Versuchen Sie immer, Ihren Code zu kommentieren!

Aufgabe 1 Windows Forms

Verwenden Sie die folgende Methode für einen asynchronen Aufruf mit Task.Run und erwarten Sie das Ergebnis von ComputePi().

```
double ComputePi()
{
    double sum = 0.0;
    const double step = 1e-9;

    for(int i = 0; i < 1_000_000_000; i++)
    {
        double x = (i + 0.5) * step;
        sum += 4.0 / (1.0 + x * x);
    }

    return sum * step;
}
```

Starten Sie die Berechnung durch einen Buttonklick. Der Click-Ereignishandler soll anschließend den Button deaktivieren. Warten Sie auf das Ergebnis der Berechnung und setzen Sie das Ergebnis der Berechnung in ein Label ein. Aktivieren Sie den Button anschließend wieder.

Um aus einem Task auf die Elemente einer Windows-Forms-Anwendung threadsicher zugreifen zu können, verwenden Sie die Methode Invoke der Klasse Control.

Ein Beispiel (Code im Task):

```
Invoke(() =>{
    // Text ändern
});
```

- Zeigen Sie während der Berechnung die Sanduhr an.
    - Mittels this.Cursor = Cursors.WaitCursor wird eine Sanduhr angezeigt. Cursors.Default ist der Standard Mauszeiger.

- Fügen Sie einen Button zum Abbruch der Berechnung ein.

- Verwenden Sie ein Progressbar-Control zum Anzeigen des Fortschritts der Berechnung.

Aufgabe 2 Windows Forms

Erstellen Sie eine kleine WindowsForms-Anwendung, welche die sog. Favicons (https://de.wikipedia.org/wiki/Favicon) von einigen Webseiten herunterlädt und anzeigt.

Verwenden Sie dazu die Methode GetByteArrayAsync aus der Klasse HttpClient.

Stellen Sie die Icons z.B. in einem FlowLayoutPanel dar. Jedes Icon könnte z.B. mittels einer PictureBox dargestellt werden.

Auf die folgende Art und Weise kann aus den Downloaddaten der Klasse HttpClient eine PictureBox erstellt werden:

```
PictureBox pictureBox = new PictureBox();

HttpClient httpClient = new HttpClient();
var bytes = await httpClient.GetByteArrayAsync("url");

// Der Stream darf nicht geschlossen werden, solange das Bild verwendet wird!
MemoryStream ms = new MemoryStream(bytes);

pictureBox.Image = Image.FromStream(ms);
```

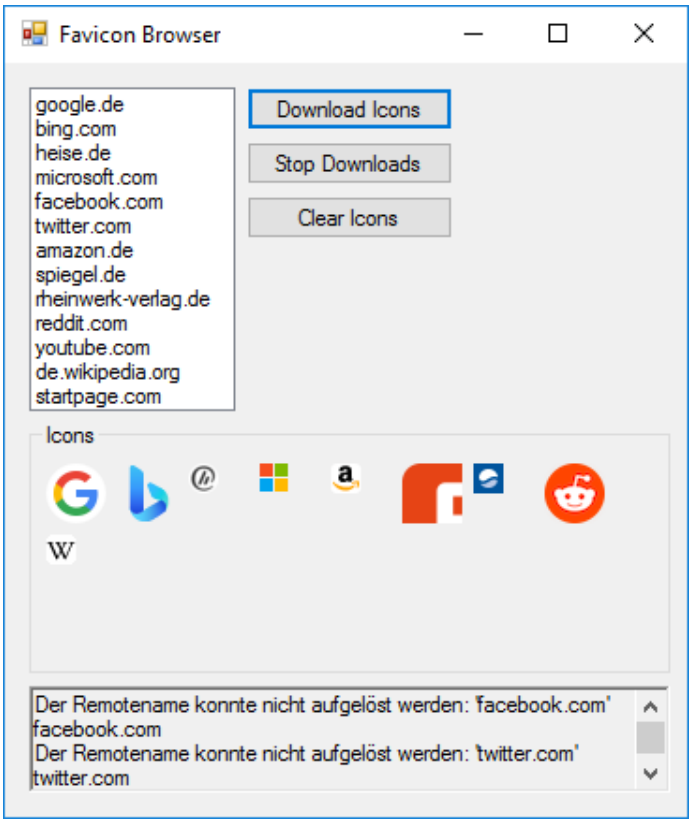
Die PictureBox Objekte können anschließend in das Panel eingefügt werden. Ein geladenes Icon soll angezeigt werden, sobald es verfügbar ist!

Ihre Anwendung sollte von den folgenden Domains versuchen das Icon zu laden und mögliche Fehler abfangen:
"google.de", "bing.com", "heise.de", "microsoft.com", "facebook.com", "twitter.com", "amazon.de", "spiegel.de", "rheinwerk-verlag.de", "reddit.com", "youtube.de", "de.wikipedia.org"

Mögliche Zusätze zur Aufgabe:

- Zeigen Sie während des Downloads die Sanduhr an. Mittels this.Cursor = Cursors.WaitCursor wird eine Sanduhr angezeigt. Cursors.Default ist der Standard Mauszeiger.

- Fügen Sie einen Button zum Abbruch aller Downloads ein.


Favicon Browser
google.de
bing.com
heise.de
microsoft.com
facebook.com
twitter.com
amazon.de
spiegel.de
rheinwerk-verlag.de
reddit.com
youtube.com
de.wikipedia.org
startpage.com
Download Icons
Stop Downloads
Clear Icons
Icons
Der Remotename konnte nicht aufgelöst werden: 'facebook.com'
facebook.com
Der Remotename konnte nicht aufgelöst werden: 'twitter.com'
twitter.com