

Introduction to Spring.NET

<http://www.springframework.net>

About

- Joeri Torfs
 - CEO Softelligent
 - <http://www.softelligent.be>
- Benny Michielsen
 - Using Spring.NET for 4 years
 - <http://www.bennymichielsen.be>

Agenda

- Dependency Management
- Dependency Injection
- AOP
- Services
- Much More

Agenda

- Dependency Management
- Dependency Injection
- AOP
- Services
- Much More

Dependency Management

```
void copy(void)
{
    int ch;
    while( (ch=ReadKeyboard()) != EOF)
        WritePrinter(ch);
}
```

Dependency Management

```
bool GtapeReader = false; // remember to clear  
  
void copy(void)  
{  
    int ch;  
    while( (ch=GtapeReader ? ReadTape() : ReadKeyboard()) != EOF)  
        WritePrinter(ch);  
}
```

Dependency Management

```
bool GtapeReader = false;
Bool GtapePunch = false;
// remember to clear

void copy(void)
{
    int ch;
    while( (ch=GtapeReader ? ReadTape() : ReadKeyboard()) != EOF)
        GtapePunch ? WritePunch(ch) : WritePrinter(ch);
}
```

Dependency Management

- Rigid
unknown impact of change
- Fragile
cascading changes, new errors
- Not reusable
developing from scratch can cost less
- Code Rots

Dependency Management

```
void Copy()
{
    int c;
    while( (c=getchar()) != EOF)
        putchar(c);
}
```

Dependency Management

```
interface Reader
{ char read(); }

interface Writer
{ void write(char c); }

public class Copy
{
    Copy(Reader r, Writer w)
    {
        itsReader = r;
        itsWriter = w;
    }
    public void copy()
    {
        int c;
        while( (c==itsReader.read()) != EOF )
            itsWriter.write(c);
    }
    private Reader itsReader;
    private Writer  itsWriter;
}
```

Agenda

- Dependency Management
- Dependency Injection
- AOP
- Services
- Much More

Dependency Injection

Demo !

Dependency Injection

- Heart of the framework
- Better separation of concerns
- Less dependencies
- Pluggable applications

Agenda

- Dependency Management
- Dependency Injection
- AOP
- Services
- Much More

AOP

Wikipedia:

“Aspect-oriented programming (AOP) is a programming paradigm that increases modularity by allowing the separation of cross-cutting concerns, forming a basis for Aspect-oriented software development.”

?

AOP

Demo !

AOP

- Crosscutting (recurring) concerns (code across multiple classes) that can not be resolved by moving it up the inheritance tree can be handled by using Aspect-oriented Programming.
- Logging, security, transactions, caching...

AOP

- Aspect = cross-cutting concern (Advisors / Interceptors)
- Joinpoint = point during execution
- Advice = action on joinpoint (around, before, throws)
- Pointcut = set of joinpoints specifying when advice should be executed

AOP

- Introduction = introducing new interfaces or fields
- Target = object containing the joinpoint (advised/proxied object)
- AOP Proxy = object with advice
- Weaving = assembling aspects to create AOP Proxy

AOP

- Several out of the box:
Logging
Exception Handling
Caching
Transaction
Retry
- Extensible

Agenda

- Dependency Management
- Dependency Injection
- AOP
- Services
- Much more

Services

Demo !

Services

- Encourages POCO classes
- Decide distribution later
- Hosting = configuration

Agenda

- Dependency Management
- Dependency Injection
- AOP
- Services
- Much More

Much more

- Web Framework
- Validation
- Data & NHibernate
- Testing
- Messaging
- ...

Other frameworks

- Castle
www.castleproject.org
- StructureMap
structuremap.sourceforge.net
- Unity
www.codeplex.com/unity
- PostSharp
www.postsharp.org
- ...
www.google.com

Q & A