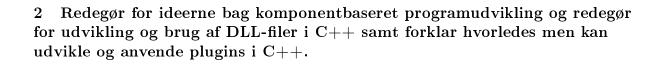
1

 $1 \quad {\bf Redeg \'{o}r} \ {\bf for} \ {\bf ideerne} \ {\bf bag} \ {\bf komponent baser et} \ {\bf programud vikling}, \ {\bf og} \\ {\bf tilh \'{o}rende} \ {\bf design principper}$



r for COM's ar c COM kompor	kitektur og te nenter med br	rminologi sam ug af ATL.	t forklar hvorled	es men
	r for COM's are COM kompor	r for COM's arkitektur og te c COM komponenter med br	r for COM's arkitektur og terminologi sam COM komponenter med brug af ATL.	for COM's arkitektur og terminologi samt forklar hvorlede COM komponenter med brug af ATL.

4 Redegør for . Nets komponentmodel og Lifecycle Management. Samt forklar hvorledes man kan udvikle og anvende komponenter i C#.

5 Redegør for begrebet dependency injection og brugen af IoC-containere, samt interface baseret programmering.

5.1 Injection teknikker

Dependency Injection dækker over teknikker der anvendes til, at indsætte afhængigheder til andre klasser og funktioner.

Mest enkelte teknikker:

Listing 5.1: Constructor Injection 1 public class Foo 2 3 private Bar _bar; Foo(Bar bar) //Constructor Injection 4 5 $_{\mathtt{bar}} = \mathtt{bar};$ 6 7 8 public Bar BarClass //Property Injection 9 10 set { _bar = value; } 11 get { return _bar; } 12 } 13 14

Fordelen ved Construction Injection er, at dette sættes så snart klassen oprettes og der funktioner i klassen ikke kan kaldes og bruge en reference = null.

Property Injection er også en god idé, hvis man på runtime har brug for at udskifte sin afhængighed, men skal sættes som det første, da referencen ellers er null.

5.2 Interfaces

Når man laver afhængigheder til andre klasser, binder man som regel til selve klassen. Problemet med dette er, at det bliver svært at teste, da man skal bruge den konkrete afhængighed i testen, hvilket ikke er ønsket i *unit tests*.

For at komme dette til livs laver man et interface for klassen - også selvom der kun skal være én der arver derfra.

Listing 5.2: Interfaces til injection

```
1 public interface IBar
2
   {}
3
4 public class Bar : IBar
5 {}
6
7
   public class Foo
8
9
     10
     Foo(IBar bar) //IBar
11
12
        _{bar} = bar;
13
14
15
     public IBar BarClass //IBar
16
17
       \mathtt{set} \; \{ \; \mathtt{\_bar} = \mathtt{value} \; ; \; \}
        get { return _bar; }
18
19
     }
20
```

På denne måde kan der testes med et framework, hvor IBar kan stubbes/mockes af som det behager. Ligeledes kan man bruge $public\ IBar\ BarCLass$ til alle typer klasser, der arver fra IBar.

5.3 IoC-Container

Inversion of Control, IoC-Containere, er det, der holder styr på klassernes afhængigheder. Det vil sige, at den ved klassen Foo skal have en klasse af type IBar.

Der findes flere IoC-Container som alle i bund og grund kan det samme, men hvis implementering varierer en smule.

Listing 5.3: IoC-Container registrering 1 public class RegistrationModule 2 private IUnityContainer _container; 3 4 public RegistrationModule(IUnityContainer container) 5 6 ${\tt _container} = {\tt container};$ 7 8 9 public void Initialize() //Implementering fra IUnityContainer 10 $\verb|_container.RegisterType< IBar|, Bar>();$ 11 12 $_$ container.RegisterType<Foo>();131415public class Test 1617 private IUnityContainer _container; 18 private Foo _foo; 19 20 public void TestFunction() 21 22 23 $var foo = _container.Resolve < Foo > ();$ 24 foo.Write(); 25 } 26

Det foregår ved, at man registrerer de enkelte implementering

Giv et overblik over Microsofts forskellige Extensibility Frameworks, dedegør for den grundlæggende arkitektur og begreber i MEF og PRISM	og

7 Redegør for begrebet "Interoperability"generelt, og redegør for brugen a PInvoke samt interoperability mellem COM og .Net.				

8 på	Redegør for problemer of .Net platformen.	og muligheder for	Cross Platform	n Development

9~ Redegør for hvorledes man designer og implementerer Windows RT komponenter.

10 Redegør for problemstillingen omkring komponenter og flertrådede programmer, samt redegør for hvilke faciliteter . Net og C# giver programudvikleren.

11 Redegør for begrebet "Services", og redegør for og implementerer Windowsservices ved brug af .Net	hvorledes m og $C\#$.	ıan designer