

Meet the Teacher

Lehrziele

- Interfaces **Comparable** und **Comparer**
- Generic Collections: **List**, **Dictionary**
- Kommandozeilenargumente in Visual Studio

Aufgabenstellung

Für die HTL Leonding Homepage bereitet das Sekretariat jedes Jahr nach Schulbeginn eine Liste mit den Sprechstunden aller Lehrer für das Homepage-Team vor. Weiters gibt es Lehrkräfte, die eine Visitenkarte mit Foto aktivieren wollen, und andere, für die nur die Sprechstunde angezeigt werden soll.

Das Endergebnis sieht so aus:

- <https://www.htl-leonding.at/kontakt/lehrer-innen/>

Ihre Aufgabe ist es, ein C# Konsolenprogramm zu schreiben, das aus zwei verschiedenen CSV-Dateien eine HTML-Datei mit der Übersicht über alle Lehrkräfte inkl. Sprechstunden und – falls es eine Visitenkarte gibt – mit einem Hyperlink auf die Visitenkarte erzeugt.

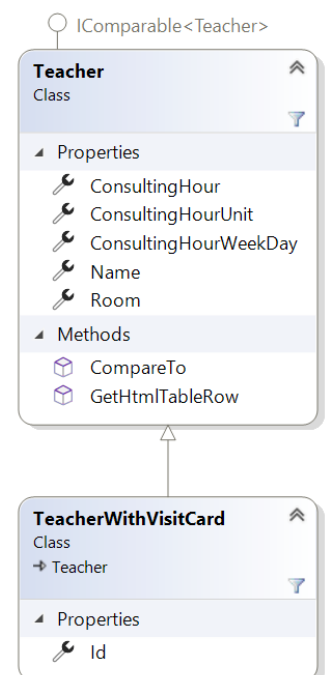
Klassendesign

Teacher

- Wie in dem Klassendiagramm rechts abgebildet, hat eine Lehrkraft immer einen **Namen** (Nachname und Vorname, diese Kombination ist eindeutig). Es ist daher ein **Konstruktor** vorzusehen, der ein Teacher-Objekt mit zumindest einem Namen erzeugt.
- Falls die Lehrkraft aktiv an der Schule unterrichtet (also nicht in Karenz ist oder in einer Austauschschule im Ausland arbeitet) und es sich um die Stammschule handelt (keine Mehrfachverwendung), so sind neben dem Namen auch folgende Sprechstundendaten angegeben:
 - o der Wochentag der Sprechstunde (**ConsultingHourWeekDay**),
 - o die Uhrzeit der Sprechstunde (**ConsultingHour**)
 - o die Einheit dieser Sprechstunde im Stundenplan (**ConsultingHourUnit**)
 - o und der Raum, in der die Sprechstunde stattfindet (**Room**)

TeacherWithVisitCard

- Eine Lehrkraft mit Visitenkarte ist ein **TeacherWithVisitCard**, der zusätzlich zum Namen eines **Teachers** verpflichtend eine Integer-Id für den Hyperlink auf die Visitenkarte enthält.
- Dementsprechend ist für diese Klasse ein Konstruktor vorzusehen, der **Name** und **Id** verlangt.



CSV-Import

Die beiden CSV-Dateien stellen einmal nur die Lehrkräfte mit Visitenkarten und einmal alle Lehrkräfte insgesamt dar:

- **2122-teacher-with-details.csv**: Name und Details-Id von Lehrkräften mit Visitenkarte
- **2122-teachers.csv**: Name und Sprechstunden-Daten aller Lehrkräfte.

Gehen Sie beim Import dieser beiden Dateien wie folgt vor:

1. Erstellen Sie aus den Lehrkräften mit Visitenkarte **TeacherWithVisitCard**-Objekte und fügen Sie diese in ein **Dictionary<string,Teacher>** ein, wobei der Teacher-Name als **Key** dienen soll.
2. Lesen Sie dann die zweite CSV-Datei ein und sehen Sie für jeden Teacher-Namen nach, ob das Dictionary bereits ein Objekt mit diesem Namen enthält, andernfalls ist ein neues Teacher-Objekt zu erstellen und ins Dictionary einzufügen. Übernehmen Sie schließlich alle Sprechstundendaten in das jeweilige Teacher-Objekt.

Nach dem Import sollten für das Schuljahr 2021/22 insgesamt 140 Teacher-Objekte (ob mit oder ohne Visitenkarten-Id) im Dictionary enthalten sein – noch unsortiert.

