтерфейс, один раз до нього прив'язати якусь логіку на C# і все це буде працювати на Android, iOS та Windows. Потім Xamarin.Forms за допомогою рендерерів (renderer) – спеціальних об'єктів для зв'язку контролів на XAML/C# з нативними контролами транслюють візуальні компоненти Xamarin.Forms у графічний інтерфейс, специфічний для кожної платформи.

## Хід роботи

### 1. Установка Xamarin

Для розробки кросплатформових додатків на Xamarin необхідно встановити середовище [Visual Studio 2019 Community](#). При встановленні Visual Studio 2019 у програмі для установника обов'язково треба вибрати пункт "Розробка мобільних програм на .NET" (рис. 1).

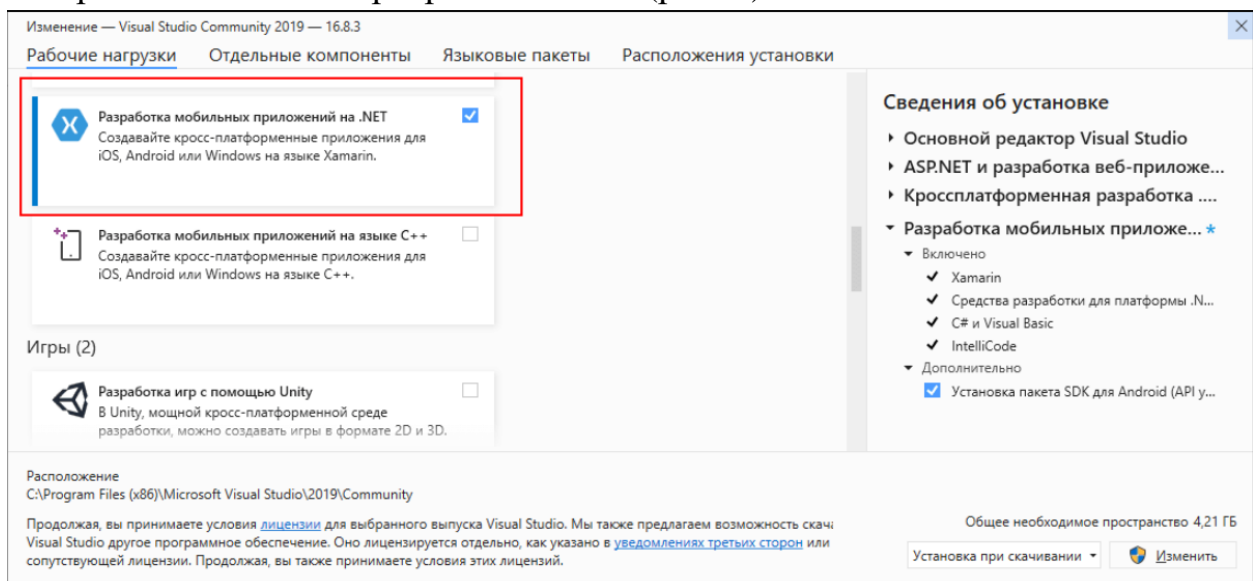


Рисунок 1 – Налаштування Visual Studio 2019 для встановлення Xamarin

Після інсталяції перевіряємо, що Xamarin встановлений. Для цього переходимо в меню Help -> About Microsoft Visual Studio (рис. 2). Якщо цільовою операційною системою є macOS X, то необхідно встановити [Visual Studio for Mac](#). Крім того, для розробки на macOS X потрібно встановити XCode, доступний в AppStore.

