

Работа с элементами управления для задания даты и времени

Во время работы с программными приложениями перед пользователями часто возникает задача выбора даты или времени или задания промежутков дат или времени.

Поэтому важным является изучение основ работы с элементами управления для задания даты и времени.

На данном занятии рассматриваются правила работы с элементами управления `DatePicker` и `TimePicker`.

Перейдем к изучению элемент управления `DatePicker`, который предназначен для выбора даты с помощью специального диалогового окна, отображаемого в пользовательском интерфейсе.

Элемент управления имеет следующие свойства:

свойство `MinimumDate` задает минимальное значение даты (значение по умолчанию равно первому дню 1900 года);

свойство `MaximumDate` задает максимальное значение даты (значение по умолчанию равно последнему дню 2100 года);

свойство `Date` задает значение выбранной даты (по умолчанию имеет значение `DateTime.Today`);

свойство `Format` задает формат даты (стандартный или настраиваемый), который по умолчанию имеет значение `D`;

свойство `TextColor` задает цвет, используемый для показа выбранной даты (по умолчанию `Color.Default`);

свойство `FontAttributes` задает параметры шрифта отображения даты (значение по умолчанию `FontAttributes.None`);

свойство `FontFamily` задает название шрифта, используемого для отображения даты (значение по умолчанию `null`);

свойство `FontSize` задает размер шрифта (значение по умолчанию 1,0);

свойство `CharacterSpacing` задает интервал между знаками текста `DatePicker`.

Значения стандартного формата даты приведены на слайде в презентации

"d"	Короткий шаблон даты	12/14/2020
"D"	Полный шаблон даты	Monday, December 14, 2020

Фрагмент программного кода для задания настраиваемого формата даты приведен на слайде в презентации

```
DateTime thisDate = new DateTime(2020, 12, 02);
```

Когда пользователь выбирает дату с использованием элемента управления DatePicker, запускается событие DateSelected.

Элемент управления DatePicker требует, чтобы значение Date находилось в диапазоне от MinimumDate до MaximumDate включительно. Если значение свойств MinimumDate или MaximumDate заданы так, что значение Date не находится между ними, то DatePicker будет корректировать значение Date.

В программном коде можно инициализировать значения свойств MinimumDate, MaximumDate, а также свойства Date:

```
DatePicker datePicker = new DatePicker  
{  
    MinimumDate = new DateTime(2018, 1, 1),  
    MaximumDate = new DateTime(2020, 12, 31),  
    Date = new DateTime(2020, 12, 02)  
};
```

Если значение DateTime указано в XAML, то средство синтаксического анализа XAML использует метод DateTime.Parse с аргументом CultureInfo.InvariantCulture для преобразования строки в значение DateTime. Даты должны быть указаны в точном формате: двузначное число месяцев, две цифры и годы, состоящие из четырех цифр, разделенных косыми чертами:

```
<DatePicker MinimumDate="01/01/2020"  
    MaximumDate="12/31/2020"  
    Date="12/14/2020" />
```

Для работы с элементом управления DatePicker создадим проект DatePickerDemo для мультиплатформенного приложения Xamarin.Forms с использованием шаблона «Пустой».

Откроем файл MainPage.xaml и наберем программный код, приведенный далее.

