Playing with AppDomains & MEF

Übung 1

Vorbereitung

- Windows ist Zielplattform
- Klonen der Übungen:

```
git clone
https://mvodep.visualstudio.com/DefaultCollection/Softwareko
mponentensysteme/_git/Softwarekomponentensysteme
```

Was sind AppDomains?

- MSDN: … an application domain, which is an isolated environment where applications execute.
- Leichtgewichtiger als die Isolation per Prozesse
- Helfen bei
  - Security
  - Reliability (Unhandeled Exceptions können **nicht** gefangen werden)
  - Versioning
  - unloading assemblies

Task 1

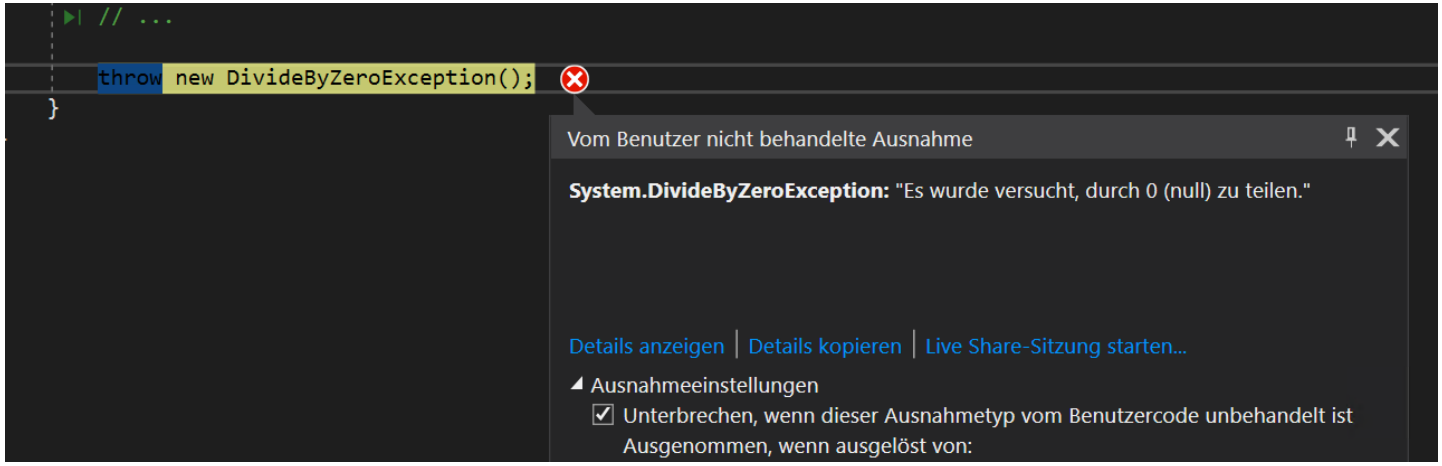
- Erstellen Sie eine Klassenbibliothek „Print“ und laden Sie das Module zur Laufzeit (=es darf kein statischer Verweise in der AppDomains Konsolen-Applikation sein)
- Derzeit wird das aktuelle Gehalt in der „Main“ Methode ausgegeben → das Formatieren des Strings soll in der Print Komponenten (eigene AppDomain PrintDomain) passieren – die Ausgabe bleibt in Main
- Eckpunkte:
  1. Legen Sie eine `Print` Klassenbibliothek an
  2. Legen Sie ein Interface „`IPrintAddIn`“ in `Common` an mit einer Methode „`string FormatSalary(Salary salary)`“
  3. Machen Sie die Implementierung in `PrintAddIn`
  4. Laden Sie das Modul in AppDomains Konsolenapp und rufen sie die Methode `FormatSalary` auf

