

Modul 008: Reflection – LAB

Ziel:

In dieser LAB verwenden wir Reflections die als Grundlagen für das Plugin-Pattern darstellen.

Das zweite Lernziel wird, das Erstellen und Einbinden weiterer Plugins sein.

Das fertige Programm soll alle vier Grundrechenarten auf Basis Plugin-Pattern darstellen.

Geschätzte Zeit: 20-25 Minuten

Systemvoraussetzungen: .Net 6.0

Tools: Visual Studio 2022

Autor: Kevin Winter

Letzte Änderung: 15.06.22

1 Vorbereitung

- 1.1 Öffnen Sie Visual Studio 2022 und erstellen Sie eine Solution mit dem Namen *CalculatorPluginSample*.

2 Aufgabenstellung 1

Schritt 1: Calculator.PluginBase:

- 2.1 In der Solution *CalculatorPluginSample* erstellen Sie ein Library-Projekt mit dem Namen -> *Calculator.PluginBase*.
- 2.2 Erstellen Sie im Library-Projekt *Calculator.PluginBase* das Interface *ICalculatorPlugin.cs*. Das Interface *ICalculatorPlugin.cs* enthält die Properties *Name* und *OperationDescription* (beide vom Typ *string*) und die Methode *Operation*, die zwei *double* Parameter beinhaltet und ein *double* zurück gibt.

Schritt 2: Calculator.AdditionPlugin:

- 2.3 In der Solution *CalculatorPluginSample* erstellen Sie ein Library-Projekt mit dem Namen -> *Calculator.AdditionPlugin*.
- 2.4 Fügen Sie im Projekt *CalculatorPluginSample* den Verweis des Projekts *Calculator.PluginBase* hinzu.

- 2.5 Erstellen Sie in der Library *Calculator.AdditionPlugin* die Klasse *AdditionOperation.cs*
- 2.6 Die Klasse *AdditionPlugin* leitet vom *ICalculatorPlugin* ab und implementieren Sie die Klasse aus. Die Property Name enthält den Namen der Rechenart, OperatorDescription beinhaltet den Operator als Zeichen und Operation berechnet die Addition.
- 2.7 Kompilieren Sie die Solution.

Schritt 3: Calculator.App:

- 2.8 In der Solution *CalculatorPluginSample* erstellen Sie ein Consolen-Projekt mit dem Namen -> *Calculator.App*
- 2.9 Kompilieren das Projekt *Calculator.App* und wechseln Sie in das Projekt-Verzeichnis *\bin\Debug\Net6* und legen dort das Verzeichnis *Plugins* an.
- 2.10 Kopieren Sie aus dem *CalculatorPluginSample* die *CalculatorPluginSample.DLL* in das Verzeichnis *Plugins*.
- 2.11 Fügen Sie im Projekt *Calculator.App* den Verweis des Projekts *Calculator.PluginBase* hinzu.
- 2.12 Implementieren Sie ein Programm, welches das *AdditionPlugin* lädt und verwendet.

3 Aufgabenstellung 2:

- 3.1 Erstelle für unsere *CalculatorApp* für die restlichen 3 Grundrechenarten, jeweils ein Plugin und füge die DLLs in das Plugin-Verzeichnis hinzu.
- 3.2 Starte das Programm und lass die das Ergebnis aller 4 Grundrechenarten anzeigen.

