**Отчет по Главе 9: Работа с платформозависимым кодом и .NET MAUI Essentials**

**1. Введение**

Данная глава посвящена методам работы с кодом, специфичным для каждой платформы, и использованию библиотеки .NET MAUI Essentials. Несмотря на то, что .NET MAUI позволяет писать большую часть кода кроссплатформенно, часто возникает необходимость в доступе к уникальным API или поведении каждой операционной системы (Windows, Android, iOS). Понимание этих механизмов позволяет расширять функциональность приложений, используя все возможности целевых платформ, а .NET MAUI Essentials предоставляет удобный кроссплатформенный доступ ко многим из этих нативных функций.

**2. Основная часть**

**2.1. Основные концепции и принципы**

* **Условная компиляция:**
  + Позволяет включать или исключать блоки кода во время компиляции на основе предопределенных символов для каждой платформы (WINDOWS, ANDROID, IOS, MACCATALYST).
  + Используются директивы препроцессора #if, #elif, #else, #endif.
  + Подход удобен для небольших фрагментов платформо-специфичного кода внутри общего файла.
* **Частичные классы и методы (Partial Classes and Methods):**
  + Позволяют разделить определение одного класса на несколько файлов.
  + Это основной механизм в .NET MAUI для организации платформо-зависимого кода. Общая часть класса (например, интерфейс или объявление частичного метода) находится в общем коде, а реализации для каждой платформы — в соответствующих папках Platforms/\*.
  + Компилятор объединяет все частичные файлы в один класс во время сборки для конкретной платформы.
* **.NET MAUI Essentials:**
  + Набор кроссплатформенных API для доступа к нативным функциям устройства. Является частью .NET MAUI и не требует отдельной установки.
  + Примеры API:
    - DeviceInfo: Информация об устройстве (модель, производитель, платформа, версия ОС).
    - AppInfo: Информация о приложении (имя, версия, пакет).
    - Connectivity: Проверка состояния сетевого подключения (доступ к интернету, тип подключения).
    - HapticFeedback: Управление виброоткликом устройства.
    - Share: Вызов нативного диалогового окна "Поделиться" для отправки текста, ссылок или файлов.
    - Browser: Открытие ссылок в системном браузере.
    - Многие другие, включая Geolocation, Permissions, Preferences и т.д.

**2.2. Примеры кода с пояснениями**

Для демонстрации была создана страница PlatformDemoPage.xaml и сервис PlatformService.

**2.2.1. Условная компиляция**  
В PlatformDemoPage.xaml.cs используется метод для определения платформы с помощью директив препроцессора.

Generated csharp

// Фрагмент PlatformDemoPage.xaml.cs

private string GetPlatformNameViaConditionalCompilation()

{

#if WINDOWS

return "Windows (Conditional)";

#elif ANDROID

return "Android (Conditional)";

#elif IOS

return "iOS (Conditional)";

#elif MACCATALYST

return "MacCatalyst (Conditional)";

#else

return "Unknown Platform (Conditional)";

#endif

}

*Пояснение:* В зависимости от целевой платформы, на которую компилируется приложение, будет включен только один из блоков return, предоставляя соответствующее имя платформы.

**2.2.2. Частичные классы (Partial Classes)**  
Создан сервис PlatformService с использованием частичных классов.

Generated csharp

// Общая часть: Services/PlatformService.cs

namespace HelloMauiApp.Services;

public partial class PlatformService

{

public partial string GetPlatformName();

}

// Реализация для Windows: Platforms/Windows/PlatformService.cs

namespace HelloMauiApp.Services;

public partial class PlatformService

{

public partial string GetPlatformName()

{

return "Windows";

}

}

*Пояснение:* Общая часть объявляет сигнатуру метода GetPlatformName. Реализация этого метода предоставляется в платформо-специфичных файлах. Во время сборки для Windows будет использована реализация из папки Platforms/Windows.

**2.2.3. .NET MAUI Essentials**  
На странице PlatformDemoPage отображается информация, полученная от различных API Essentials, и кнопки вызывают соответствующие действия.

Generated csharp

// Фрагмент PlatformDemoPage.xaml.cs

private void DisplayPlatformInfo()

{

// ...

DeviceInfoModelLabel.Text = DeviceInfo.Current.Model;

DeviceInfoPlatformLabel.Text = $"{DeviceInfo.Current.Platform} {DeviceInfo.Current.VersionString}";

AppInfoVersionLabel.Text = AppInfo.Current.VersionString;

ConnectivityLabel.Text = Connectivity.Current.NetworkAccess.ToString();

// ...