Программный код включает два элемента управления DatePicker. Их можно использовать для выбора двух дат. Программа вычисляет количество дней между этими датами. Программа не изменяет значения свойств MinimumDate и MaximumDate, и поэтому две даты должны находиться в диапазоне от 1 января 1900 до 31 декабря 2100.

```
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"  
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"  
    xmlns:local="clr-namespace: DatePickerDemo "  
    x:Class="DatePickerDemo.MainPage"  
    BackgroundColor="Yellow">  
<StackLayout Margin="10">  
    <Label Text="Дни между датами"
```

```
FontSize="30"
FontAttributes="Bold"
TextColor="Black"
Style="{DynamicResource TitleStyle}"
Margin="0, 20"
HorizontalTextAlignment="Center" />
```

```
<Label Text="Начальная дата:"
      FontSize="30"
      FontAttributes="Bold"
      TextColor="Red"/>
```

```
<DatePicker x:Name="startDatePicker"
            Format="D"
            Margin="30, 0, 0, 30"
            FontSize="20"
            FontAttributes="Bold"
            TextColor="Black"
            DateSelected="OnDateSelected" />
```

```
<Label Text="Конечная дата:"
      FontSize="30"
      FontAttributes="Bold"
      TextColor="Blue"/>
```

```
<DatePicker x:Name="endDatePicker"
            MinimumDate="{Binding Source={x:Reference startDatePicker},
                               Path=Date}"
            Format="D"
            Margin="30, 0, 0, 30"
            FontSize="20"
            FontAttributes="Bold"
            TextColor="Black"
            DateSelected="OnDateSelected" />
```

```
<StackLayout Orientation="Horizontal"
            Margin="0, 0, 0, 30">
  <Label Text="Количество дней между датами: "
        VerticalOptions="Center"
        FontSize="20"
        FontAttributes="Bold"
        TextColor="Green"/>
  <Switch x:Name="includeSwitch"
        Toggled="OnSwitchToggled" />
</StackLayout>
```

```
<Label x:Name="resultLabel"
      FontAttributes="Bold"
      HorizontalTextAlignment="Center"
```

```
FontSize="80"
TextColor="Black"/>
```

```
</StackLayout>
</ContentPage>
```

Каждому свойству `Format` элемента управления `DatePicker` присваивается свойство «D» для длинного формата даты. Обратите внимание, что элемент `DatePicker` с именем `endDatePicker` имеет привязку, предназначенную для его свойства `MinimumDate`.

Источником привязки является свойство `Date` элемента управления `DatePicker` с именем `startDatePicker`. Это гарантирует, что конечная дата всегда будет позже или равна начальной дате.

Помимо двух объектов `DatePicker` в программном коде приведен элемент управления `Switch`, предназначенный для включения начальной и конечную дату в общий результат.

Элементы управления `DatePicker` имеют обработчики, прикрепленные к событию `DateSelected`.

К элементу управления `Switch` привязан обработчик события `Toggled`.

Эти обработчики находятся в файле `MainPage.xaml.cs` и активируют вычисление дней между двумя датами.

Откройте файл `MainPage.xaml.cs` и наберите проведенный далее программный код.

В программном коде приведены обработчики `OnDateSelected` и `OnSwitchToggled`, которые обращаются к процедуре `Recalculate`.

Процедура `Recalculate` производит определение диапазона дат с помощью промежутка времени типа `TimeSpan`. Исходными данными для определения промежутка времени являются значения дат `endDatePicker.Date` и `startDatePicker.Date`, получаемые от элементов управления `DatePicker`. Также с помощью метода `IsToggled` элемента управления `Switch` производится учет начальной и конечной даты в диапазоне дат в случае включения элемента управления `Switch`.

```
using System;
using Xamarin.Forms;
using Xamarin.Forms.Xaml;

namespace DatePickerDemo
{
    [XamlCompilation(XamlCompilationOptions.Compile)]
    public partial class MainPage : ContentPage
    {
        public MainPage()
        {
            InitializeComponent();
        }

        void OnDateSelected(object sender, DateChangedEventArgs args)
```

```

    {
        Recalculate();
    }

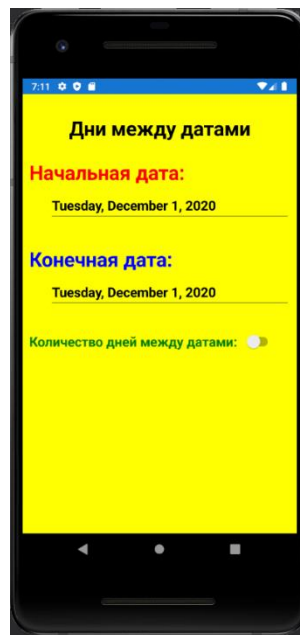
    void OnSwitchToggled(object sender, ToggledEventArgs args)
    {
        Recalculate();
    }

    void Recalculate()
    {
        TimeSpan timeSpan = endDatePicker.Date - startDatePicker.Date +
            (includeSwitch.IsToggled ? TimeSpan.FromDays(1) : TimeSpan.Zero);

        resultLabel.Text = String.Format("{0}", timeSpan.Days);
    }
}