Generieren der HTML-Table

Für das Erstellen der HTML-Tabelle benötigt man die Values aus dem Dictionary, die alle Teacher-Objekte enthalten. Mit der Methode `ToList()` kann man die Values in eine generische, sortierbare Liste umwandeln:

```
string header =
@"
<table id="tabelle">

<tr>
<th align="center">Name</th>
<th align="center">Tag</th>
<th align="center">Einheit</th>
<th align="center">Zeit</th>
<th align="center">Raum</th>
</tr>

";

StringBuilder sb = new StringBuilder(header);

List<Teacher> teachers = _allTeachers.Values.ToList();
```

Abhängig von der gewählten Sortierreihenfolge muss diese Liste schließlich sortiert werden, bevor man mithilfe eines **StringBuilders** den gesamten HTML-Text der Tabelle erstellen kann. Dieser Text besteht zumindest aus einem `Html-Table-Element` und einer `Html-Table-Kopfzeile`.

Nun kann man mithilfe der `StringBuilder`-Methode **Append** für jedes Teacher-Objekt eine `Table-Row (tr)` anfügen. Dazu ist bereits eine **Teacher**-Methode namens **GetHtmlTableRow()** vorbereitet, die zu implementieren ist.

Hinweise:

- Verwenden Sie für die Implementierung von **GetHtmlTableRow** Delta-Programmierung, sodaß sich die beiden Klassen **Teacher** und **TeacherWithVisitCard** ausschließlich in der Erstellung der `Table-Data-`

Spalte für **Name** unterscheiden (mit oder ohne Hyperlink auf die Visitenkarte!)

- Mit ***sb.ToString()*** kann das Ergebnis des ***StringBuilders*** abgerufen werden. Vergessen Sie nicht auf das Schließen des Html-Table-Elements!

Frei wählbare Sortierung

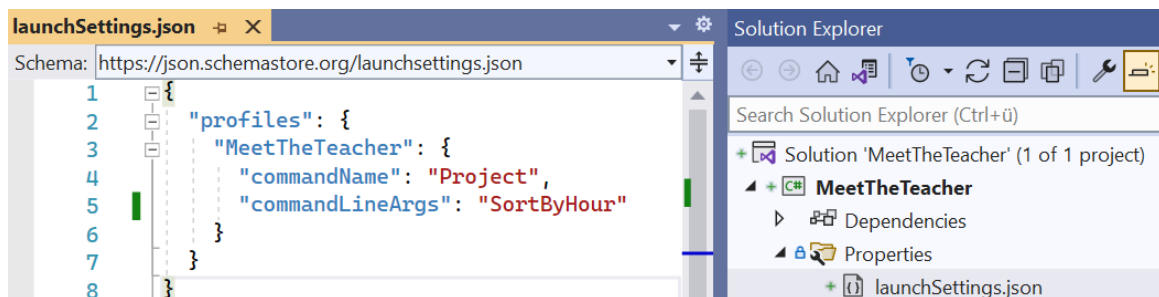
Über einen Kommandozeilenparameter soll auswählbar sein, ob die HTML-Tabelle

- Nach Namen (**SortByName**, default)
- Oder nach Zeitpunkt der Sprechstunde aufsteigend (**SortByHour**)

Sortiert sein soll. Dazu soll das Programm mit einem optionalen Kommandozeilenargument aufrufbar sein. Beispiele siehe rechts:

Setzen von Command Line Arguments in Visual Studio

Visual Studio erlaubt die Konfiguration von Kommandozeilenparametern in der json-Datei „launchSettings.json“ im Properties-Ordner des Projekts:



Implementierung der Sortierung (siehe Muster-Outputs!):

- **Comparable<Teacher>**: Dieses Interface soll verwendet werden, um die Default-Sortierung SortByName umzusetzen
- **Comparer<Teacher>**: Dieses Interface soll verwendet werden, um die alternative Sortierung nach Sprechstunde umzusetzen. Dabei gelten folgende Regeln:
 - o Hat ein Teacher gar keine Sprechstunde, so wird er/sie nach hinten gereiht. Bei mehreren Teachers ohne Sprechstunde wird dabei nach Namen sortiert.
 - o Ansonsten wird zuerst nach Wochentag, dann nach Einheit, und bei gleicher Sprechstunde nach Teacher-Name sortiert

```
T:\>MeetTheTeacher.exe
Welcome @ HTL Leo HTML table generator
=====
No sort order requested, sorting by name is default.
In total 140 teachers created from csv files.
File Output.html has been generated.
```

```
T:\>MeetTheTeacher.exe bla
Welcome @ HTL Leo HTML table generator
=====
You requested sort order bla!
Sorry, sort order bla unknown, sorting by name is default.
In total 140 teachers created from csv files.
File Output.html has been generated.
```

```
T:\>MeetTheTeacher.exe SortByHour
Welcome @ HTL Leo HTML table generator
=====
You requested sort order SortByHour!
Sort order 'SortByHour' has been set successfully.
In total 140 teachers created from csv files.
File Output.html has been generated.
```

