

Практическая работа №20

Использование элементов управления для ввода данных

1 Цель работы

1.1 Научиться применять элементы управления Avalonia UI для ввода и вывода данных.

2 Литература

2.1 Avalonia documentation – Текст : электронный // AvaloniaUI, 2024. – URL: <https://docs.avaloniaui.net/>

3 Подготовка к работе

3.1 Повторить теоретический материал (см.п.2).

3.2 Изучить описание лабораторной работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Создайте новую страницу ControlsView и класс ControlsViewModel, настройте взаимодействие между ними.

5.2 Применение TextBlock

5.2.1 Расположите на странице TextBlock

5.2.2 Внутри TextBlock разместите 4 элемента Run с текстом. Для каждого из них установите различный цвет шрифта.

5.2.3 Между 2 и 3 Run расположите элемент LineBreak для переноса строки

5.2.4 Добавьте в TextBlock элемент Span, переместите внутрь его первый и второй Run. Для Span укажите другой размер шрифта и цвет фона.

5.2.5 Используя элементы Bold, Italic, Underline измените начертание фрагментов текста в TextBlock.

5.2.6 Добавьте в TextBlock изображение при помощи InlineUIContainer

```
<InlineUIContainer BaselineAlignment="Baseline">
  <Image
    Width="32"
    Height="32"
    VerticalAlignment="Top"
    Source="/Assets/avalonia-logo.ico" />
</InlineUIContainer>
```

5.2.7 Добавьте в TextBlock кнопку при помощи InlineUIContainer

5.3 Применение TextBox

5.3.1 Расположить на странице 2 TextBox для логина и пароля.

5.3.2 Указать для них Watermark для отображения подписи для поля ввода

5.3.3 Для поля ввода пароля указать PasswordChar

5.3.4 Добавить еще несколько TextBox для ввода Email и числового кода, указать для них различные значения TextInputOptions.ContentType

5.4 Применение CheckBox

5.4.1 Расположить на странице CheckBox. Протестировать его работу.

5.5 Применение RadioButton

5.5.1 Расположить на странице 4 RadioButton. Для каждой пары RadioButton указать различное значение свойства GroupName.

5.5.2 Протестировать работу RadioButton.

5.6 Применение DatePicker

5.6.1 Расположить на странице DatePicker для выбора даты рождения.

5.6.2 Добавить на страницу метку, выводящую значение возраста человека на основе выбранной даты. Для этого создайте вычисляемое свойство Age, которое вы будете оповещать об изменении основного свойства selectedDate

```
[ObservableProperty]
[NotifyPropertyChangedFor(nameof(Age))]
private DateTime selectedDate;

public int Age => //вычисление возраста
```

5.7 Применение ComboBox

5.7.1 Расположить на странице ComboBox и TextBlock

5.7.2 Создать в ControlsViewModel свойство Languages, содержащее список языков программирования, и поле selectedLanguage, помеченное атрибутом ObservableProperty.

5.7.3 Осуществить привязку свойств к ComboBox для выбора языка, и TextBlock для отображения выбранного. Протестировать их работу

5.8 Применение Slider

5.8.1 Расположить на странице Slider и Rectangle, залитый цветом

5.8.2 В зависимости от значения слайдера необходимо изменять прозрачность Rectangle.

5.9 Применение ToggleSwitch

5.9.1 Расположить на странице ToggleSwitch, установить для него значение OnContent и OffContent, для отображения текста Включено/Выключено

5.10 Валидация свойств

5.10.1 Создайте новую страницу Avalonia (UserControl) – RegistrationView. На странице расположите следующие элементы:

Поле ввода логина
Поле ввода пароля
Поле ввода подтверждения пароля
Поле ввода email
Поле ввода номера телефона

5.10.2 Создайте класс `UserCreditionals`, унаследованный от `ObservableValidator`

5.10.3 Определите в классе поля для всех необходимых свойств привязки:

```
[ObservableProperty]  
[NotifyDataErrorInfo]  
private string name;
```

`NotifyDataErrorInfo` используется для оповещения о изменении ошибок валидации при изменении значения свойства.

5.10.4 Создайте новый `ViewModel – RegistrationViewModel`. Определите в ней свойство для привязки типа `UserCreditionals`.

5.10.5 Используя атрибуты валидации, установите в `UserCreditionals` следующие ограничения:

Логин может содержать только латинские буквы, цифры и нижнее подчеркивание, длина логина – не менее 3 символов и не более 20

Пароль должен быть длиной не менее 8 символов, содержать латинские буквы верхнего и нижнего регистра, спецсимволы

Email должен содержать корректный email, а номер телефона - корректный номер.

Все поля должны быть заполнены.

5.10.6 Проверить работу валидаторов полей ввода.

6 Порядок выполнения работы

6.1 Выполнить все задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Какой элемент управления используется для ввода текстовых данных?

8.2 Какие элементы управления используются для выбора одного или нескольких вариантов из списка?

8.3 От какого класса должны наследоваться классы с поддержкой валидации?