

Arbeitsblatt: DNET1

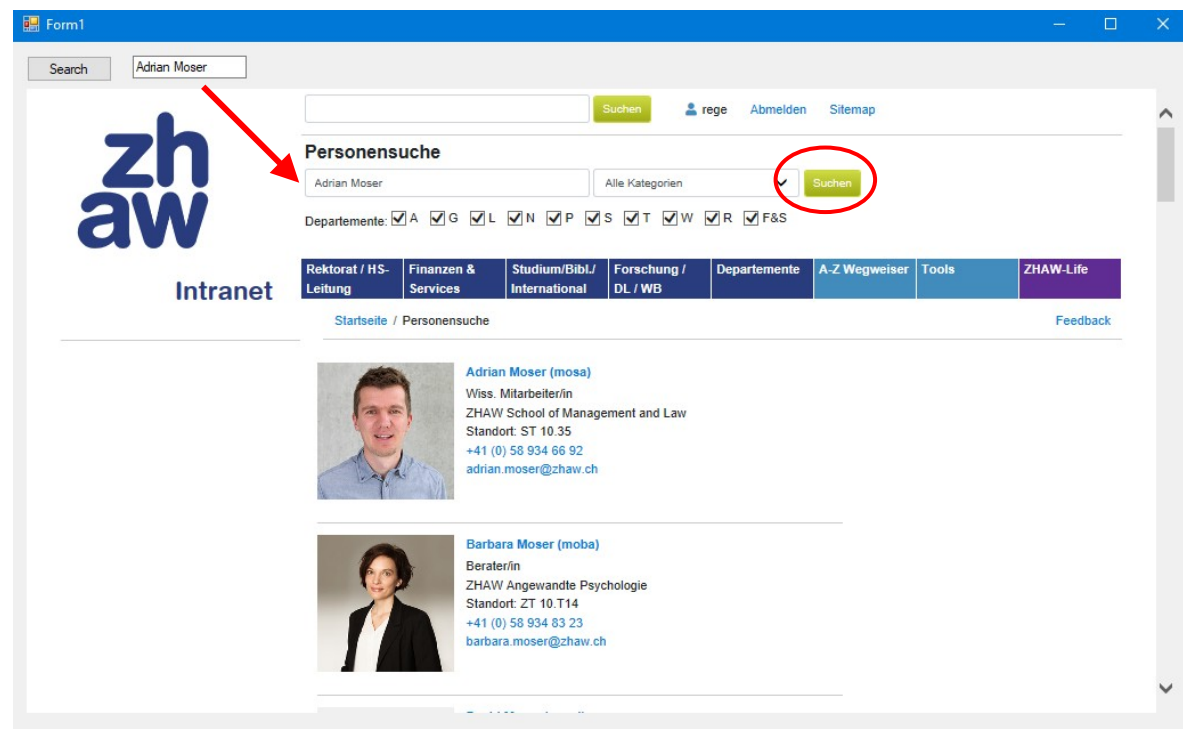
Name: Kurznamen:

Fortsetzung NextChange

Aufgabe 1 – Browser Einbindung und Fernsteuerung

Es gibt auch die Möglichkeit, die Adressliste des Intranets zu verwenden. Dabei stellt sich jedoch das Problem des entfernten Zugriffs. Eine einfache Möglichkeit ist die Darstellung in einem Browser Fenster der URL:

<https://intra.zhaw.ch/personensuche/>



Hinweis:

- Anchors setzen, damit das Browser Control frei bewegbar wird (Docking zu none setzen).

Nach dem Einbinden erfolgt der Aufruf einer Seite einfach mittels:

```
webBrowser1.Navigate ("https://intra.zhaw.ch/personensuche/");
```

Sobald die Seite fertig geladen wurde, wird ein Event ausgelöst der abgefangen werden kann.

```
webBrowser1.DocumentCompleted += new  
WebBrowserDocumentCompletedEventHandler(webBrowser1_DocumentCompleted);
```

Die Bearbeitungsmethode folgende Signatur besitzen:

```
private void webBrowser1_DocumentCompleted(Object sender,  
WebBrowserDocumentCompletedEventArgs e)
```

Da mehrere Seiteladeevents erfolgen können muss mittels:

`e.Url.ToString().StartsWith("https://intra.zhaw.ch/personensuche/")` noch überprüft werden ob wir uns wirklich auf Intranet Seite befinden.

Danach soll der Name der zu suchen Person ins Formular eingefüllt werden. Das entsprechende Form sieht wie folgt aus.

```
<form name="person-search" action="personensuche/" method="post">  
  <div class="form-group">  
    <input class="form-control person-search-query"  
name="tx_zhawperson_zhawperson[personSearchQuery]" value="" type="text">  
    <select class="form-control person-search-category"  
name="tx_zhawperson_zhawperson[personSearchCategory]">  
      <option value="" selected="selected">Alle Kategorien</option>  
      <option value="1000">Professoren/innen</option>  
      <option value="2000">Dozierende</option>  
    </select>  
    <input class="btn btn-primary" value="Suchen" type="submit">  
  </div>
```

Hinweise:

- Setzen von Feldern

Die Textfelder des Passwort-Formulars greifen wir folgendermassen zu:

```
private HtmlElement getElementByClassName(string name) {  
    foreach (HtmlElement elem in webBrowser1.Document.All){  
        if (elem.GetAttribute("className") == name) return elem;  
    }  
    return null;  
}
```

```
HtmlElement e1 = getElementByClassName("form-control person-search-query");  
e1.SetAttribute("value", "<StudentenName>");
```

- Drücken des Knopfes

Das Formular wird durch einfaches aktivieren des Submit-Knopfes abgeschickt:

```
HtmlElement e2 = getElementByClassName("btn btn-primary");  
e2.invokeMember("click")
```

Abgabe:

Praktikum: DT8.1

Datei: Form1.cs

Aufgabe 2 – Low Level Programmierung

Für einen Kalender ist die exakte Uhrzeit wichtig. Leider sind die Uhren der meisten Computer ziemlich ungenau und sie sollte deshalb periodisch überprüft und allenfalls korrigiert werden. Es gibt im Internet für diesen Zweck sogenannte NIST Internet Time Server, die auf Anfrage die exakte Zeit liefern. Eine Zusammenstellung der verfügbaren Time Server findet man unter <http://tf.nist.gov/tf-cgi/servers.cgi#>

Leider kann in .NET nicht direkt die Uhr der eigenen Maschine gesetzt werden. Es gibt jedoch in der Kernel32.dll einen entsprechenden Aufruf; siehe Zusatzdokument (pdf)

Schreiben Sie eine Methode, welche die Uhr mit der NIST Zeit synchronisiert.

Hinweis: es muss folgendes `struct` mit den zu C++ analogen Datentypen als **ref** dem Aufruf übergeben werden.

```
public struct SystemTime                                // win32 struct
{
    public short wYear;
    public short wMonth;
    public short wDayOfWeek;
    public short wDay;
    public short wHour;
    public short wMinute;
    public short wSecond;
    public short wMilliseconds;
};
```

Hinweise

- Sie müssen den Aufruf unter Admin Rechten ausführen; Rechts-Klick und runas
- Oder mittels Anleitung <http://www.codeproject.com/KB/cs/zetaimpersonator.aspx> (Quelle finden Sie auf der .NET Seite).

Abgabe:

Praktikum: DT8.2

Datei: SyncTime.cs