

Oefening 14.20 - 14.23

```
<Window x:Class="Oef14_21_22_23_Zoekmethoden.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="Oef14.21-23 Duur zoekmethoden" Height="350" Width="525">
    <Grid>
        <TextBlock HorizontalAlignment="Left" Margin="40,40,0,0"
            Text="Sleutel:" VerticalAlignment="Top"/>
        <TextBox HorizontalAlignment="Left" Height="23" Margin="104,39,0,0"
            Name="keyTextBox" Text="Vak999999" VerticalAlignment="Top"
Width="120"/>
        <Button Content="Zoek 1" HorizontalAlignment="Left" Margin="273,43,0,0"
            VerticalAlignment="Top" Width="75"
            Name="zoek1Button" Click="zoek1Button_Click"/>
        <TextBlock HorizontalAlignment="Left" Margin="379,45,0,0"
            Text="(dictionary)" VerticalAlignment="Top"/>
        <Button Content="Zoek 2" HorizontalAlignment="Left" Margin="273,81,0,0"
            VerticalAlignment="Top" Width="75"
            x:Name="zoek2Button" Click="zoek2Button_Click"/>
        <TextBlock HorizontalAlignment="Left" Margin="380,82,0,0"
            Text="(array)" VerticalAlignment="Top"/>
        <Button Content="Zoek 3" HorizontalAlignment="Left" Margin="273,119,0,0"
            VerticalAlignment="Top" Width="75"
            x:Name="zoek3Button" Click="zoek3Button_Click"/>
        <TextBlock HorizontalAlignment="Left" Margin="380,120,0,0"
            Text="(list)" VerticalAlignment="Top"/>
        <TextBox HorizontalAlignment="Left" Height="116" Margin="40,160,0,0"
            Name="resultTextBox" VerticalAlignment="Top" Width="308"/>
    </Grid>
</Window>
```

```
public class Vak
{
    public string Naam { get; set; }
    public string Docent { get; set; }
    public int Uren { get; set; }

    public Vak(string naam, string docent, int uren)
    {
        Naam = naam;
        Docent = docent;
        Uren = uren;
    }
}
```

```
public partial class MainWindow : Window
{
    private const int MaxLength = 1000000;
    private DateTime startTime;
    private Dictionary<string, Vak> vakDictionary;
    private Vak[] vakArray;
    private List<Vak> vakList;

    public MainWindow()
    {
        InitializeComponent();
    }
}
```

```

vakDictionary = new Dictionary<string, Vak>();
vakArray = new Vak[MaxLength];
vakList = new List<Vak>();

for (int i = 0; i < MaxLength; i++)
{
    vakDictionary.Add("Vak" + i, new Vak("Vak" + i, "Docent" + i, 5));
    vakArray[i] = new Vak("Vak" + i, "Docent" + i, 5);
    vakList.Add(new Vak("Vak" + i, "Docent" + i, 5));
}

}

// Dictionary
private void zoek1Button_Click(object sender, RoutedEventArgs e)
{
    StartChronometer();
    string key = keyTextBox.Text;
    if (vakDictionary.ContainsKey(key))
    {
        Vak vak = vakDictionary[key];
        ToonVak(vak);
    }
    else
    {
        MessageBox.Show("Vak: " + key + " niet gevonden.");
    }
    StopChronometer();
}

// Array
private void zoek2Button_Click(object sender, RoutedEventArgs e)
{
    StartChronometer();

    string key = keyTextBox.Text;
    int index = 0;
    bool found = false;

    while (index < vakArray.Length && !found)
    {
        if (key == vakArray[index].Naam)
        {
            ToonVak(vakArray[index]);
            found = true;
        }
        index++;
    }

    StopChronometer();

    if (!found)
    {
        MessageBox.Show("Vak: " + key + " niet gevonden.");
    }
}

// List
private void zoek3Button_Click(object sender, RoutedEventArgs e)
{
    StartChronometer();

```

```

        string key = keyTextBox.Text;
        int index = 0;
        bool found = false;

        while (index < vakList.Count && !found)
        {
            if (key == vakList[index].Naam)
            {
                ToonVak(vakList[index]);
                found = true;
            }
            index++;
        }

        StopChronometer();

        if (!found)
        {
            MessageBox.Show("Vak: " + key + " niet gevonden.");
        }
    }

    private void StartChronometer()
    {
        startTime = DateTime.Now;
    }

    private void StopChronometer()
    {
        TimeSpan timeSpan = DateTime.Now - startTime;
        string line = String.Format("Duur: {0} sec en {1} msec.",
                                    timeSpan.Seconds, timeSpan.Milliseconds);

        MessageBox.Show(line);
    }

    private void ToonVak(Vak vak)
    {
        resultTextBox.Clear();

        resultTextBox.AppendText("Vakgegevens voor " + vak.Naam);
        resultTextBox.AppendText(Environment.NewLine);
        resultTextBox.AppendText(Environment.NewLine);
        resultTextBox.AppendText("Docent: " + vak.Docent);
        resultTextBox.AppendText(Environment.NewLine);
        resultTextBox.AppendText("Uren: " + vak.Uren);
    }
}

```