

Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»
Филиал
«Минский радиотехнический колледж»

Учебная дисциплина
«Конструирование программ и языка программирования»

Инструкция
по выполнению лабораторной работы № 29
«Разработка отладка и испытание программ с использованием событий
мыши и клавиатуры»

Минск 2024 г.

Лабораторная работа № 29

Тема работы: «Разработка отладка и испытание программ с использованием событий мыши и клавиатуры»

1 Цель работы

Сформировать умения и навыки разработки программ с использованием событий мыши и клавиатуры.

2 Задание

Выполнить задание по вариантам, представленным в пункте 5 данной инструкции.

3 Оснащение работы

ПК, среда Visual Studio 2019, MSword.

4 Основные теоретические сведения

Для элементов управления в WPF определено большое количество событий, которые условно можно разделить на несколько групп:

- события клавиатуры;
- события мыши;
- события стилуса;
- события сенсорного экрана/мультикас;
- события жизненного цикла.

Подключение обработчиков событий

Подключить обработчики событий можно декларативно в файле xaml-кода, а можно стандартным способом в файле отделенного кода.

Декларативное подключение:

```
<Button x:Name="Button1" Content="Click"
Click="Button_Click" />
```

Для определения маршрутизированных событий в классе создавалось статическое поле по типу **RoutedEvent**:

```
public static RoutedEvent СобытиеEvent
```

Это поле, как правило, имеет суффикс *Event*. Затем это событие регистрируется в статическом конструкторе.

Пример обработки события по нажатию клавиши:

```
public abstract class ButtonBase : ContentControl,
...
{
    // определение событие
    public static readonly RoutedEvent ClickEvent;

    static ButtonBase()
    {
```

```

        // регистрация маршрутизированного события
        ButtonBase.ClickEvent =
EventManager.RegisterRoutedEvent("Click",
        RoutingStrategy.Bubble,
typeof(RoutedEventHandler), typeof(ButtonBase));
        //.....
    }
    // обертка над событием
    public event RoutedEventHandler Click
    {
        add
        {
            // добавление обработчика
            base.AddHandler(ButtonBase.ClickEvent,
value);
        }
        remove
        {
            // удаление обработчика
            base.RemoveHandler(ButtonBase.ClickEven
t, value);
        }
    }
    // остальное содержимое класса
}

```

Маршрутизированные события регистрируются с помощью метода `EventManager.RegisterRoutedEvent()`. В этот метод передаются последовательно имя события, тип события (поднимающееся, прямое, опускающееся), тип делегата, предназначенного для обработки события, и класс, который владеет этим событием.

Модель событий WPF отличается от событий WinForms не только декларативным подключением. События, возникнув на одном элементе, могут обрабатываться на другом. События могут подниматься и опускаться по дереву элементов.

Так, маршрутизируемые события делятся на три вида:

- прямые (`direct events`) - они возникают и отрабатывают на одном элементе и никуда дальше не передаются. Действуют как обычные события.

- поднимающиеся (`bubbling events`) - возникают на одном элементе, а потом передаются дальше к родителю - элементу-контейнеру и далее, пока не достигнет наивысшего родителя в дереве элементов.

- опускающиеся, туннельные (`tunneling events`) - начинает отрабатывать в корневом элементе окна приложения и идет далее по вложенным элементам, пока не достигнет элемента, вызвавшего это событие.

Все маршрутизируемые события используют класс **RoutedEventArgs** (или его наследников), который предоставляет доступ к следующим свойствам:

- source: элемент логического дерева, являющийся источником события.
- originalsource: элемент визуального дерева, являющийся источником события. Обычно то же самое, что и Source
- routedevent: представляет имя события
- handled: если это свойство установлено в True, событие не будет подниматься и опускаться, а ограничится непосредственным источником.

К событиям клавиатуры можно отнести следующие события, представленные в таблице 36.1.

Таблица 36.1 – События клавиатуры в WPF

Событие	Тип события	Описание
KeyDown	Поднимающееся	Возникает при нажатии клавиши
PreviewKeyDown	Туннельное	Возникает при нажатии клавиши
KeyUp	Поднимающееся	Возникает при освобождении клавиши
PreviewKeyUp	Туннельное	Возникает при освобождении клавиши
TextInput	Поднимающееся	Возникает при получении элементом текстового ввода (генерируется не только клавиатурой, но и стилусом)
PreviewTextInput	Туннельное	Возникает при получении элементом текстового ввода

В WPF для мыши определены следующие события, представленные в таблице 36.2.

