

Lab 9 – Dynamic Language Runtime

Aufgabe 1 – Dynamic Objects

Schreiben Sie im Namespace `Fhnw.Ecnf.RoutePlanner.RoutePlannerLib.Dynamic` die dynamische Klasse `World`, die mit beliebigen Städtenamen als Methode aufgerufen werden kann.

Dem Konstruktor der Klasse `World` soll hierzu ein `Cities`-Objekt übergeben werden, das die Liste der bekannten Städte enthält

Falls die Stadt mit dem verwendeten Methodennamen existiert, soll das entsprechende `City`-Objekt zurückgegeben werden; sonst soll der String `"The city "<cityname>" does not exist!"` zurückgegeben werden.

Ein Beispiel:

```
var cities = new Cities();
Cities.ReadCities(filename);

var w = new World(cities);
dynamic city = w.Windisch(); // city exists, should return Windisch
Console.WriteLine(city.Name); // print City Name
city = w.Bern(); // city exists, should return Bern
city = w.Entenhausen(); // string: "The city "Entenhausen" does not exist!"
```

Hinweis: Für diese Aufgabe müssen Sie die Assembly `Microsoft.CSharp.dll` in ihr Projekt `RoutePlannerLib` und `Test-Projekt` einbinden.

Aufgabe 2 – Tests

Testen Sie Ihre Implementierung mit den Unit-Tests in den Files `«*Test_Lab07.cs»` im `«caseStudyFiles»` Ordner.

Testfragen

- Wann ist der Einsatz von `dynamic` sinnvoll? Was sind die Risiken?
- Welchen Datentyp hat `dynamic` zur Laufzeit? Zur compile-time?
- Was unterscheidet `dynamic` und `var`?
- Was ist ein `ExpandoObject`?
- Was versteht man unter *call-site caching*?
- Wozu dient `DynamicObject`?