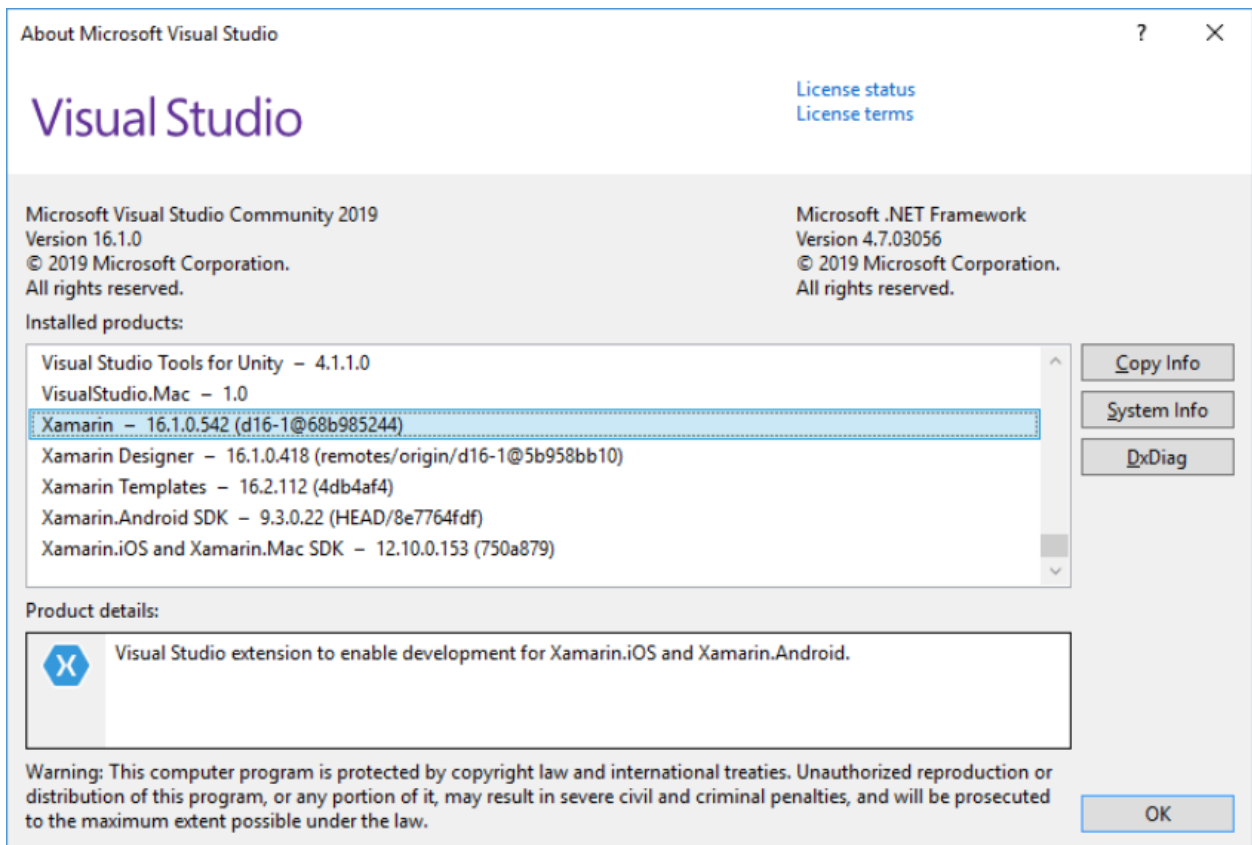


Рисунок 2 – Інформаційне вікно Microsoft Visual Studio про встановлені пакети

## 2. Створення та запуск проекту

Для створення кросплатформних програм для Xamarin Forms у Visual Studio 2019 призначений шаблон проекту, який називається Mobile App (Xamarin.Forms) (рис. 3).

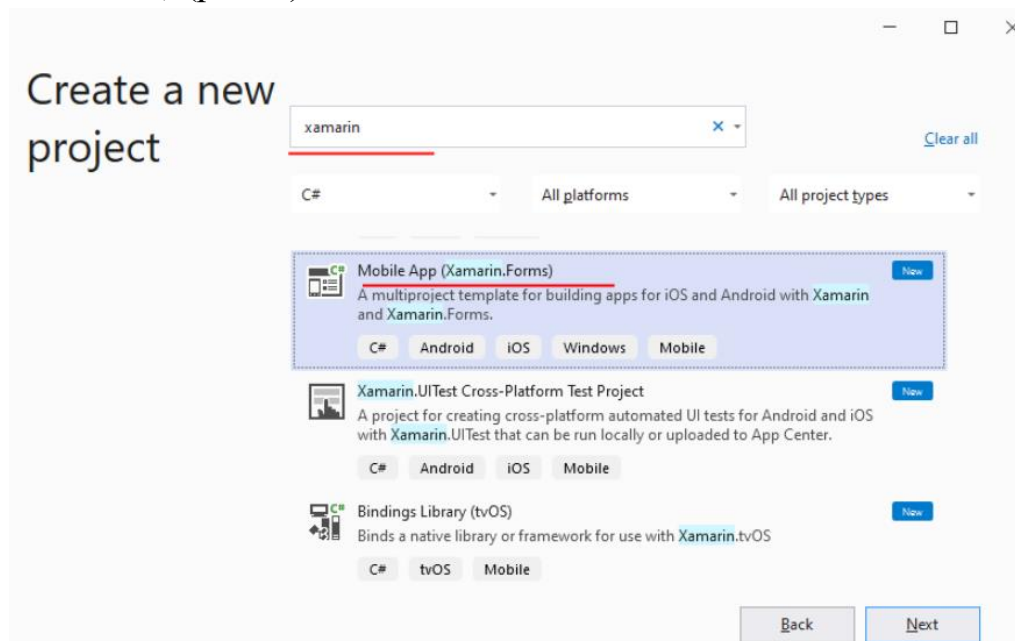


Рисунок 3 – Створення проекту типу Mobile App (Xamarin.Forms)

Створюємо проект FullApp. Шаблони Floyout – проект з бічним меню, Tabbed – проект програми, яка використовує вкладки для навігації між сторінками, Blank – порожній шаблон, що створює проект з мінімальною функціональністю. Вибираємо шаблон Blank (рис. 4).