```

Запустим программное приложение для работы с элементами управления DatePicker. При запуске программного приложения в обоих элементах управления DatePicker отображается дата, в которую произведен запуск программного приложения.



При касании любого элемента DatePicker производится выбор даты.



После выбора двух дат в приложении отображается число дней между этими датами (скриншоты для выключенного и включенного элемента управления Switch:



Перейдем к работе с элементом управления TimePicker, который предназначен для выбора времени с использованием специального диалогового окна.

Элемент управления TimePicker имеет следующие свойства:

- свойство Time обозначает выбранное время;

- свойство TextColor обозначает цвет цифр, используемый для показа времени;

свойство `FontAttributes` задает параметры шрифта для отображения времени (значение по умолчанию `FontAtributes.None`);

свойство `FontFamily` задает название шрифта, используемого для отображения времени (значение по умолчанию `null`);

свойство `FontSize` задает размер шрифта для отображения времени;

свойство `CharacterSpacing` задает интервал между знаками текста в `TimePicker`;

свойство `Format` задает формат даты (стандартный или настраиваемый), который по умолчанию имеет значение `t`.

Значения стандартного формата даты приведены на слайде в презентации.

"t"	Короткий шаблон времени	1:45
"T"	Полный шаблон времени	1:45:30

В программном коде Си Шарп можно задать свойству `Time` значение типа `TimeSpan`, который определяет длительность времени, отсчитываемую от полуночи.

```
TimePicker timePicker = new TimePicker  
{  
    Time = new TimeSpan(4, 15, 26)  
};
```

Если значение свойства `Time` указано в XAML, то это значение преобразуется в `TimeSpan` и проверяется, чтобы число миллисекунд было больше или равно 0, а число часов - меньше 24. При этом компоненты времени должны быть разделены двоеточиями.

```
<TimePicker Time="4:15:26" />
```

Также отметим, что в случае изменения значения свойства `Time` запускается событие `PropertyChanged`.

Для работы с элементом управления `TimePicker` создадим проект `TimePickerDemo` для мультиплатформенного программного приложения `Xamarin.Forms` с использованием шаблона «Пустой». Откроем файл `MainPage.xaml` и наберем программный код, приведенный далее в презентации.

Из программного кода видно, что внутри страницы типа `ContentPage` находится макет типа `StackLayout`, внутри которого находятся:

элемент управления `Label` с названием страницы «`TimePicker Demo`»;

элемент управления Entry, в который вводится название события, о котором необходимо напомнить;

элемент управления Label с указанием необходимости выбрать время для выдачи напоминания о проведении мероприятия;

элемент управления TimePicker с настроенным временем и обращением к событию OnTimePickerChanged;

макет типа StackLayout, внутри которого находится элемент управления Label с текстом «Включить таймер» и элемент управления Switch для включения таймера со ссылкой на обработчик OnSwitchToggled.

```
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
xmlns:local="clr-namespace:TimePickerDemo "
x:Class="TimePickerDemo.MainPage"
BackgroundColor="Bisque">
  <StackLayout Margin="20">
    <Label Text="Создание напоминания"
      FontAttributes="bold"
      HorizontalOptions="Center"
      FontSize="40"
      TextColor="Green"/>
    <Entry x:Name="Ent"
      FontSize="30"
      Placeholder="О чем Вам напомнить?"
      FontAttributes="Bold"/>
    <Label Text=" Выберите время"
      FontAttributes="bold"
      TextColor="black"
      FontSize="30"/>
    <TimePicker x:Name="TmPicker"
      FontSize="45"
      Time="14:00:00"
      Format="T"
      PropertyChanged="OnTimePickerChanged" />
    <StackLayout Orientation="Horizontal">
      <Label Text="Включить таймер:"
        FontSize="30"
        FontAttributes="Bold"
        TextColor="black"/>
      <Switch x:Name="Sw"
        HorizontalOptions="EndAndExpand"
        Toggled="OnSwitchToggled"
        ThumbColor="Red" />
    </StackLayout>
  </StackLayout>
</ContentPage>
```


