

## ПРАКТИЧНА РОБОТА № 9. СТВОРЕННЯ TOOLBAR MENU ЗАСОБАМИ XAMARIN.FORMS

**Мета:** набуття здобувачами освіти навичок створення та відлагодження кросмобільного додатку з Toolbar Menu засобами Xamarin.Forms у середовищі Visual Studio 2019; створення діалогового вікна DisplayAlert.

### Короткі теоретичні відомості

Клас Xamarin.Forms ToolbarItem – це особливий тип кнопки, яку можна додати до колекції Page об'єкта ToolbarItems. Кожен ToolbarItem об'єкт відобразиться у вигляді кнопки на панелі навігації програми. Екземпляр ToolbarItem може мати значок і відобразиться як основний або додатковий пункт меню. Клас ToolbarItem успадковується від MenuItem. На рис. 1 показано ToolbarItem об'єкти на панелі навігації в iOS та Android.

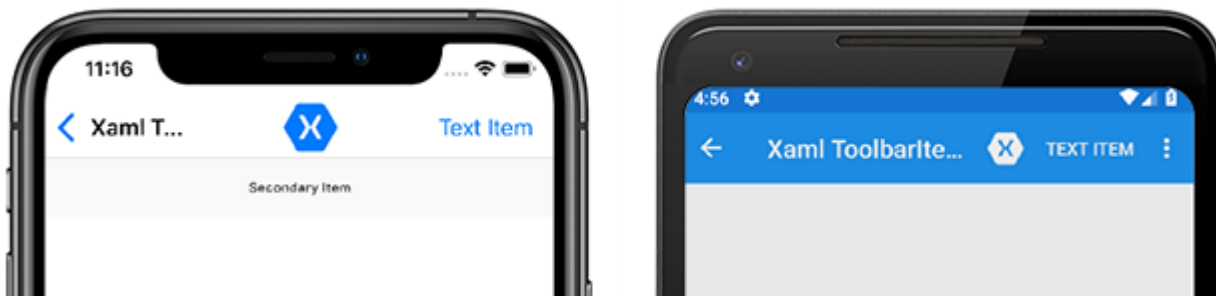


Рисунок 1 – ToolbarItem об'єкти на панелі навігації

Клас ToolbarItem визначає такі властивості:

- **Order** (ToolbarItemOrder) – це значення переліку, яке визначає, чи відображається ToolbarItem екземпляр в основному або додатковому меню;
- **Priority** integer – це значення, що визначає порядок відображення елементів у Page колекції об'єкта ToolbarItems;

Клас ToolbarItem успадковує такі властивості, що зазвичай використовуються від MenuItem класу:

- **Command** – це об'єкт ICommand, що дозволяє прив'язувати дії користувача, такі як торкання пальця або клацання, до команд, визначених у моделі подання;
- **CommandParameter** object – це значення, що вказує параметр, який повинен бути переданий у Command;
- **IconImageSource** ImageSource – значення, що визначає піктограму відображення об'єкта ToolbarItem;
- **Text** – це об'єкт string, що визначає ToolbarItem текст, що відображається.

Об'єкт `ToolBarItem` можна створити у XAML. Властивості `Text` можна `IconImageSource` встановити, щоб визначити, як кнопка відображається на панелі навігації. У цьому прикладі показано, `ToolBarItem` як створити екземпляр з деякими загальними наборами властивостей і додати його до колекції `ContentPage.ToolbarItems`:

```
<ContentPage.ToolbarItems>
  <ToolBarItem Text="Example Item"
              IconImageSource="example_icon.png"
              Order="Primary"
              Priority="0" />
</ContentPage.ToolbarItems>
```

У `ToolBarItem` буде показано об'єкт із текстом, значком та першим в основній області навігаційної панелі. Можна `ToolBarItem` також створити в коді та додати до колекції `ToolBarItems`:

```
ToolBarItem item = new ToolBarItem
{
    Text = "Example Item",
    IconImageSource = ImageSource.FromFile("example_icon.png"),
    Order = ToolBarItemOrder.Primary,
    Priority = 0
};

// "this" refers to a Page object
this.ToolbarItems.Add(item);
```

Клас `ToolBarItem` успадковує `Clicked` подію від `MenuItem` класу. Обробник подій можна підключити до події `Clicked`, щоб реагувати на торкання або клацати `ToolBarItem` екземпляри в XAML:

```
<ToolBarItem ...
    Clicked="OnItemClicked" />
```

Обробник подій також можна приєднати до коду:

```
ToolBarItem item = new ToolBarItem { ... }
item.Clicked += OnItemClicked;
```