При створенні проекту в опції Platform вибираємо під які ОС буде створюватись проект: Android, iOS, Windows (UWP). Windows (UWP) доступний під Windows, якщо для Visual Studio встановлено інструмент для розробки під UWP.

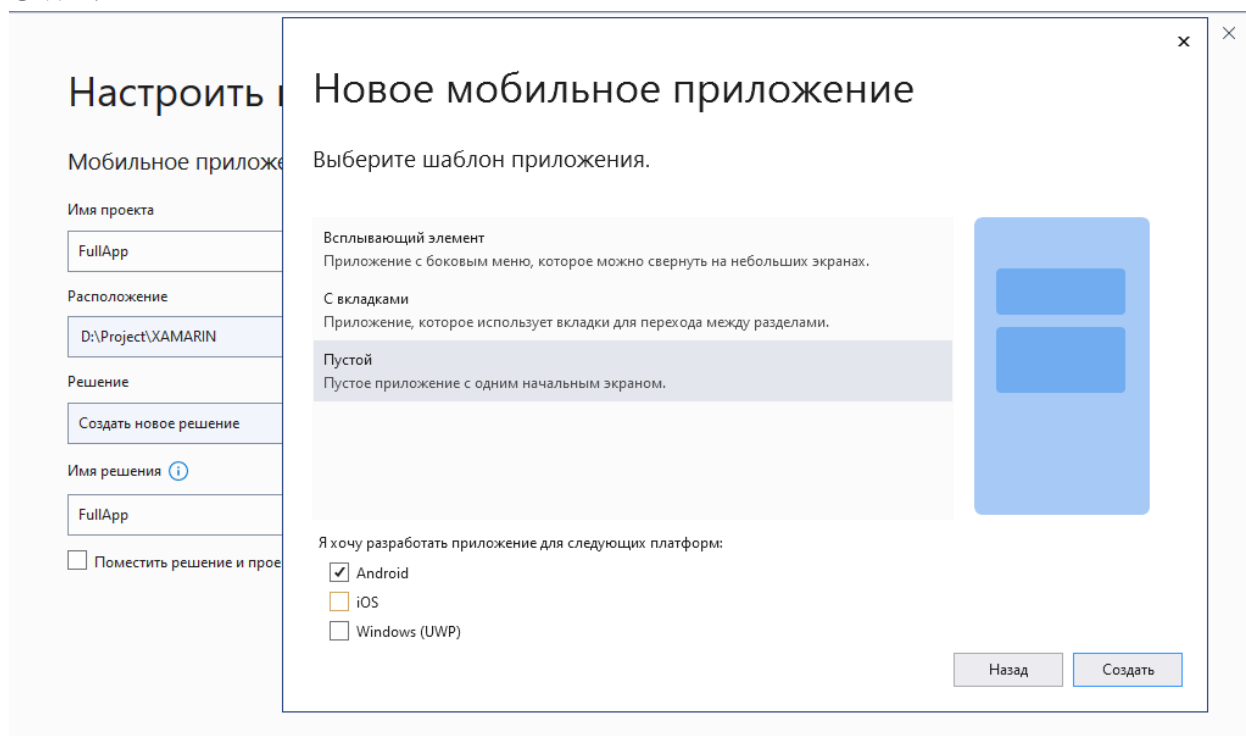


Рисунок 4 – Створення проекту з шаблоном Blank

Visual Studio згенерує нове рішення. Якщо вибрати всі три платформи, то створене рішення містить чотири проекти:

- **FullApp**: головний проект бібліотеки, яка містить усю основну логіку програми;
- **FullApp.Android**: проект для Android;
- **FullApp.iOS**: проект для iOS;
- **FullApp.UWP**: проект для Windows 10.

Однак, залежно від операційної системи, набір проектів може бути іншим. Головним є проект FullApp. Він містить весь код і визначення інтерфейсу, яку потім будуть використовувати всі інші проекти.

У головному проекті FullApp буде чотири основні файли:

- **App.xaml**: файл, який визначає ресурси, спільні для всієї програми;
- **App.xaml.cs**: файл із кодом C#, з якого починається виконання програми;

- **MainPage.xaml** : файл з візуальним інтерфейсом для єдиної сторінки MainPage у вигляді xaml;
- **MainPage.xaml.cs** : файл, який містить логіку MainPage мовою C#;
- **AssemblyInfo.cs** : файл із кодом мовою C#, який використовується для встановлення налаштувань програм

### 3. Запуск на Android

Якщо робоча машина, на якій ведеться розробка, підтримує віртуалізацію, можна для налагодження програм використовувати емулятори: емулятор Android від Microsoft або Android Player.

