

# Függőség befecskendezés (1)

- Felhasznált irodalom:
  - Dhanji R. Prasanna. *Dependency Injection Design patterns using Spring and Guice*. Manning, 2009.  
<https://www.manning.com/books/dependency-injection>
  - Mark Seemann. *Dependency Injection in .NET*. Manning, 2011.  
<https://www.manning.com/books/dependency-injection-in-dot-net>
  - Steven van Deursen, Mark Seemann. *Dependency Injection*. Second Edition. Manning, 2019.  
<https://www.manning.com/books/dependency-injection-in-dot-net-second-edition>

# Függőség befecskendezés (2)

- A függőség befecskendezés (DI – *dependency injection*) kifejezés Martin Fowlertől származik.
  - Martin Fowler. *Inversion of Control Containers and the Dependency Injection pattern*. 2004.  
<https://martinfowler.com/articles/injection.html>
- A vezérlés megfordítása (IoC – *inversion of control*) nevű architekturális minta alkalmazásának egy speciális esete.
  - Martin Fowler. *InversionOfControl*. 2005.  
<https://martinfowler.com/bliki/InversionOfControl.html>

# Függőség befecskendezés (3)

- Definíció (Seemann):
  - *„Dependency Injection is a set of software design principles and patterns that enable us to develop loosely coupled code.”*
  - A függőség befecskendezés olyan szoftvertervezési elvek és minták összessége, melyek lazán csatolt kód fejlesztését teszik lehetővé.
- A lazán csatoltság a kód karbantarthatóságát javítja.

# Függőség befeccskendezés (4)

- Egy objektumra egy olyan szolgáltatásként tekintünk, melyet más objektumok kliensként használnak.
- Az objektumok közötti kliens-szolgáltató kapcsolatot függésnek nevezzük. Ez a kapcsolat tranzitív.

# Függőség befecskendezés (5)

- **Függőség (*dependency*)**: egy kliens által igényelt szolgáltatást jelent, mely a feladatának ellátásához szükséges.
- **Függő (*dependent*)**: egy kliens objektum, melynek egy függőségre vagy függőségekre van szüksége a feladatának ellátásához.
- **Objektum gráf (*object graph*)**: függő objektumok és függőségeik egy összessége.
- **Befecskendezés (*injection*)**: egy kliens függőségeinek megadását jelenti.
- **DI konténer (*DI container*)**: függőség befecskendezési funkcionalitást nyújtó programkönyvtár.
  - Az *Inversion of Control (IoC) container* kifejezést is használják rájuk.

# Függőség befecskendezés (6)

- A függőség befecskendezés objektum gráfok hatékony létrehozásával, ennek mintáival és legjobb gyakorlataival foglalkozik.
- A DI keretrendszerek lehetővé teszik, hogy a kliensek a függőségeik létrehozását és azok befecskendezését külső kódra bízzák.

# Függőség befecskendezés (7)

- Példa: nincs függőség befecskendezés

```
public interface SpellChecker {  
    public boolean check(String text);  
}  
  
public class TextEditor {  
    private SpellChecker spellChecker;  
  
    public TextEditor() {  
        spellChecker = new HungarianSpellChecker();  
    }  
  
    // ...  
}
```

# Függőség befecskendezés (8)

- Függőség befecskendezés konstruktorral (*constructor injection*):

```
public class TextEditor {  
    private SpellChecker spellChecker;  
  
    public TextEditor(SpellChecker spellChecker) {  
        this.spellChecker = spellChecker;  
    }  
  
    // ...  
}
```



# Függőség befecskendezés (9)

- Függőség befecskendezés beállító metódussal (*setter injection*):

```
public class TextEditor {  
    private SpellChecker spellChecker;  
  
    public TextEditor() {}  
  
    public void setSpellChecker(SpellChecker spellChecker) {  
        this.spellChecker = spellChecker;  
    }  
  
    // ...  
}
```

# Függőség befecskendezés (10)

- Függőség befecskendezés beállító interfésszel (*interface injection*):

```
public interface SpellCheckerSetter {  
    void setSpellChecker(SpellChecker spellChecker);  
}  
  
public class TextEditor implements SpellCheckerSetter {  
    private SpellChecker spellChecker;  
  
    public TextEditor() {}  
  
    @Override  
    public void setSpellChecker(SpellChecker spellChecker) {  
        this.spellChecker = spellChecker;  
    }  
  
    // ...  
}
```

# Függőség befecskendezés (11)

- C++ keretrendszer:
  - *[Boost].DI* (licenc: *Boost Software License*)  
<http://boost-experimental.github.io/di/>
  - *Fruit* (licenc: *Apache License 2.0*)  
<https://github.com/google/fruit>
  - *Hypodermic* (licenc: *MIT License*)  
<https://github.com/ybainier/Hypodermic>
  - ...

# Függőség befecskendezés (12)

- Java:
  - *JSR 330: Dependency Injection for Java*  
<https://www.jcp.org/en/jsr/detail?id=330>
    - Szabványos annotációk biztosítása függőség befecskendezéshez.
    - A Java EE 6-ban jelent meg.
    - `javax.inject` csomag.
      - Lásd:  
<https://javaee.github.io/javaee-spec/javadocs/javax/inject/package-summary.html>
    - A specifikációt implementáló DI keretrendszer szükséges használatához!
      - Például: *Dagger*, *Guice*, *HK2*, *Spring Framework*, ...

# Függőség befecskendezés (13)

- Java keretrendszerek:
  - *Dagger* (licenc: *Apache License 2.0*)  
<https://google.github.io/dagger/>
  - *Guice* (licenc: *Apache License 2.0*)  
<https://github.com/google/guice>
  - *HK2* (licenc: CDDL + GPLv2) <https://hk2.java.net/>
  - *Java EE CDI* <https://javaee.github.io/tutorial/cdi-basic.html>
  - *Spring Framework* (licenc: *Apache License 2.0*)  
<https://projects.spring.io/spring-framework/>  
<http://www.vogella.com/tutorials/SpringDependencyInjection/article.html>
  - ...

# Függőség befecskendezés (14)

- .NET keretrendszer:
  - *Castle Windsor* (licenc: *Apache License 2.0*)  
<https://github.com/castleproject/Windsor>
  - *Ninject* (licenc: *Apache License 2.0*)  
<http://www.ninject.org/>
  - *StructureMap* (license: *Apache License 2.0*)  
<http://structuremap.github.io/>
  - ...