Таблица 36.2 – События мыши в WPF

Событие	Тип события	Описание
GotMouseCapture	Поднимающееся	Возникает при получении фокуса с помощью мыши
LostMouseCapture	Поднимающееся	Возникает при потере фокуса с помощью мыши

MouseEnter	Прямое	Возникает при вхождении указателя мыши в пределы элемента
MouseLeave	Прямое	Возникает, когда указатель мыши выходит за пределы элемента
MouseDown	Поднимающееся	Возникает при нажатии левой кнопки мыши
PreviewMouseDown	Туннельное	Возникает при нажатии левой кнопки мыши
MouseUp	Поднимающееся	Возникает при освобождении левой кнопки мыши
PreviewMouseUp	Туннельное	Возникает при освобождении левой кнопки мыши
MouseDown	Поднимающееся	Возникает при нажатии правой кнопки мыши
PreviewMouseDown	Туннельное	Возникает при нажатии правой кнопки мыши
MouseUp	Поднимающееся	Возникает при освобождении правой кнопки мыши
PreviewMouseUp	Туннельное	Возникает при освобождении правой кнопки мыши
MouseDown	Поднимающееся	Возникает при нажатии кнопки мыши

PreviewMouseDown	Туннельное	Возникает при нажатии кнопки мыши
MouseUp	Поднимающееся	Возникает при освобождении кнопки мыши
PreviewMouseUp	Туннельное	Возникает при освобождении кнопки мыши
MouseMove	Поднимающееся	Возникает при передвижении указателя мыши
PreviewMouseMove	Туннельное	Возникает при передвижении указателя мыши
MouseWheel	Поднимающееся	Возникает при передвижении колесика мыши
PreviewMouseWheel	Туннельное	Возникает при передвижении колесика мыши

5. Порядок выполнения работы

Номер варианта соответствует Вашему номеру по списку.

Варианты заданий:

1. Создайте список, в котором элементы могут быть выделены, как в файловых менеджерах. При клике на элемент списка выделяется только этот элемент, отменяется выделение остальных элементов.

2. Создайте список, в котором элементы могут изменять начертание символов, например, с обычного начертания на курсивное. При клике на элемент списка выделяется только этот элемент, но не отменяется начертание остальных элементов.

3. Создайте список, в котором элементы могут изменять цвет символов, например, с черного на красный. При клике на элемент списка выделяется только этот элемент, но не отменяется начертание остальных элементов.

4. Создайте список, в котором элементы могут изменять начертание символов, например, с обычного начертания на курсивное. При клике правой кнопкой мыши на элемент списка происходит смена начертания на обычное.

5. Создайте приложение, в котором по щелчку правой кнопкой мыши на форму изменяется цвет формы на тот, который выберет пользователь.
6. Создайте приложение, в котором по щелчку левой кнопки мыши на форму происходит появление надписей с номером лабораторной работы и Вашей фамилией.
7. Создайте приложение, в котором по щелчку правой кнопки мыши по форме появляется диалоговое окно для выбора цвета формы.
8. Создайте приложение, в котором по нажатию клавиши Enter на клавиатуре будет происходить подтверждение ввода данных.
9. Создайте приложение, в котором по нажатию клавиши Esc будет происходить выход из приложения.
10. Создайте приложение, в котором по нажатию на сочетание клавиш будет происходить вызов диалогового окна для сохранения документа.
11. Создайте приложение, в котором по нажатию на сочетание клавиш будет происходить открытие диалогового окна для открытия файла.
12. Создайте приложение, в котором по нажатию на сочетание клавиш будет происходить копирование и вставка текста.
13. Создайте приложение, в котором по нажатию на сочетание клавиш будет происходить открытие диалогового окна для печати документов.
14. Создайте приложение, в котором по нажатию на сочетание клавиш будет происходить открытие диалогового окна для выбора цвета формы.
15. Создайте приложение, в котором по нажатию на сочетание клавиш будет происходить открытие диалогового окна для выбора шрифта текста.

6. Форма отчета о работе

Лабораторная работа № _____

Номер учебной группы _____

Фамилия, инициалы учащегося _____

Дата выполнения работы _____

Тема работы: _____

Цель работы: _____

Оснащение работы: _____

Результат выполнения работы: _____

7. Контрольные вопросы и задания

1. На какие две группы делятся события в WPF?
2. Прямые события – это...?

8. Рекомендуемая литература

1. Рихтер, Дж. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C# / Дж. Рихтер. СПб. : Изд-во Питер, 2021. 896 с.

2. Прайс, М. Дж. C# 10 и .NET 6. Современная кросс-платформенная разработка / М. Дж. Прайс. СПб : Изд-во Питер, 2023. 848 с.

3. Васильев, А.Н. Программирование на C# для начинающих. Особенности языка / А.Н. Васильев. М. : Эксмо, 2022. 528 с.

4. Фримен, А. ASP.NET Core 3 с примерами на C# для профессионалов / А. Фримен. СПб. : Изд-во Вильямс, 2021. 1184 с.