Якщо є пристрій з ОС Android, то можна його використовувати для тестування. Для цього потрібно підключити цей пристрій до комп'ютера за допомогою кабелю USB. А на мобільному пристрої встановити режим розробника в параметрах. Цілком можливо, що мобільний пристрій підключений до комп'ютера, належним чином налаштований, але Visual Studio його все одно не бачить. У цьому випадку можна перезавантажити сервер Adb. Для цього треба в Visual Studio перейти в меню до пункту Tools -> Android -> Restart Adb Server (рис. 5).

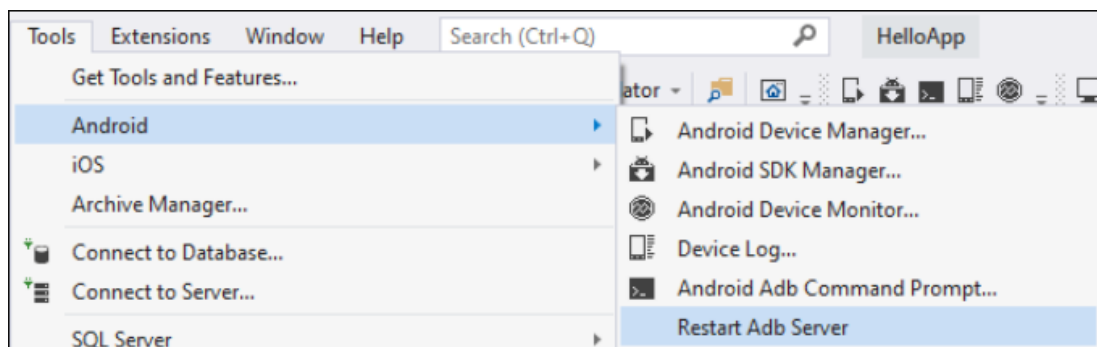


Рисунок 5 – Перезавантаження сервера Adb

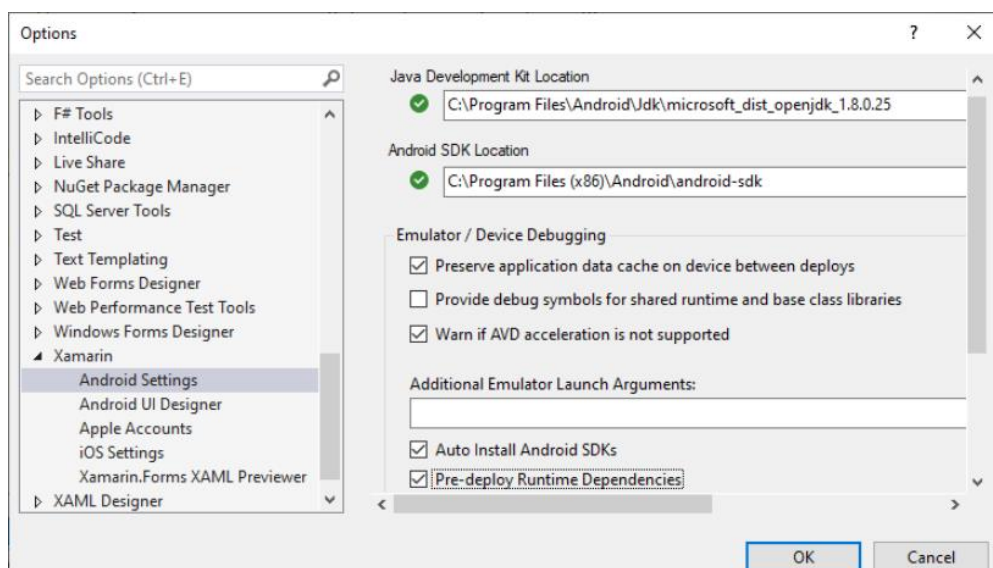


Рисунок 6 – Налаштування параметрів проекту для ОС Android

Перед запуском необхідно налаштувати деякі параметри, пов'язані з Android. Перейдемо до меню Tools -> Options та у вікні виберемо пункт Xamarin -> Android Settings (рис. 6). Налаштовуємо розташування Java SE SDK та Android SDK, а також налаштування емулятора.

Далі переходимо у властивості проекту для Android, і в налаштуваннях встановити версію середовища Android, яке буде використовуватися для компіляції програми. Слід враховувати, що остання версія Xamarin Forms не підтримує версії нижче за Android 10 як цільову версію Android.

Важливу роль відіграють налаштування маніфесту (рис. 7), де можемо встановити різні дозволи, іконку та назву програми, цільову та мінімальну версію Android, які будуть підтримуватись додатком, та інші налаштування.

