

Coding Dojo



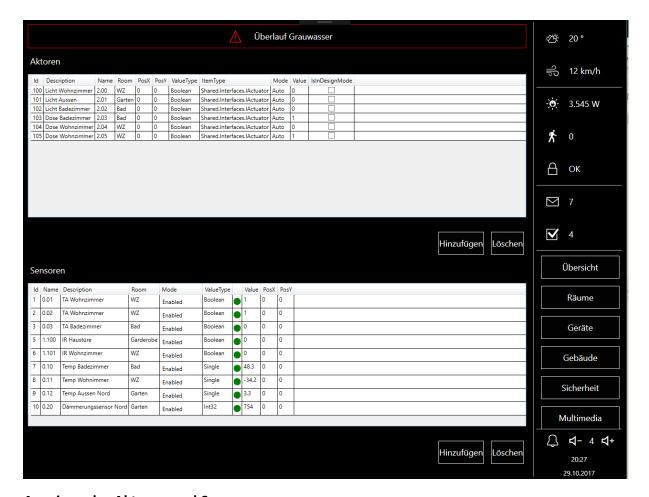
3

Beschreibung:

IN dieser Aufgabenstellung entwickeln wir Coding Dojo2 weiter. Sie können dabei entweder auf ihrer Version des Dojos weiterarbeiten oder die Referenzlösung verwenden.

Folgende Punkte werden ergänzt:

- Anzeigen der Aktoren und Sensoren in einem Datagrid.
- Implementieren der Uhrzeit & Datum
- Vorbereiten von je zwei Buttons für Aktoren/Sensoren (Hinzufügen, Löschen) aber ohne Funktion



Anzeigen der Aktoren und Sensoren:

Dazu wird im Repository des Dojos in der Referenzlösung ein Projekt zur Verfügung gestellt das die Daten liefert. Sie müssen also die Referenzlösung öffnen und einmal Kompilieren. Dann können sie die Shared.dll in Ihr Projekt integrieren.

Die Shared Assembly bietet die Strukturen für die Aktoren und Sensoren und stellen das Model dar. Dieses sollte in der WPF Applikation durch ein ViewModel abstrahiert werden. (In

der Referenzlösung geschieht dies durch das ItemVm aus dem CodingDojo3.ViewModel Namespace).

Damit sie auch Daten anzeigen können und die Wertänderungen Simuliert werden können sie die Simulator Klasse (aus dem CodingDojo3. DataSimulation Namespace) in Ihr Projekt übernehmen. Sobald sie in Ihrem MainViewModel eine Instanz des Simulators erstellen legt dieser Demo Daten an und startet eine Simulation die die Werte verändert. Dabei werden nur jene Werte geändert die als Mode entweder Auto (Aktoren) oder Enabled (Sensoren) haben. Die Änderungen sollten in Ihrer GUI angezeigt werden.

In der Referenzlösung wurden die Spalten eines Datagrids automatisch erstellt und im zweiten (Sensoren) manuell definiert – da stehen dann mehrere Möglichkeiten zu Verfügung (Converter, nur ausgewählte Properties anzeigen, Unterschiedliche Darstellung zwischen Edit und normaler Darstellung einer Zelle – siehe Mode).

e	nsore	n									
ld	Name	Description	Room		Mode	ValueTy	pe		Value	PosX	PosY
1	0.01	TA Wohnzimmer	WZ		Enabled	Boolear		•	0	0	0
2	0.02	TA Wohnzimmer	WZ		Enabled v	Boolear			0	0	0
3	0.03	TA Badezimmer	Bad		Enabled	Boolear	Г	•	1	0	0
5	1.100	IR Haustüre	Gardero	be	Disabled Auto	Boolear	Г	•	0	0	0
6	1.101	IR Wohnzimmer	WZ		Manual	Boolear	Г	•	0	0	0
7	0.10	Temp Badezimmer	Bad		Enabled	Single		•	0	0	0
8	0.11	Temp Wohnimmer	WZ Garten		Enabled	Single	ngle		4.2	0	0
9	0.12	Temp Aussen Nord			Enabled	Single		•	21.7	0	0
10	0.20	Dämmerungssensor Nord	Garten		Enabled	Int32		•	444	0	0

Je nach Wissensstand können sie hier auch die automatische Befüllung der Spalten wählen. Zur Realisierung des MVVM Patterns wurde MVVM Light verwendet. Es ist jedoch eine gute Übung die Interfaces und Klassen selbst zu implementieren (INotifyPropertyChanged und ICommand).

Implementieren der Uhrzeit & Datum

Das Binding muss die GUI über Änderungen informieren (PropertyChanged). Und mittels DispatcherTimer werden die Werte alle X Sekunden aktualisiert.

Das Datum kann mittels DateTime.Now ermittelt werden.

Vorbereiten von je zwei Buttons für Aktoren/Sensoren

Die Buttons werden lediglich positioniert, besitzen aber noch keine Funktion.