

ITKPU Lab MEF

Formål:

At opnå erfaring med brugen af MEF til implementering af plug-ins i .Net applicationer.

Forudsætninger

At du har læst om MEF.

Denne opgave er en kopi af opgave 2, blot bruges den nu C# og MEF.

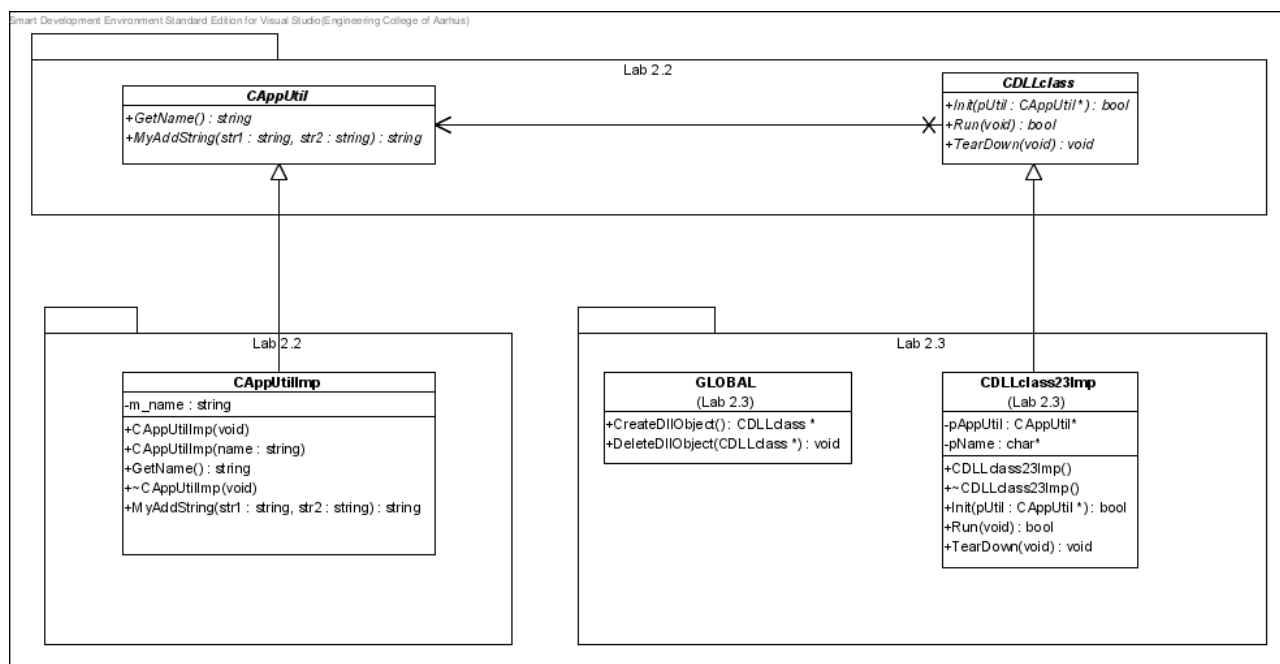
Delopgave 1:

Lav grænsefladen mellem applikationen og plug-in udvidelserne (kontrakten).

- Design interfacet som hovedklassen i assembly'en (plug-in DLL'en) skal implementere.
F.eks.:

```
interface IDLLcontract
{
    Init(IAppUtil util);
    bool Run();
    void TearDown();
}
```
- Design interfacet som assembly'en (plug-in DLL'en) modtager en reference til i funktionen Init.
F.eks.

```
interface IAppUtil
{
    string GetName();
    string MyAddString(string str1, string str2);
}
```



Delopgave 2:

Lav applikationen.

- Implementer en klasse som leverer en implementering af IAppUtil (denne klasse bruges ikke af hovedprogrammet, men kan kaldes fra plug-in DLL'en via call-back).
- Lav hovedprogrammet i applikationen således at det loader alle assemblies i et bestemt subdirectory (mappe) derefter kalder funktionerne Init, Run og TearDown på DIIObjektet i denne assembly.

Delopgave 3:

Lav en plug-in

Lav en plug-in assembly.

1. Lav en assembly (dll) som eksporterer en implementering af interfacet `IDLLcontract`.
Du bestemmer selv hvad implementeringen af funktionerne skal lave, men mindst en af dem skal bruge en funktion fra `IAppUtil`.
2. Test at din applikation kan load og kalde assembly'en.

Delopgave 4:

Lav en ny plug-in DLL.

1. Lav en ny dll som overholder det samme interface som i delopgave 3, men hvor implementeringerne af funktionerne laver noget andet.
2. Test at din applikation (uden at der er lavet ændringer i den) også kan load og kalde den nye DLL'en.