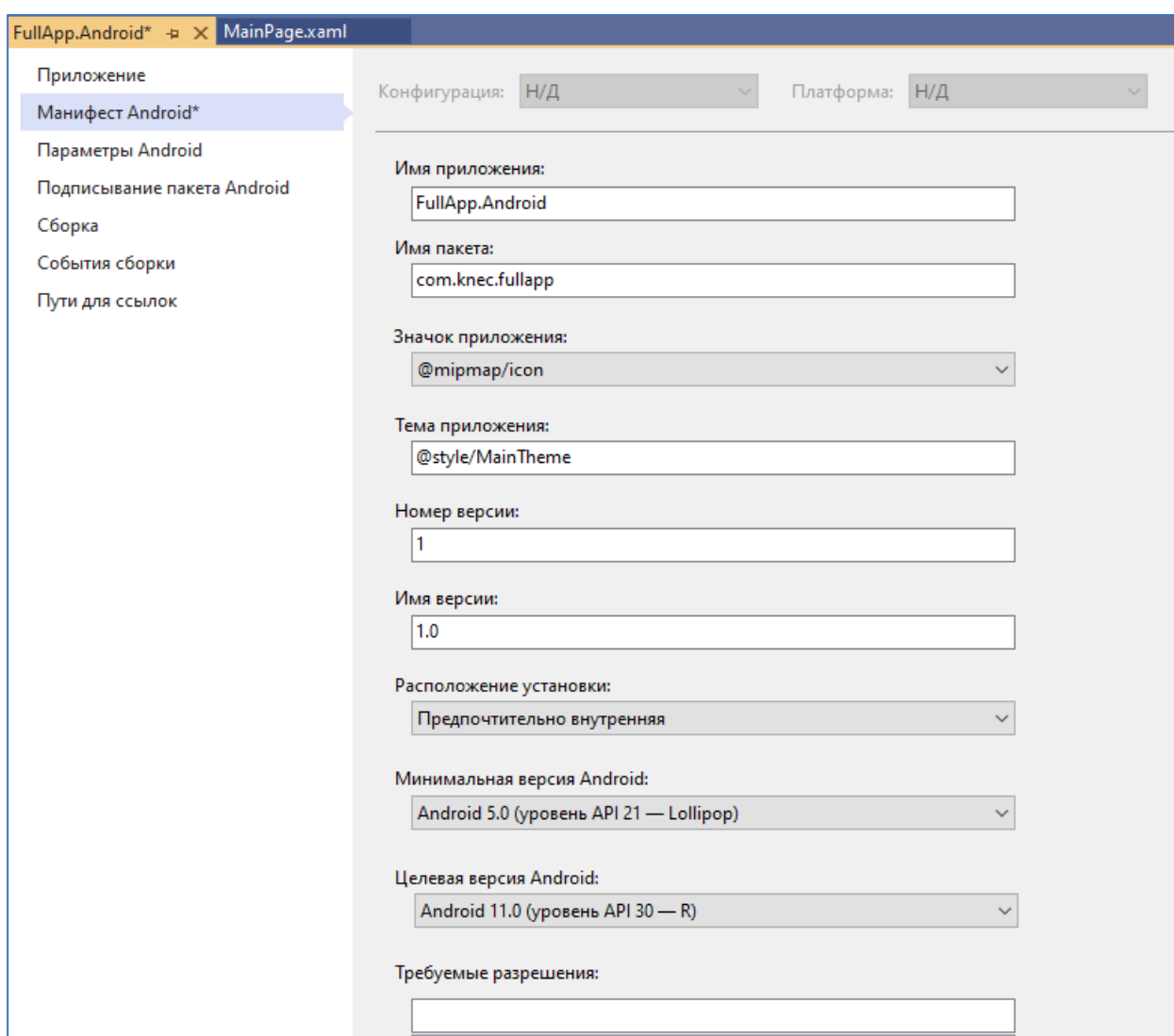


Рисунок 7 – Налаштування файлу AndroidManifest.xml

Версію середовища варто враховувати, якщо маємо певний мобільний пристрій, який використовується для тестування. Якщо його версія Android нижче мінімальної, Visual Studio може просто не побачити цей пристрій, навіть

якщо він підключений через USB до комп'ютера. Якщо все налаштовано правильно, то в панелі інструментів Visual Studio відобразить підключений пристрій (рис. 8).

