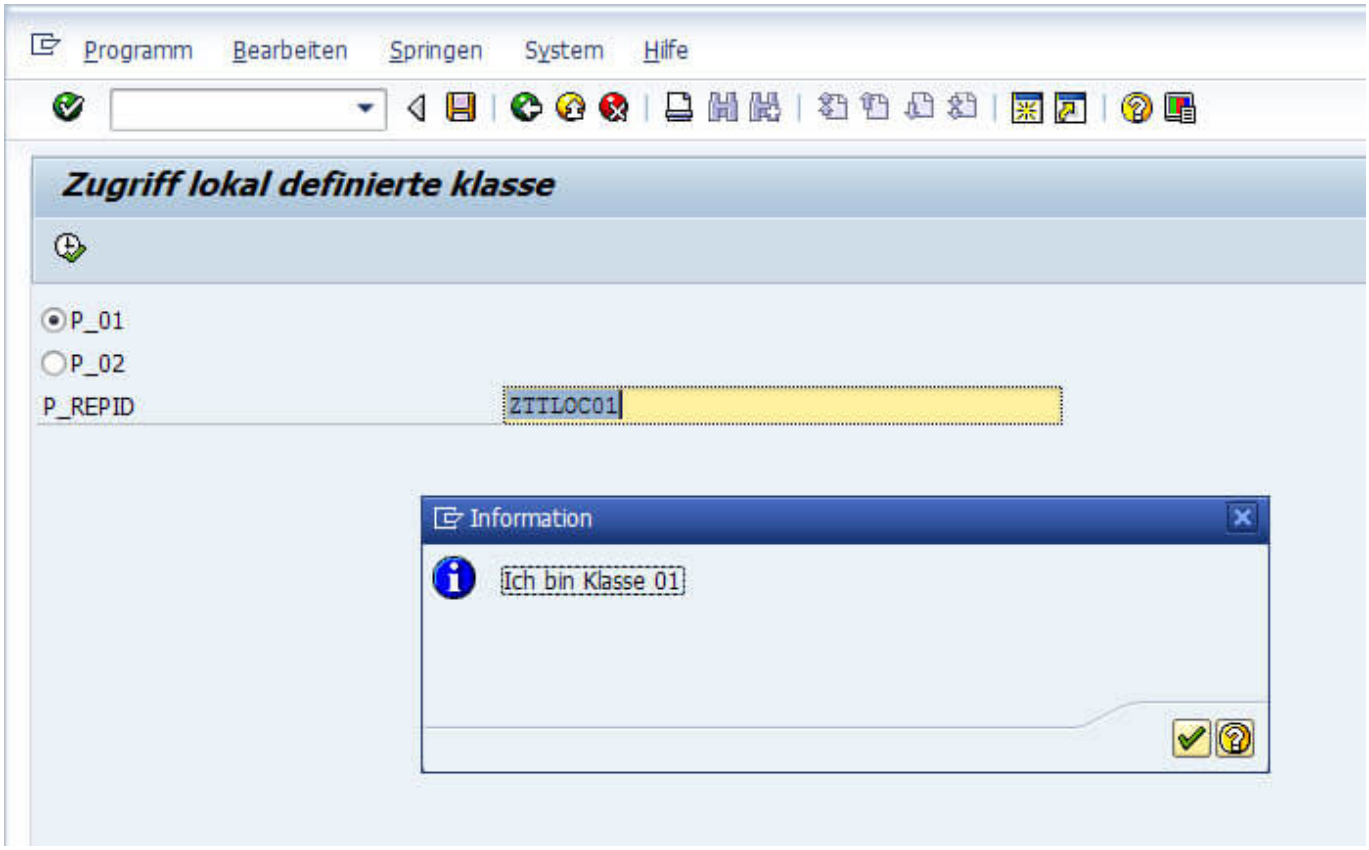


Global. Lokal. Egal.

Im Rahmen eines Projektes haben wir uns mit der dynamischen Erzeugung von Klassen beschäftigt und Stefan hat dabei einen netten Trick herausgefunden.

Globale Klasse

Normalerweise gibt es in einem Report nur die folgenden zwei Möglichkeiten um eine Klasse zu instantiieren:

1. Ein Objekt mit Referenz zu einer **global** definierten Klasse (SE24)
2. Ein Objekt mit Referenz zu einer im selben Programm definierten **lokalen** Klasse

Stefan hat nun eine Möglichkeit gefunden, wie man lokale Klassen, die in beliebigen Programmen definiert sind, erzeugen kann. Und das geht so:

Lokale Klasse

Wir brauchen ein Programm mit einer lokal definierten Klasse. In dem folgenden Programm werden zwei lokale Klassen definiert. Beide haben die Methode INFO.
[crayon-59edd135570e5808642876/]