У наступному коді показаний приклад реалізації `OnItemClicked` обробника події:

```
void OnItemClicked(object sender, EventArgs e)
{
```

```

ToolBarItem item = (ToolBarItem)sender;
messageLabel.Text = $"You clicked the \"{item.Text}\" toolbar item.";
}

```

Якщо для властивості Order встановлено значення Primary, ToolBarItem об'єкт буде відображатися на головній навігаційній панелі на всіх платформах. Якщо для властивості Order встановлено значення Secondary, то поведінка залежить від платформ. У UWP та Android Secondary меню елементів відображається як три точки, які можна торкнутися або клацнути, щоб відобразити елементи у вертикальному списку. У iOS Secondary меню елементів відображається під навігаційною панеллю у вигляді горизонтального списку. На рис. 2 показано додаткове меню в iOS та Android.



Рисунок 2 – Secondary меню елементів для iOS та Android

### Хід роботи

1. Створюємо проект ToolBarMenu. Вибираємо шаблон Blank. При створенні проекту в опції Platform вибираємо платформу Android під яку буде створюватись проект.

2. Відкриваємо клас App.xaml.cs (рис. 3) в якому присвоюємо об'єкту MainPage екземпляр класу NavigationPage та в конструкторі класу NavigationPage передаємо екземпляр сторінки яку будемо запускати.

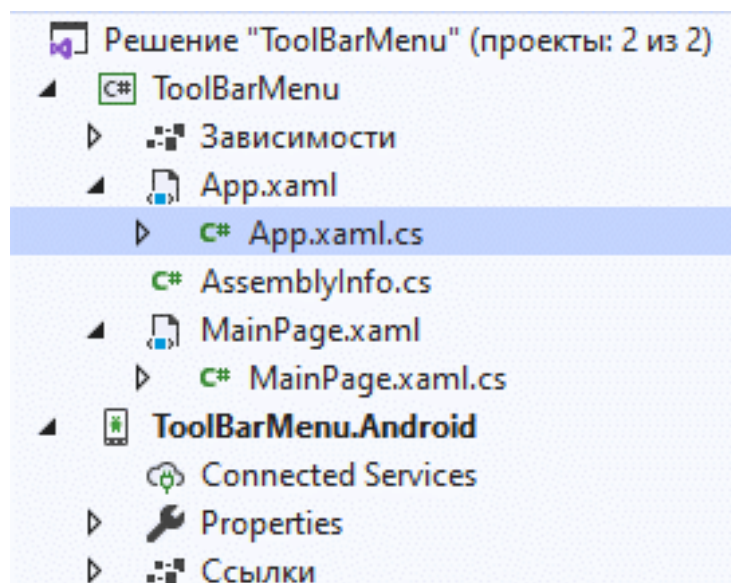


Рисунок 3 – Структура проекту ToolBarMenu. Відкриття файлу App.xaml.cs

```

public App()
{
    InitializeComponent();
    MainPage = new NavigationPage(new ToolBarMenu.MainPage());
}

```

Переходимо до MainPage.xaml. Додамо Title = «ТВMenu» та видаляємо частину коду. Зміст файлу MainPage.xaml після редагування:

```

<?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="ToolBarMenu.MainPage"
             Title="ТВMenu">
    <Label Text="Welcome to Xamarin.Forms!"
          VerticalOptions="Center"
          HorizontalOptions="Center" />
</ContentPage>

```

3. Відкриваємо файл MainPage.xaml.cs. У конструкторі класу MainPage створюємо екземпляр класу ToolbarItem. В ініціалізаторі класу проініціалізуємо декілька властивостей: Text, Order = ToolbarItemOrder.Primary (відповідає за пріоритет елемента меню), Priority = 0 (щоб елемент меню був вверху списку), Icon. Рисунок phone.png скачується з Інтернету та додаємо до папки Drawable (рис. 4).