}

private async void ShareText\_Clicked(object sender, EventArgs e)

{

await Share.Default.RequestAsync(new ShareTextRequest

{

Title = "Share MAUI Info",

Text = "Check out .NET MAUI!",

Uri = "https://aka.ms/dotnet-maui"

});

}

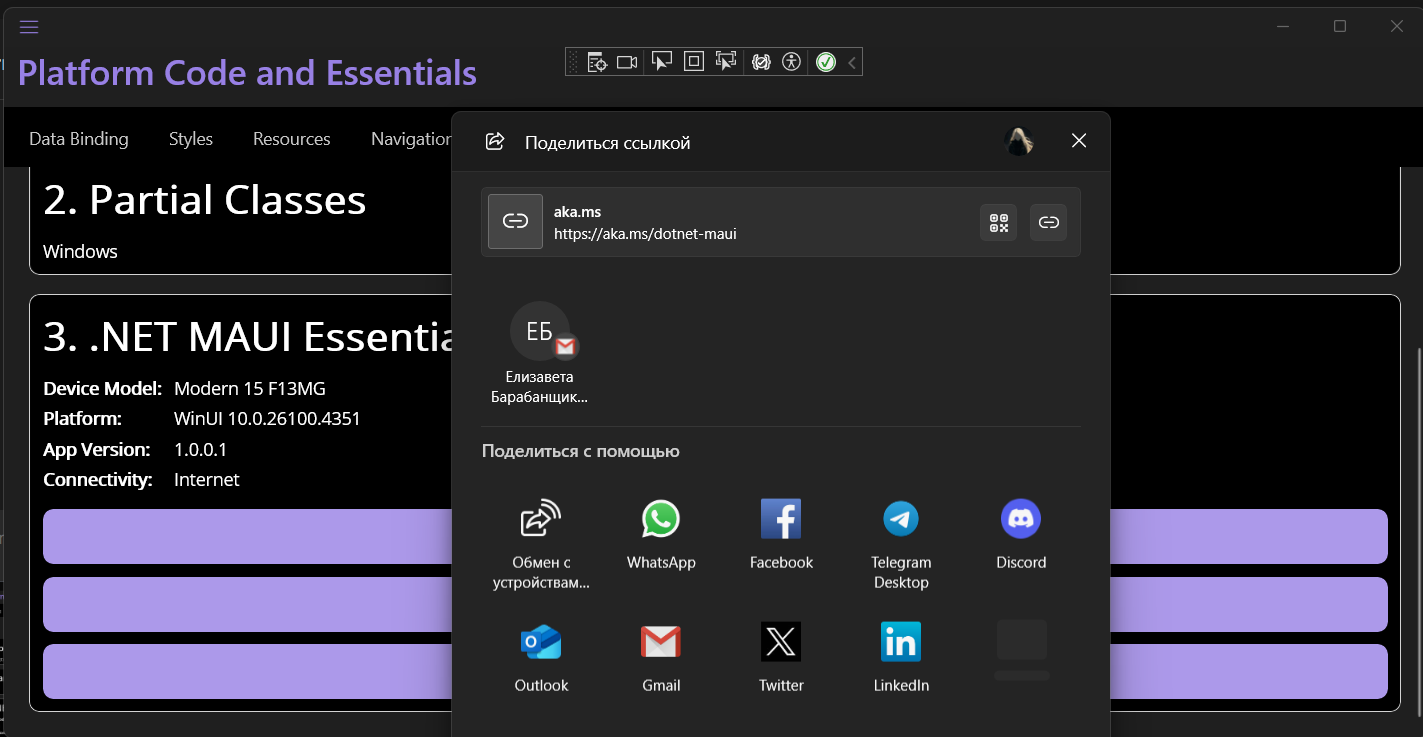