Dynamische Erzeugung von Klasseninstanzen

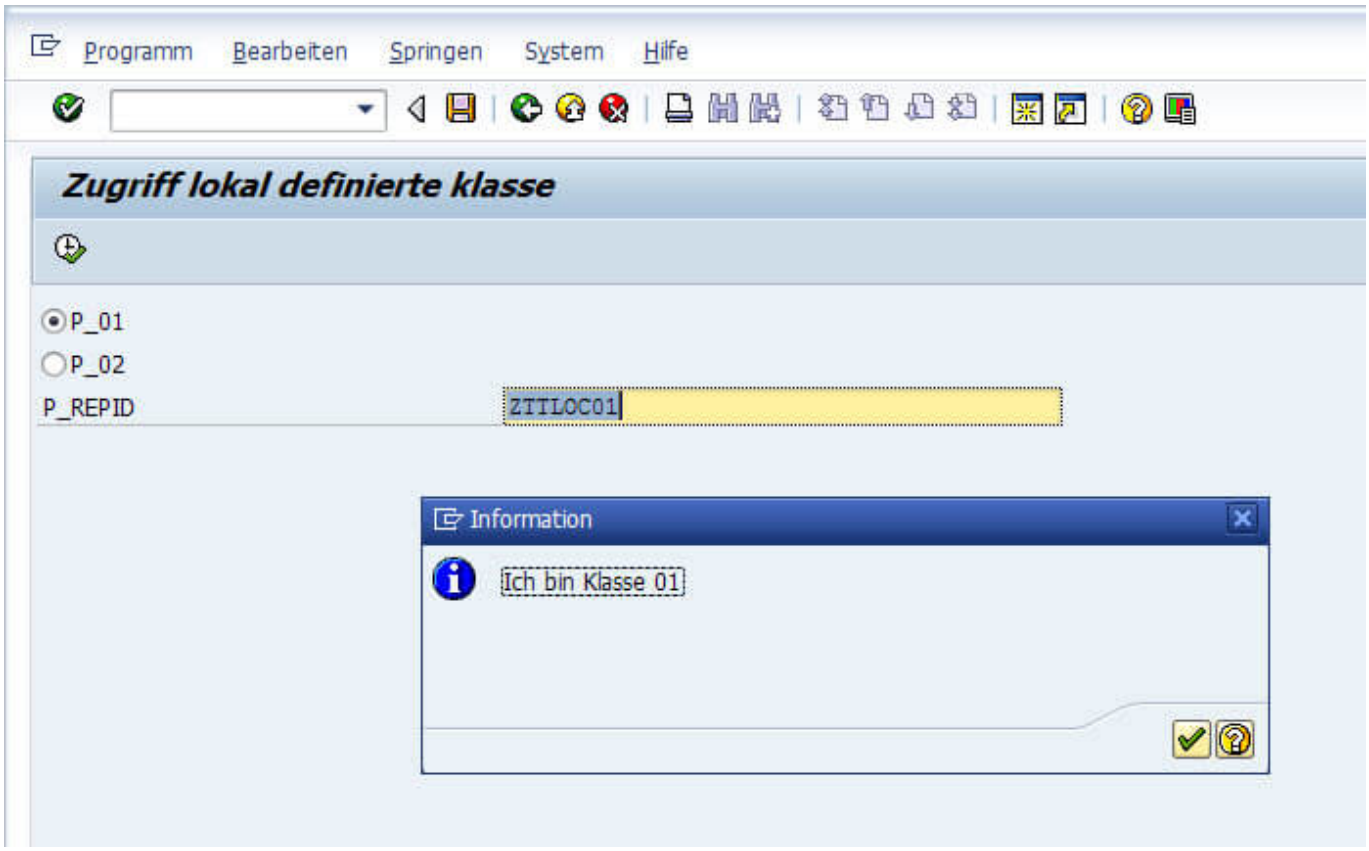
Wenn man den Namen einer Klasse erst zur Laufzeit ermitteln kann, dann kann man ein Objekt dieser Klasse dynamisch wie folgt erzeugen:

[crayon-59edd135570f1806999124/]
Sinnvoller Weise verwendet man hierfür ein Interface, dass alle in Frage kommenden Klassen implementiert haben.

Ihr Auftritt Herr Kollege

Unser Trick besteht nun darin, dass wir den Klassennamen genauer spezifizieren. Wir geben dem Klassennamen die Information mit, in welchem Programm die Klasse vorhanden ist. Der Klassenname sieht dann zum Beispiel wie folgt aus:
[crayon-59edd135570f6009186133/]
Wir müssen also nur noch die intern verwendete Struktur zusammenbasteln. Das folgende Programm erzeugt je nach Auswahl ein Objekt der im Programm ZTTLOC01 lokale definierten Klasse LCL_LOCAL_01 oder LCL_LOCAL_02.
[crayon-59edd135570fb819823531/]

Screenshot



Das rufende Programm muss natürlich genau wissen, was es tut. Je nachdem wie dynamisch der Aufruf einzelner Methoden sein soll, können ebenfalls dynamisch ermittelte Parameter über die Parameterliste übergeben werden.

Verwendet man ein Interface, dass die zu verwendenden Klassen implementieren, dann kann man verwendete Interface-Methoden direkt aufrufen.