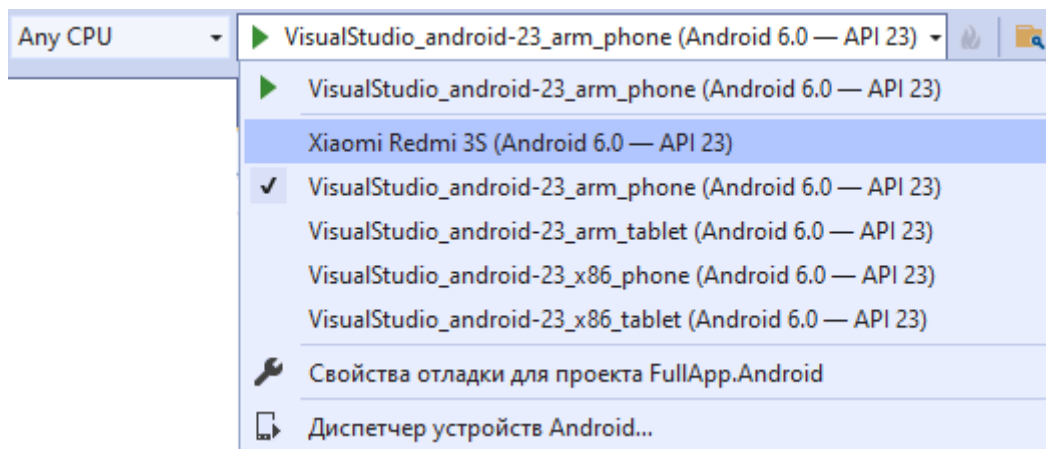


Рисунок 8 – Вибір пристрою для тестування додатку

У даному випадку це пристрій Xiaomi Redmi 3S. У списку можуть відображатися інші підключені пристрої та емулятори. Якщо необхідно встановити нову версію Android API або компоненти з Android SDK, можна перейти до меню Tools -> Android -> Android SDK Manager і там відзначити потрібні компоненти для встановлення.

Переходимо до файлу MainPage.xaml та вносимо деякі зміни в коді.