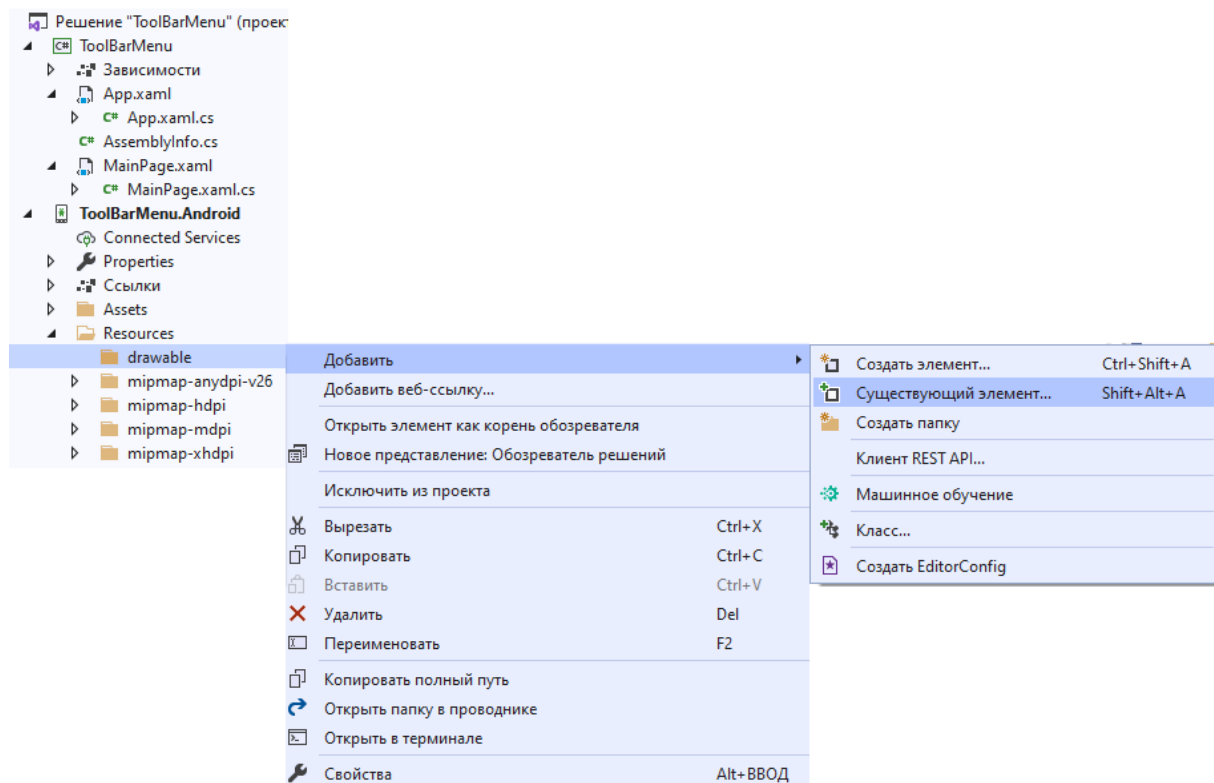


Рисунок 4 – Додавання файлу в папку drawable

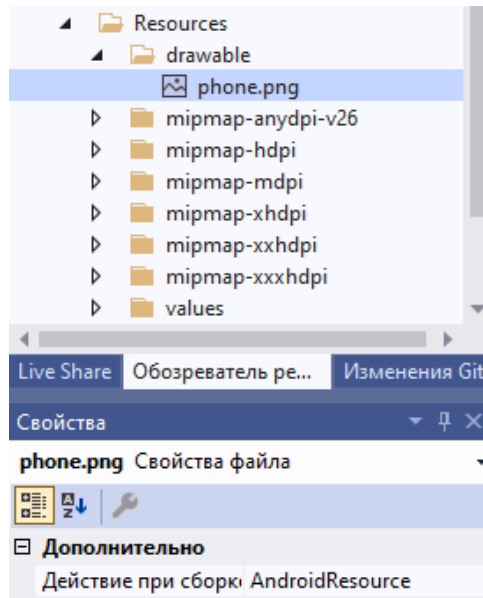


Рисунок 5 – Властивості файлу phone.png

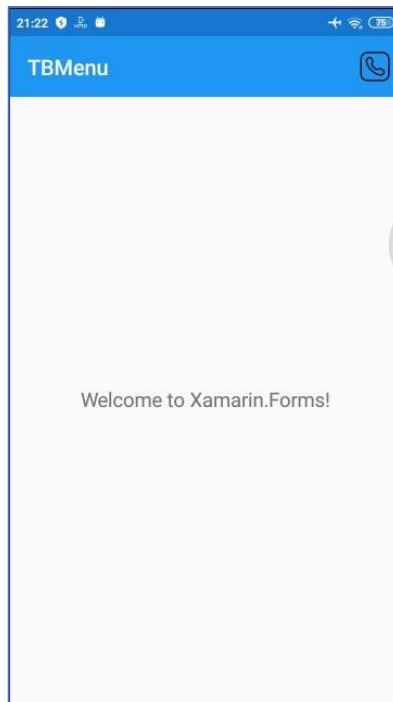


Рисунок 6 – Вікно мобільного телефону з запущеним додатком

Перевіряємо, щоб властивість файлу phone.png «Действие при сборке» було встановлено як «AndroidResource».

```
public MainPage()
{
    InitializeComponent();
    ToolbarItem tb = new ToolbarItem
    {
        Text = "Зателефонувати",
        Order = ToolbarItemOrder.Primary,
```

```

        Priority = 0,
        IconImageSource = "phone.png"
    };

    ToolbarItems.Add(tb);
}

```

4. Запускаємо проект на виконання (рис. 6).

