

Infos zur Klausur in Fachtheorie für .NET KandidatInnen

Bitte Vorder- und Rückseite vorher durchlesen und erst dann mit der Bearbeitung beginnen!

Pfade

Die Solution befindet sich im Ordner *P:/SPG_Fachtheorie/SPG_Fachtheorie.sln*. Sie liegt direkt am Netzlaufwerk, d. h. es ist kein Sichern der Arbeit erforderlich.

Damit das Kompilieren schneller geht, ist der Ausgabepfad der Projekte auf *C:/Scratch/(Projekt)* umgestellt. Das ist zum Auffinden der generierten Datenbank wichtig.

Nullable Reference Types

Das Feature *nullable reference types* wurde in dieser Solution deaktiviert. Es können daher Properties mit einer *public set* Methode ohne Warnung verwendet werden.

SQLite Studio

Zur Betrachtung der Datenbank in *C:/Scratch/Aufgabe1.Test/Debug/net6.0* bzw. *C:/Scratch/Aufgabe2.Test/Debug/net6.0* steht die Software *SQLite Studio* zur Verfügung. Die exe Datei befindet sich in *C:/Scratch/SPG_Fachtheorie/SQLiteStudio/SQLiteStudio.exe*. Mittels *Database - Add a Database* kann die erzeugte Datenbank (*Application.db* bzw. *Grades.db*) geöffnet werden.

Auswählen und Starten der Webapplikation (Visual Studio)

In der Solution gibt es 2 Projekte: *Aufgabe3Mvc* und *Aufgabe3RazorPages*. Du kannst wählen, ob du die Applikation mit MVC oder RazorPages umsetzen willst. Entferne das nicht benötigte Projekt aus der Solution und lege dein verwendetes Projekt als Startup Projekt fest (Rechtsklick auf das Projekt und *Set as Startup Project*).

Beim ersten Start erscheint die Frage *Would you like to trust the ASP.NET Core SSL Certificate?* Wähle *Yes* und *Don't ask me again*. Bestätige den nachfolgenden Dialog zur Zertifikatsinstallation.

Verwenden von JetBrains Rider

Im Labor B4.12 steht im Pfad *C:/Scratch/JetBrains.Rider-2022.1.1.win/bin/rider64.exe* JetBrains Rider zur Verfügung. Starte Rider über diese EXE Datei und öffne die Solution von *P:/SPG_Fachtheorie/SPG_Fachtheorie.sln*.

Falls Probleme mit Rider auftreten, steht auch Visual Studio 2022 auf jedem PC zur Verfügung. Bitte die Solution mit VS2022 öffnen und dort weiterarbeiten.

Anpassen der Seed Methode

Die Seed Methode generiert eine realistische Notenverteilung (10% Nicht genügend). Da zum Testen mehrere Nicht genügend erforderlich sind, ändere in der Datei *SPG_Fachtheorie.Aufgabe2/GradeContext.cs* die Zeile 110 auf den Wert

```
var markWeights = new float[] { 0.1f, 0.1f, 0.1f, 0.2f, 0.5f };
```

Dadurch werden 50% Nicht genügend generiert. Füge außerdem vor Zeile 115 am Beginn der 2. foreach Schleife eine Abfrage ein, dass nur Noten zugewiesen werden, wenn der Gegenstand auch der Klasse des Schülers entspricht. Sonst bekommt z. B. auch ein Schüler der 4AHIF eine Note aus DBI in der 3CHIF.

```
// ...
// foreach(var l in lessons)
// {
//     if (l.ClassId != s.ClassId) { continue; } // Dieses if einfügen.
//     grades.Add(new Grade
//     {
//         ...
//     });
// }
```

Hinweis: Die GUID Werte ändern sich bei jedem Seed Vorgang. Dies ist ggf. bei Unittests und bei Verlinkungen in der Webapplikation zu berücksichtigen.

Navigations

Für manche Abfragen ist es hilfreich, die 1:n Beziehung beidseitig abzubilden. Dafür dürfen die Modelklassen in *SPG_Fachtheorie.Aufgabe2/Model* geändert werden. So kann z. B. in der Klasse *Student* eine Collection von Grades eingefügt werden, um leichter auf die Noten des Schülers zugreifen zu können. EF Core befüllt die Collection dann mit den Noten des Schülers. Beachte jedoch, dass LazyLoading nicht aktiviert ist.

```
public class Student
{
    // ...
    public List<Grade> Grades { get; }
}
```

Unterlagen

Auf *R:/Exams/FT_HIF* befinden sich die Beispielprojekte auf Basis von RazorPages bzw. MVC.

Viel Erfolg!