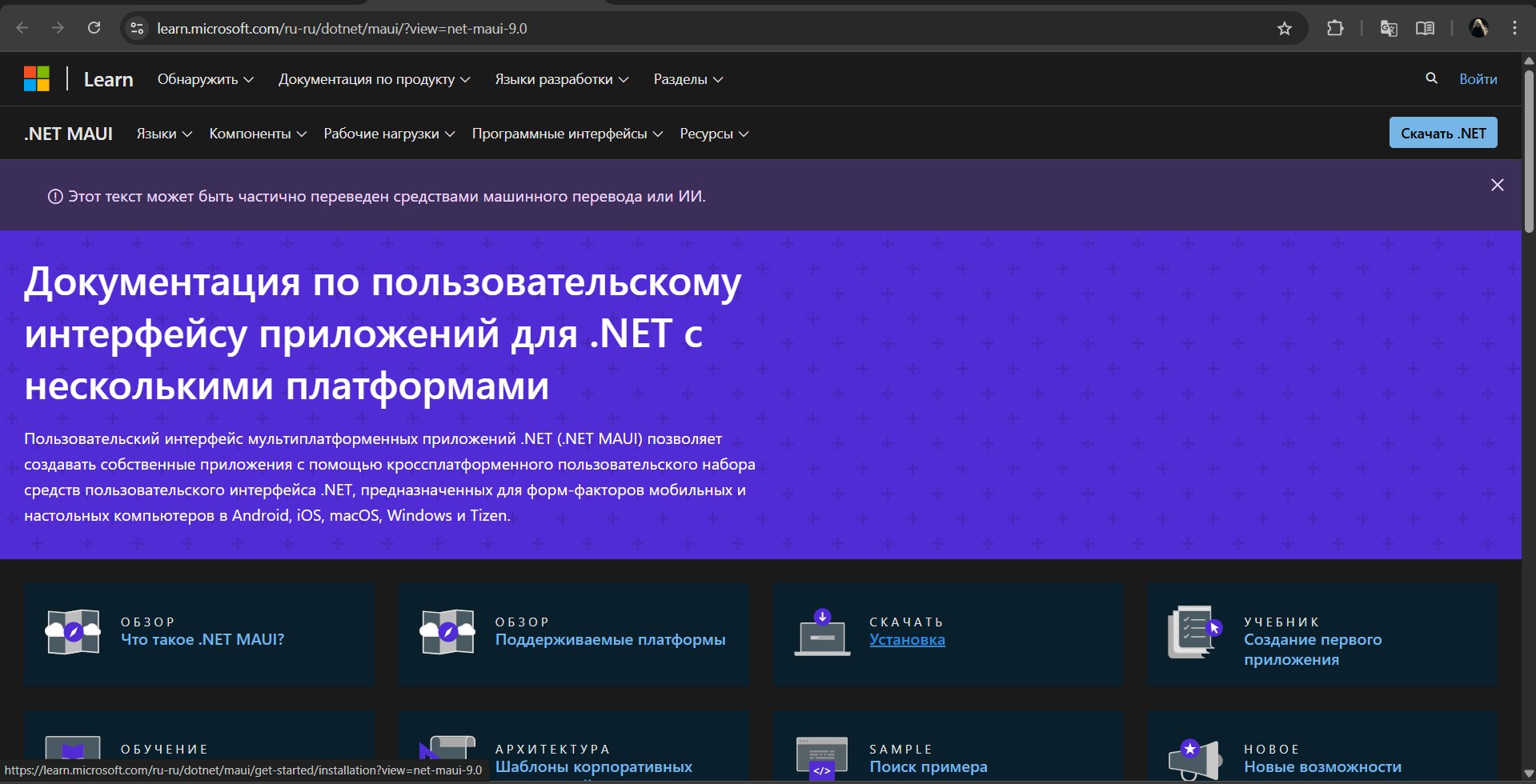
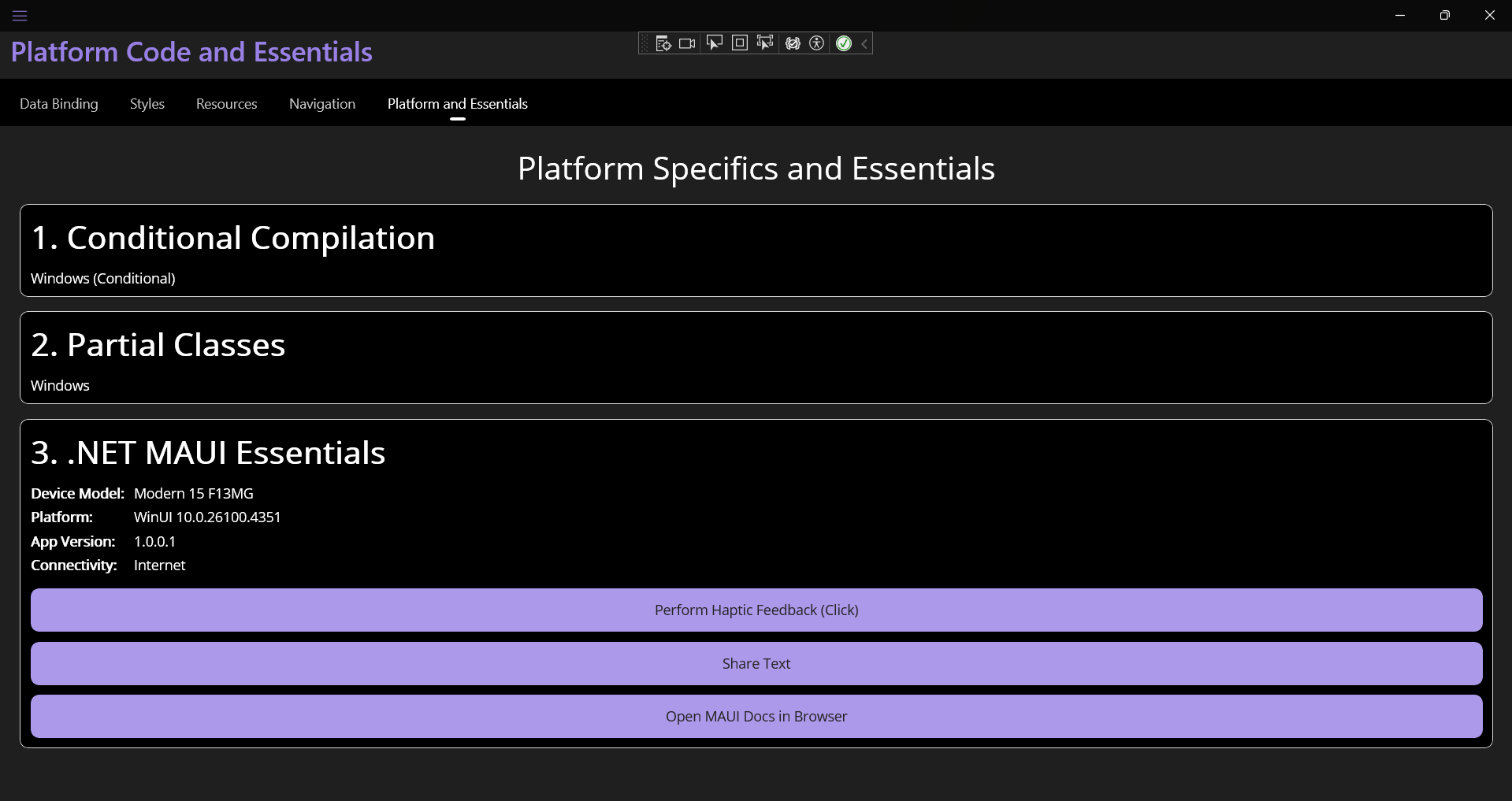
*Пояснение:* Код демонстрирует простоту получения информации об устройстве и вызова нативных функций, таких как "Поделиться", с помощью статических классов из библиотеки Essentials.

