﻿using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Globalization;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace Ksiegarnia

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

List<Book> lst = new List<Book>();

lst.Add(new Book("Android Studio. Podstawy tworzenia aplikacji (ebook)", "Andrzej Stasiewicz", 47.00M));

lst.Add(new Book("Myślenie matematyczne. Twój nowy sposób pojmowania świata", "Keith Devlin", 29.90M));

lst.Add(new Book("Excel 2013 PL. Programowanie w VBA dla bystrzaków", "John Walkenbach", 49.00M));

lst.Add(new Book("PowerPoint 2007 PL. Pierwsza pomoc", "Roland Zimek", 3.00M));

lst.Add(new Book("Czysty kod. Podręcznik dobrego programisty", "Robert C. Martin", 69.00M));

lst.Add(new Book("Python. Zacznij programować!", "Rob Miles", 89.00M));

lst.Add(new Book("Git. Rozproszony system kontroli wersji", "Włodzimierz Gajda", 54.90M));

lst.Add(new Book("Python. Wprowadzenie. Wydanie IV", "Mark Lutz", 149.00M));

lista.ItemsSource = lst;

ListCollectionView view = (ListCollectionView)CollectionViewSource.GetDefaultView(lista.ItemsSource);

view.Filter = FilterBook;

}

public class ProductByPriceFilter

{

public decimal MinimumPrice

{

get;

set;

}

public ProductByPriceFilter(decimal minimumPrice)

{

MinimumPrice = minimumPrice;

}

public bool FilterItem(Object item)

{

Book product = item as Book;

if (product != null)

{

return (product.Price < MinimumPrice);

}

return false;

}

}

private bool FilterBook(object obj)

{

Book product = (Book)obj;

return (product.Price < 100);

}

private void validationError(object sender, ValidationErrorEventArgs e)

{

if (e.Action == ValidationErrorEventAction.Added)

{

MessageBox.Show(e.Error.ErrorContent.ToString());

}

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

decimal minimumPrice;

if (Decimal.TryParse(txtMinPrice.Text, out minimumPrice))

{

ListCollectionView view = CollectionViewSource.GetDefaultView(lista.ItemsSource) as ListCollectionView;

if (view != null)

{

ProductByPriceFilter filter = new ProductByPriceFilter(minimumPrice);

view.Filter = filter.FilterItem;

}

}

}

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{

ListCollectionView view = CollectionViewSource.GetDefaultView(lista.ItemsSource) as ListCollectionView;

view.Filter = null;

}

private void Button\_Click\_2(object sender, RoutedEventArgs e)

{

ListCollectionView view =(ListCollectionView)CollectionViewSource.GetDefaultView(lista.ItemsSource);

view.SortDescriptions.Add(new SortDescription("Title", ListSortDirection.Ascending));

}

}

public class Book : IDataErrorInfo

{

private decimal price;

public string Title { get; set; }

public string Author { get; set; }

public decimal Price

{

get { return price; }

set { price = value; }

}

public Book(string title, string author, decimal price)

{

Title = title;

Author = author;

Price = price;

}

public string this[string columnName]

{

get

{

if (columnName == "Price")

{

if (price <= 0)

return "Cena musi być większa od 0.";

}

return null;

}

}

public string Error { get { return null; } }

}

[ValueConversion(typeof(decimal), typeof(string))]

public class PriceConverter : IValueConverter

{

public object Convert(object value, Type targetType,

object parameter, CultureInfo culture)

{

decimal price = (decimal)value;

return price.ToString("C", culture);

}

public object ConvertBack(object value, Type targetType,

object parameter, CultureInfo culture)

{

string price = value.ToString();

decimal result;

if (Decimal.TryParse(price, NumberStyles.Any,

culture, out result))

{

return result;

}

return value;

}

}

public class PriceToBackgroundConverter : IValueConverter

{

public decimal MaximumPriceToHighlight { get; set; }

public Brush HighlightBrush { get; set; }

public Brush DefaultBrush { get; set; }

public object Convert(object value, Type targetType, object parameter, System.Globalization.CultureInfo culture)

{

decimal price = (decimal)value;

if (price <= MaximumPriceToHighlight)

return HighlightBrush;

else

return DefaultBrush;

}

public object ConvertBack(object value, Type targetType, object parameter, System.Globalization.CultureInfo culture)

{

throw new NotSupportedException();

}

}

public class PositivePriceRule : ValidationRule

{

private decimal min = 0;

private decimal max = Decimal.MaxValue;

public decimal Min

{

get { return min; }

set { min = value; }

}

public decimal Max

{

get { return max; }

set { max = value; }

}

public override ValidationResult Validate(object value, CultureInfo culture)

{

decimal price = 0;

try

{

if (((string)value).Length > 0)

price = Decimal.Parse((string)value,

NumberStyles.Any, culture);

}

catch

{

return new ValidationResult(false,

"Niedozwolone znaki.");

}

if ((price < Min) || (price > Max))

{

return new ValidationResult(false,

"Cena nie jest w zakresie od" + Min + " do " + Max + ".");

}

else

{

return new ValidationResult(true, null);

}

}

}

}