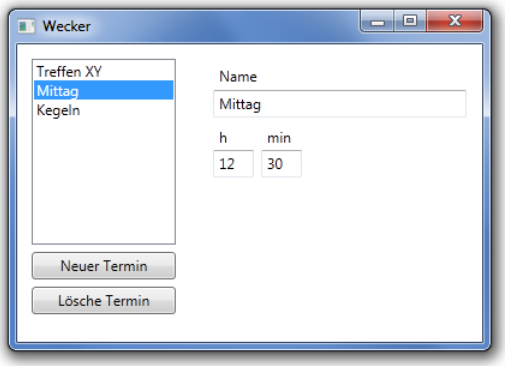
Praktikum 29./30. März 2011

Jörn Loviscach

Versionsstand: 29. März 2011, 00:52

In diesem Praktikum soll ein Wecker mit mehreren Weckzeiten programmiert werden.

Die grafische Oberfläche enthält eine `ListBox`, in der die Namen der Termine aufgelistet sind. Man kann mit einem `Button` neue Termine anlegen und mit einem weiteren `Button` einen in der Liste angewählten Termin löschen. Die Daten des jeweiligen Termins – Name, Stunde und Minute – stehen in drei Elementen vom Typ `TextBox`.



Erzeugen Sie ein neues Projekt (C#, WPF) und fügen Sie per Mausklick rechts im „Projekmappen-Explorer“ eine neue Klasse namens `AlarmTime` hinzu. Jede Instanz davon soll eine Weckzeit darstellen. Geben Sie dieser Klasse öffentliche Variablen für Stunde, und Minute der Weckzeit, beide vom Type `int`, und für den Namen, vom Typ `string`. (Öffentliche Variablen sind unsauber; später machen wir das anders.) Geben Sie dieser Klasse außerdem eine öffentliche Funktion `ToString()`, welche den Namen zurückliefert. Diese Funktion muss mit dem Zusatz `override` versehen werden, weil sie eine geerbte Funktion überschreibt (Details demnächst in der Vorlesung).

Der Hinzufügen-Button legt ein neue Instanz von `AlarmTime` an, füllt diese mit Daten aus den `TextBox`en und hängt sie mit `... .Items.Add(...)` an das Ende der `ListBox`. Die Stunden- und Minutenangaben aus den Textboxen lassen sich mit der Funktion `int.Parse` aus Zeichenketten zu Zahlen verwandeln.

Der Löschen-Button holt mit `... .SelectedIndex` die Nummer der aktuell gewählten Zeile der `ListBox` bzw. -1, wenn nichts ausgewählt ist. Mit `... .Items.RemoveAt(...)` lässt sich dann ein Eintrag entfernen.

Klickt man in der `ListBox` auf einen vorhandenen Eintrag, sollen die drei `TextBox`en auf dessen Daten gesetzt werden. Dazu lässt sich das `SelectionChanged`-Ereignis der `ListBox` benutzen. Mit `(AlarmTime)... .SelectedItem` erhält man die aktuell ausgewählte `AlarmTime` oder, wenn nichts ausgewählt ist, den Wert `null`.

Damit der Wecker auch wirklich weckt, soll eine Instanz der Klasse `System.Windows.Threading.DispatcherTimer` mehrmals pro Minute eine Funktion aufrufen, die alle Einträge der `ListBox` durchgeht und überprüft, ob irgendeine der Weckzeiten gleich der aktuellen Stunde und Minute ist. Wenn ja, soll der Fensterhintergrund für die Dauer dieser Minute auf rot geschaltet werden. Die aktuelle Zeit erhält man mit `DateTime.Now`. Mit `... .Items.Count` erfährt man die Anzahl der Einträge in der `ListBox`. Mit `(AlarmTime)... .Items[...]` kann man auf einen davon zugreifen.

Ist eine Weckzeit in der `ListBox` angewählt und ändert man dann den Namen oder die Uhrzeit in den `TextBox`en, sollen die Änderungen in diese Weckzeit übernommen werden. Benutzen Sie dazu das Ereignis `TextChanged` jeder der drei `ListBox`. Dafür, dass die `ListBox` Änderungen des Namens übernimmt, ist ein `... .Items.Refresh()` nötig.

Der Knopf zum Löschen einer Weckzeit soll deaktiviert sein, wenn kein Eintrag in der Liste ausgewählt ist. Setzen Sie dafür die Eigenschaft `isEnabled` dieses Knopfs zum passenden Zeitpunkt auf `true` oder `false`.