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="FullApp.MainPage">

    <StackLayout>
        <Frame BackgroundColor="#2196F3" Padding="24" CornerRadius="0">
            <Label Text="Welcome to Xamarin.Forms!"
                  HorizontalTextAlignment="Center"
                  TextColor="White" FontSize="36"/>
        </Frame>
        <Label Text="Start developing now"
              FontSize="Title" Padding="30,10,30,10"/>
        <Label Text="Це підписаний додаток"
              FontSize="16" Padding="30,0,30,0"/>
    </StackLayout>
</ContentPage>
```

Запускаємо проект (рис. 9)

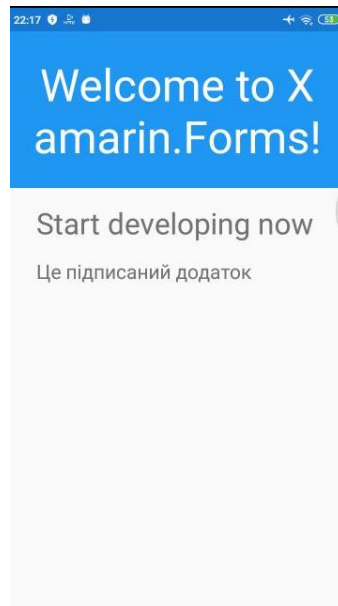


Рисунок 9 – Запуск додатку на фізичному пристрої

#### 4. Створення інсталяційного файлу

Додаток необхідно підписувати та збирати в реліз для отримання повноцінного додатку, який можна встановлювати на будь-якому пристрої. Для цього змінюємо конфігурацію зборки з Debug на Release (рис. 10)

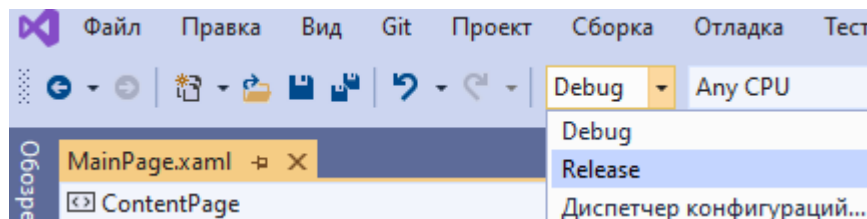


Рисунок 10 – Змінюємо конфігурацію зборки

Вибираємо іконку для додатку. Відкриваємо в провіднику папку з Resources (рис. 11).

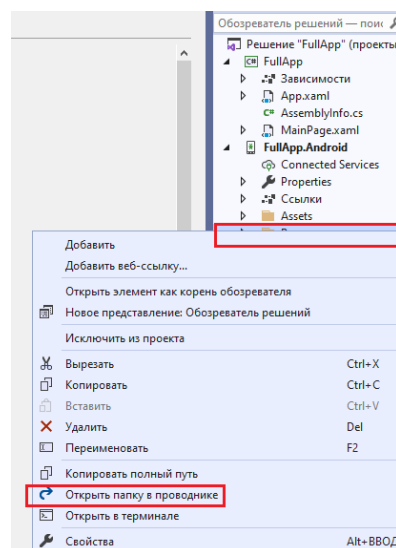


Рисунок 11 – Вибір переходу до папки з Resources проекту



Перевіряємо наявність файлів іконок в папках для різних екранів (рис. 12).

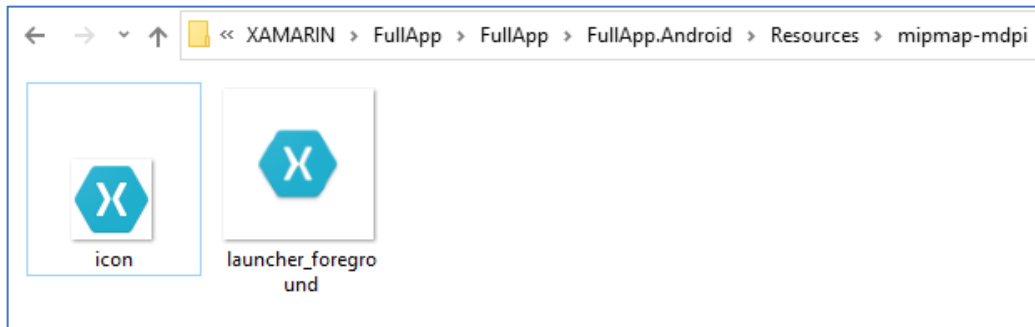


Рисунок 12 – Іконки додатку в папці Resources

Далі викликаємо контекстне меню для проекту FullApp.Android. Вибираємо команди Clean, Rebuild, Archive (рис. 13). Результат виконання команди Archive наведений на рис. 14.

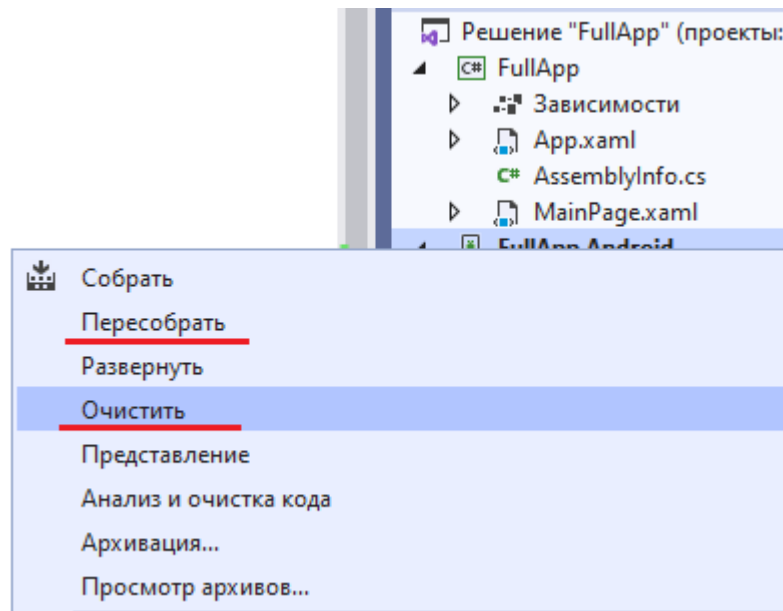


Рисунок 13 – Контекстне меню проекту FullApp.Android

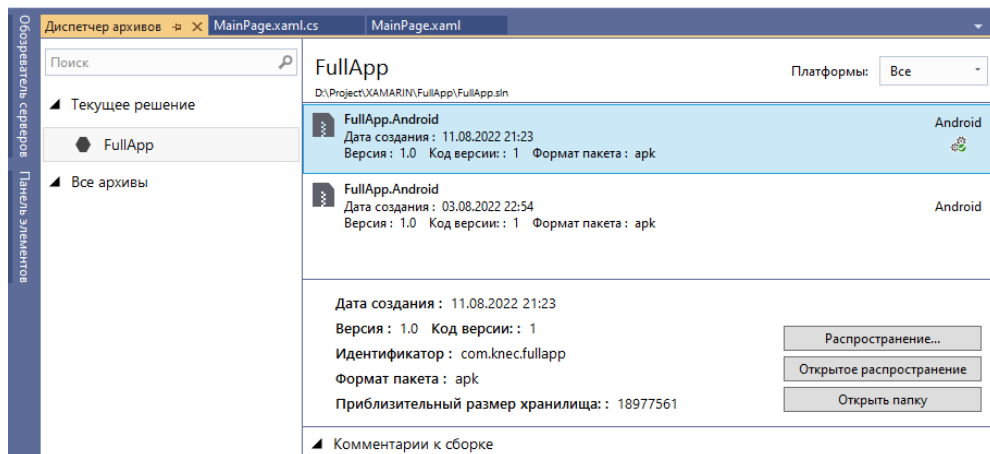


Рисунок 14 – Результат виконання команди Archive

Натискаємо на кнопку «Open Folder», переглядаємо файли архіву проекту (рис. 15).