5. Додаємо до проекту пункти меню та метод обробки натиснення на елемент меню `tb`, який буде викликати діалогове вікно `DisplayAlert` для користувача.

```

public MainPage()
{
    InitializeComponent();
    ToolbarItem tb = new ToolbarItem
    {
        Text = "Зателефонувати",
        Order = ToolbarItemOrder.Primary,
        Priority = 0,
        IconImageSource = "phone.png"
    };

    tb.Clicked += Tb_Clicked;
    ToolbarItem tb1 = new ToolbarItem
    {
        Text = "Створити новий контакт",
        Order = ToolbarItemOrder.Secondary,
        Priority = 1,
    };

    ToolbarItem tb2 = new ToolbarItem
    {
        Text = "Видалити",
        Order = ToolbarItemOrder.Secondary,
        Priority = 2,
    };

    ToolbarItem tb3 = new ToolbarItem
    {
        Text = "Про програму",
        Order = ToolbarItemOrder.Secondary,
        Priority = 3,
    };
    ToolbarItems.Add(tb);
    ToolbarItems.Add(tb1);
    ToolbarItems.Add(tb2);
    ToolbarItems.Add(tb3);
}

```

```

private async void Tb_Clicked(object sender, EventArgs e)
{
    await DisplayAlert("Виклик", "Йде набір номера ...",
        "Скинути");
}
}

```

6. Запускаємо проект на виконання. Перевіряємо роботу елементів ToolbarMenu (рис. 7).

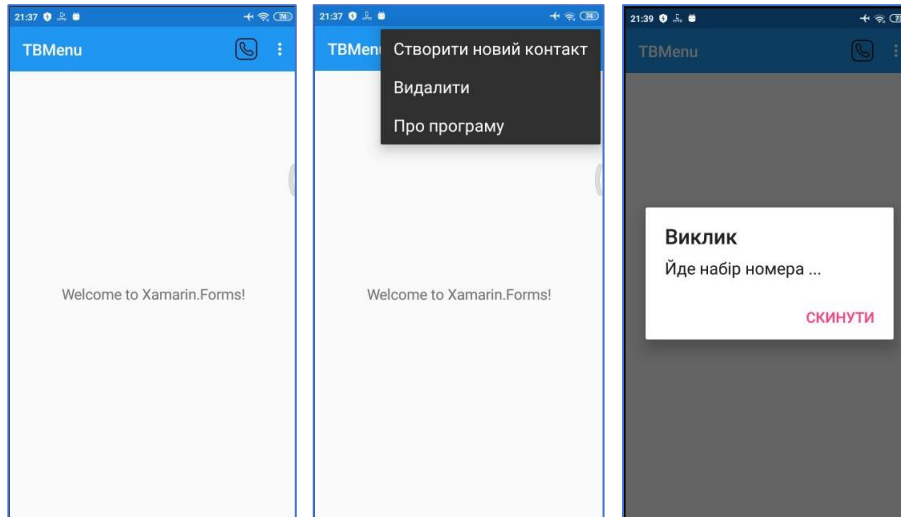


Рисунок 7 – Результати роботи додатку ToolBarMenu

### Контрольні питання

1. Як додати до об'єкту ToolbarItem значок?
2. Як задати об'єкту ToolbarItem основний пункт меню?
3. Як задати об'єкту ToolbarItem додатковий пункт меню?
4. Які властивості має об'єкт класу ToolbarItem?
5. Як створити об'єкт ToolbarItem із текстом та значком у файлі MainPage.xaml?
6. Як створити об'єкт ToolbarItem із текстом та значком у файлі MainPage.xaml.cs?
7. Що задає властивість Priority для об'єкта класу ToolbarItem?
8. Як додаток до проекту метод обробки натиснення на елемент меню?
9. Як викликати діалогове вікно DisplayAlert за натисненням на елемент меню?

### Список використаних джерел

1. Xamarin.Forms ToolbarItem. URL: <https://docs.microsoft.com/en-us/xamarin/xamarin-forms/user-interface/toolbaritem>
2. Xamarin Forms Toolbar. URL: <https://www.xamarinhelp.com/xamarin-forms-toolbar/>
3. Page.DisplayAlert Method. URL: <https://docs.microsoft.com/en-us/dotnet/api/xamarin.forms.page.displayalert?view=xamarin-forms>