**3. Заключение**

Владение инструментами для работы с платформо-специфичным кодом является обязательным для серьезной кроссплатформенной разработки. Условная компиляция и, в особенности, частичные классы предоставляют элегантные способы организации такого кода. Библиотека .NET MAUI Essentials значительно упрощает разработку, предоставляя унифицированный доступ к десяткам нативных API, что сокращает время разработки и повышает качество приложений, включая настольные, где также важен доступ к системным функциям.

**4. Скриншоты и примеры**

Демонстрация работы была выполнена на странице PlatformDemoPage, запущенной на платформе Windows.

*Рис. 9.1 - 9.3. Демонстрация определения платформы и работы API .NET MAUI Essentials (Browser, Share).*

**Процесс выполнения примеров:**

1. Создана страница PlatformDemoPage.
2. Реализован сервис PlatformService с использованием частичных классов для определения имени платформы.
3. На странице продемонстрировано определение платформы через условную компиляцию и через вызов PlatformService.
4. С помощью .NET MAUI Essentials на страницу выведена информация об устройстве, приложении и сетевом подключении.
5. Добавлены кнопки для вызова нативных функций: виброотклик (HapticFeedback), открытие браузера (Browser) и вызов диалога "Поделиться" (Share).
6. Страница добавлена в навигацию AppShell и протестирована.  
   Все функции работали корректно на платформе Windows. Нажатие кнопок успешно вызывало соответствующие системные действия.

**5. Документация**

* **Ссылка на GitHub с исходным кодом:** <https://github.com/Nikita-Levuskin/C-sharp-practic/tree/main/1>
* **Описание выполненных шагов:**
  1. Изучены материалы по теме "Работа с платформозависимым кодом" и ".NET MAUI Essentials" (глава 9) с ресурса [metanit.com/sharp/maui/](https://www.google.com/url?sa=E&q=https%3A%2F%2Fmetanit.com%2Fsharp%2Fmaui%2F).
  2. Создана страница PlatformDemoPage для демонстрации.
  3. Реализованы примеры условной компиляции и частичных классов.
  4. Продемонстрировано использование API из .NET MAUI Essentials: DeviceInfo, AppInfo, Connectivity, HapticFeedback, Share, Browser.
  5. Проведено тестирование на платформе Windows и подготовлены скриншоты.