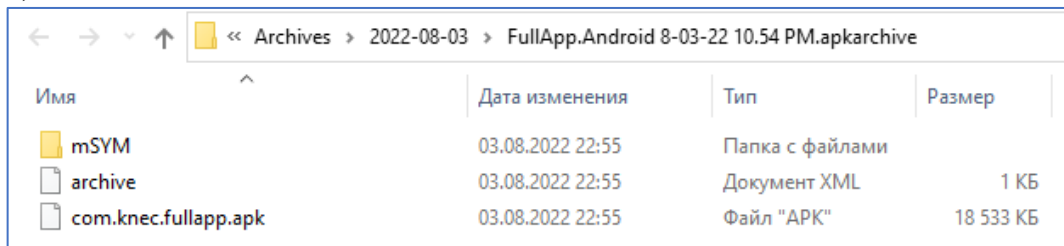


Рисунок 15 –Файлы проекту після команди Archive

Далі натискаємо кнопку «Distribute» та вибираємо режим «Спеціальний» (рис. 16).

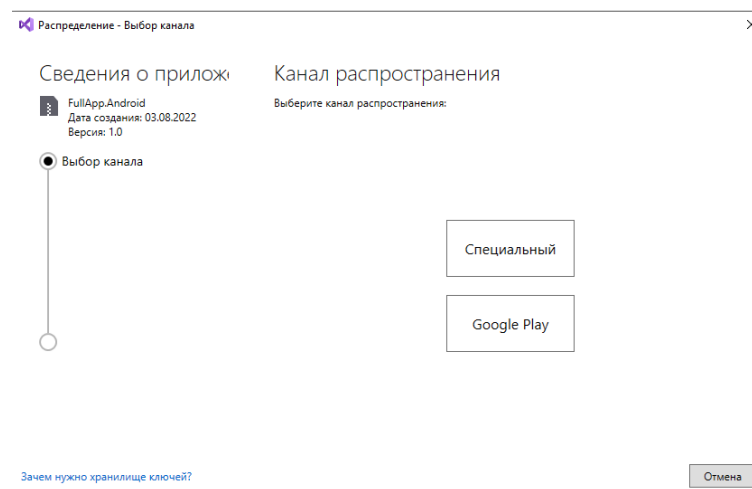


Рисунок 17 – Канал розповсюдження. Спеціальний

Нажимаємо кнопку «+» (рис. 18) для створення ключа для підписування додатку та заповнюємо форму, що відкрилась (рис. 19).

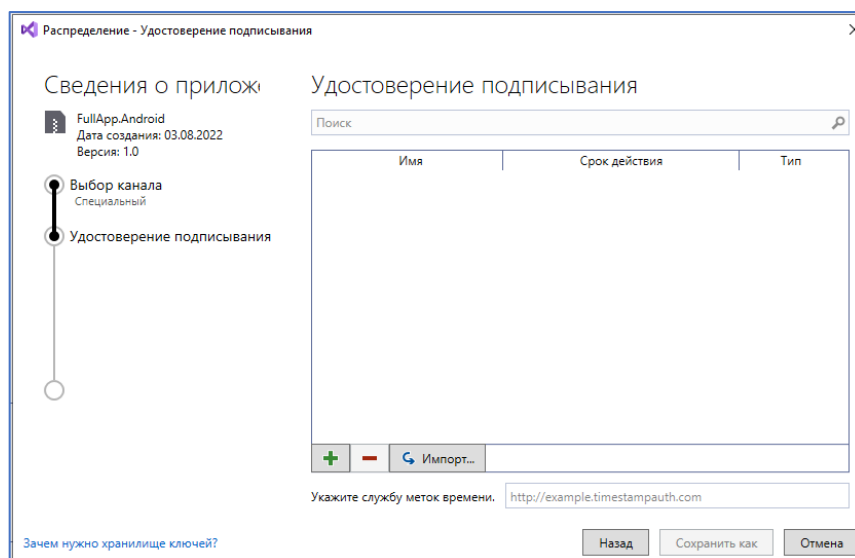


Рисунок 18 – Створення ключа для підписування додатку

Рисунок 19 – Форма для створення ключа додатку

Створюємо ключ та вибираємо папку де буде збережений кінцевий арк файл (рис. 20), який можна встановлювати на телефон.

Рисунок 20 – Вибір папки для збереження підписаного арк файлу

## Контрольні питання

1. Які переваги надає розробникам платформа Xamarin Forms?
2. Який проєкт є головним, що містить основну логіку програми?
3. Які основні файли входять до головного проєкту та їх призначення?
4. Які є способи тестування додатку, що розроблений в Xamarin Forms?
5. Яке призначення файлу AndroidManifest.xml?

6. Як підписати додаток, що розроблений в Xamarin Forms?
7. Як додати іконку до додатку?
8. Як створити інсталяційний файл додатку, що розроблений в Xamarin Forms?

### **Список використаних джерел**

1. Visual Studio 2019. <https://visualstudio.microsoft.com/downloads>.
2. Visual Studio for Mac. URL: <https://visualstudio.microsoft.com/vs/mac>.
3. Publishing an Application. URL: <https://docs.microsoft.com/en-us/xamarin/android/deploy-test/publishing>.
4. Signing the Android Application Package. URL: <https://docs.microsoft.com/en-us/xamarin/android/deploy-test/signing/?tabs=windows>.
5. Optimizing Xamarin Apps & Libraries with the Linker. URL: <https://montemagno.com/optimizing-xamarin-apps-libraries-with-the-linker/>