

**Практична робота №2**  
**Технологія Windows Presentation Foundation**  
**Система оцінювання**

№	Тема	К-ть балів
1.	<i>Захист принаймні одного завдання з роботи</i>	1
2.	Завдання 1	0,4*
3.	Завдання 2	0,2*
4.	Завдання 3	0,8*
5.	Завдання 4	0,5*
6.	Завдання 5	1*
7.	Завдання 6	0,8*
7.	<i>Здача звіту</i>	0,3
	<b>Всього</b>	<b>5</b>

\* – діє бонусна система

**Завдання на практичне заняття**

- Використовуючи код зі статті <https://www.wpfutorial.net/LogicalAndVisualTree.html>, виведіть повне візуальне дерево для кнопки (Button) та напису (Label).
- Виконайте покрокову інструкцію [https://www.bestprog.net/uk/2017/12/07/an-example-of-creating-a-new-application-that-supports-the-wpf-windows-presentation-foundation-platform-creating-a-hello-application-hello-world-overview-of-the-main-project-files\\_ua/](https://www.bestprog.net/uk/2017/12/07/an-example-of-creating-a-new-application-that-supports-the-wpf-windows-presentation-foundation-platform-creating-a-hello-application-hello-world-overview-of-the-main-project-files_ua/)
- (Внутрішні [розширення](#) [розмітки](#) XAML) Розширення розмітки дозволяють аналізатору XAML отримувати значення властивості з виділеного зовнішнього класу.
  - {x:Type} дозволяє отримати доступ до опису метаданих вказаного елемента. Виведіть в окремих написах метадані (повністю кваліфіковані імена типів) для Button і Label. Порівняйте результати з виводом задачі 1.
  - {x:Static} забезпечує посилання на статичний член потрібного типу. Оберіть кілька статичних властивостей з класу [Environment](#) (OSVersion, ProcessorCount тощо) та виведіть їх значення на відповідні написи (label) у вікні
  - {x:Null}. Установіть значення фону кнопки Null
  - {x:Code}. Вбудуйте в XAML-код кнопки її обробник натиснення. Рекомендується, щоб при натисненні на кнопку виводилось повідомлення щодо того, хто та коли виконав натиснення.
  - {x:Array}. Виведіть в елемент ListBox масив значень. Для цього потрібно використати властивість ItemsSource:

```
<ListBox>
    <ListBox.ItemsSource>
        ...
    </ListBox.ItemsSource>
</ListBox>
```

4. (Власне розширення розмітки) Зберіть проект відповідно до [статті](#) та прокоментуйте у звіті описаний код. Запропонуйте та внесіть зміни у наповнення проекту.
5. ([Властивості залежностей](#)) Технологія WPF ввела нову систему властивостей, яка відрізняється від стандартних CLR-властивостей. Класична властивість є лише обгорткою навколо приватного поля класу. У той же час ідея властивостей залежності полягає в обчисленні значення властивості на основі інших зовнішніх вхідних даних: стилів, тем, системних властивостей, анімацій тощо. Детальніше про переваги властивостей залежності можна дізнатись [тут](#).  
Максимально розгляньте властивості залежностей, які має кнопка Button та застосуйте їх до неї. Зокрема, можна змінити колір фону (навіть встановити зображення на фон), текст на кнопці, спробувати роботу тригера (Style.Triggers) тощо. Створіть для кнопки власну властивість залежностей, наприклад, як у [статті](#), та детально опишіть у звіті виконані кроки.
6. ([Прикріплені властивості](#)) Прикріплені властивості вважаються різновидом властивостей залежності, проте вони визначаються в класі, відмінному від класу, де будуть використовуватись. Зокрема, для панелі Grid визначено властивість Grid.Row (індекс рядка в таблиці-гріді), проте застосовуватиметься дана властивість на інших елементах інтерфейсу (наприклад, кнопці), щоб вказати, в якому рядку таблиці їх розмістити. Приклад створення прикріпленої властивості показано в [статті](#).  
Створіть прикріплену властивість, яка застосовуватиметься до кнопки та виконуватиме якусь із трансформацій, доступних у мові XAML.