Task 2

- Ohje – Fehler passieren. Es wurde ein Bug eingebaut: Alle 3 Aufrufe wird eine `ApplicationException` geworfen (e.g. durch falschen State in der AppDomain)
- Fangen Sie den Fehler der Salary Komponente in AppDomains Main und starten Sie das `BillingAddIn` neu
  - Stellen Sie sich vor, dass die viel komplexer ist und 10 andere AddIns vorhanden sind → diese funktionieren aber noch → gezieltes Neustarten macht daher durchaus Sinn
- Nutzen Sie `AppDomain.Unload`

Info

- Visual Studio unterbricht im Debug Modus – einfach abdrehen



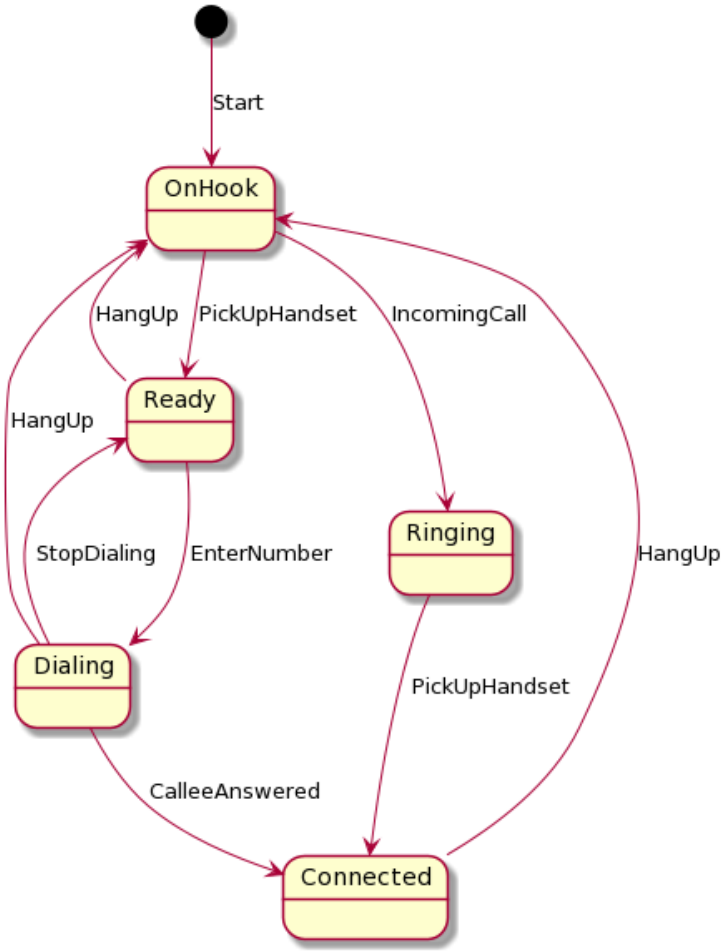
Task 3: MEF

- MEF erlaubt es eine Applikation modular zusammenzubauen: „Gib mir eine Implementierung von `IPlugin2`“
- Bauen Sie ein `Plugin2` welches eine Methode `int Add(int, int)` hat und zwei Zahlen addiert (Projekt Extensibility). Es soll der Interface `IPlugin2` implementieren

Bonus Task 4: Stateful Ports

- https://github.com/dotnet-state-machine/stateless
  - Ähnliches Beispiel: https://github.com/dotnet-state-machine/stateless/tree/dev/example/TelephoneCallExample
- Übergänge = Methoden im Interface
- Bauen der Statemaschine.
- OK:
  - Start - PickUpHandset - EnterNumber - CalleAnswered - HangUp - IncomingCall - PickHandset – HangUp
- NOK:
  - Start - EnterNumber - HangUp

Bonus Task 4: Stateful Ports



Ziel der Übung

- AppDomain
  - Erster Kontakt mit Interfaces
  - Isolation von Komponenten über AppDomains (Speicherisolation in einem Prozess)
  - Fehlerbehandlung in einer Komponente
  - Connector: Serialisierung über Speichergrenzen hinaus (kein IPC)
- MEF
  - Dynamischen laden von Komponenten
- Ports
  - Stateful Ports