Откройте файл MainPage.xaml.cs и наберите программный код. В программном коде сначала инициализируется работа таймера в соответствии с типом устройства.

Приводится программный код обработчика OnTimerTick для события TimerTick, которое происходит каждую секунду при срабатывании таймера.

Это соответствует срабатыванию обработчика OnTimerTick, который проверяет, включен ли элемент управления Switch, а также сравнивает текущий момент времени с временем напоминания о наступлении события, название которого было введено в элемент управления Entry с именем Ent.

Метод DisplayAlert отображает диалоговое окно, в котором пользователю выдается напоминание о наступлении события.

Приводится программный код обработчика OnTimePickerChanged, на который происходит ссылка из файла MainPage.xaml.

Обработчики OnTimePickerChanged и OnSwitchToggled, на которые происходит ссылка из файла MainPage.xaml, производят обращение к методу SetTriggerTime.

Метод SetTriggerTime определяет время выдачи напоминания о наступлении события на основе значения свойства DateTime.Today и значения TimeSpan.

При этом свойству DateTime.Today соответствует значение типа DateTime, указывающее текущую дату, но с временем полуночи.

Значение TimeSpan соответствует обозначает промежуток времени, начиная с полуночи, получаемый от TimePicker с именем TmPicker.

Если время выдачи напоминания уже прошло сегодня, то предполагается, что напоминание будет выдано завтра (то есть, через один день).

```
using System;
using System.ComponentModel;
using Xamarin.Forms;
namespace TimePickerDemo
{
    public partial class MainPage: ContentPage
    {
        DateTime TT;

        public MainPage()
        {
            InitializeComponent();
            Device.StartTimer(TimeSpan.FromSeconds(1), OnTimerTick);
        }
        bool OnTimerTick()
        {
            if (Sw.IsToggled && DateTime.Now >= TT)
            {
                Sw.IsToggled = false;
                DisplayAlert("Timer Alert",
```

```

        "The '" + Ent.Text + "' timer has elapsed", "OK");
    }
    return true;
}
void OnTimePickerChanged(object sender,
    PropertyChangedEventArgs args)
{
    if (args.PropertyName == "Time")
    {
        SetTriggerTime();
    }
}
void OnSwitchToggled(object sender, ToggledEventArgs args)
{
    SetTriggerTime();
}
void SetTriggerTime()
{
    if (Sw.IsToggled)
    {
        TT = DateTime.Today + TmPicker.Time;
        if (TT < DateTime.Now)
        {
            TT += TimeSpan.FromDays(1);
        }
    }
}
}
}

```

Запустим программное приложение для работы с элементом управления `TimePicker`. В пользовательском интерфейсе отображается элемент `Entry` для ввода названия мероприятия, о наступлении которого необходимо напомнить.

С помощью элемента управления `TimePicker` производится выбор момента времени, в который необходимо выдать напоминание о проведении мероприятия.

Для выбора момента времени элемент управления `TimePicker` предоставляет пользователю специальное диалоговое окно.

После окончания выбора времени напоминания о мероприятии необходимо нажать кнопку `ОК` в диалоговом окне.

В элементе управления `TimePicker` отобразится время мероприятия. Далее необходимо включить таймер с помощью включения элемента управления `Switch`.

При достижении момента времени, в который необходимо напомнить о мероприятии, в пользовательском интерфейсе отображается диалоговое окно с напоминанием. После нажатия на кнопку `ОК` в диалоговом окне происходит отключение таймера

