

# Implementacja wydajnego wzorca wstrzykiwania zależności dla złożonych grafów zależności

Adrian Mularczyk

Uniwersytet Wrocławski  
Wydział Matematyki i Informatyki  
Kierunek: Informatyka

# Agenda

- 1 Przedstawienie problemu
- 2 Wstrzykiwanie zależności
- 3 Implementacja
- 4 Wyniki
- 5 Podsumowanie

# SOLID

- S - SRP (Single responsibility principle)
- O - OCP (Open/closed principle)
- L - LSP (Liskov substitution principle)
- I - ISP (Interface segregation principle)
- D - DIP (Dependency inversion principle)

# Dependency Inversion Principal

Wysokopoziomowe moduły nie powinny zależeć od modułów niskopoziomowych. Zależności między nimi powinny wynikać z abstrakcji.

# Kontenery wstrzykiwania zależności

Obiekt, który przechowuje mapę, w której abstrakcje (interfejsy, klasy abstrakcyjne) mają przyporządkowane implementacje (klasy implementujące interfejsy lub dziedziczące z klas abstrakcyjnych).

# Kontenery wstrzykiwania zależności

- Rejestracja
- Tworzenie obiektów

# Problem

- W grafach zależności często powtarzają się typy
- Utworzenie instancji nowego obiektu zajmuje czas.
- Nowe obiekty są często tworzone

# Cel Pracy

Stworzenie wydajnego kontenera wstrzykiwania zależności, który będzie wydajny dla złożonych grafów zależności, a także który będzie efektywnie tworzył kolejne instancje tej samej klasy.



# Rodzaje wstrzykiwań zależności

- Wstrzykiwanie przez konstruktor.
- Wstrzykiwanie przez metodę.
- Wstrzykiwanie przez właściwość.

# Implementacje przemysłowe

- Unity
- Ninject
- Autofac
- StructureMap
- Windsor
- Grace
- Dryloc
- LightInject
- SimpleInjector

# Implementacja

- CIL
- Reflection.Emit

# Dwa rozwiązania

- Partial Emit Function
- Full Emit Function

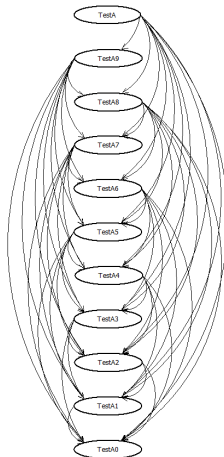
# Partial Emit Function

<jakiś rysunek że z kawałku tworzy się co nowego>

# Full Emit Function

<jakiś rysunek że jest jeden duży kawałek>

# Przypadek testowy A - graf zależności



# Przypadek testowy A - Transient

Liczba iteracji: 1 i 10

Autofac	0
<b>NiquoloCPartial</b>	<b>1</b>
Windsor	1
<b>NiquoloCFull</b>	<b>8</b>
Unity	8
LightInject	10
StructureMap	10
Ninject	11
SimpleInjector	13
Dryloc	14
Grace	15

<b>NiquoloCPartial</b>	<b>3</b>
Autofac	6
<b>NiquoloCFull</b>	<b>8</b>
LightInject	10
SimpleInjector	14
StructureMap	14
Dryloc	15
Grace	16
Unity	16
Windsor	16
Ninject	90



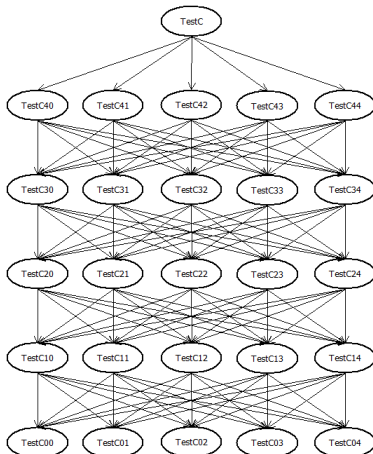
## Przypadek testowy A - Transient

Liczba iteracji: 100 i 1000

<b>NiquloCFull</b>	<b>9</b>
LightInject	11
SimpleInjector	15
Dryloc	16
Grace	18
<b>NiquloCPartial</b>	<b>19</b>
StructureMap	54
Autofac	59
Unity	88
Windsor	155
Ninject	882

<b>NiquloCFull</b>	<b>18</b>
LightInject	19
SimpleInjector	29
Dryloc	29
Grace	37
<b>NiquloCPartial</b>	<b>173</b>
StructureMap	417
Autofac	587
Unity	813
Windsor	1529
Ninject	8934

## Przypadek testowy C - graf zależności



## Przypadek testowy C - Transient

Liczba iteracji: 1 i 10

Autofac	2
<b>NiquoloCPartial</b>	<b>4</b>
Windsor	7
Unity	19
StructureMap	26
<b>NiquoloCFull</b>	<b>31</b>
LightInject	36
Dryloc	39
Ninject	39
SimpleInjector	41
Grace	61

<b>NiquoloCPartial</b>	<b>10</b>
Autofac	24
<b>NiquoloCFull</b>	<b>32</b>
LightInject	37
Dryloc	40
StructureMap	40
SimpleInjector	41
Unity	47
Grace	62
Windsor	62
Ninject	353

## Przypadek testowy C - Transient

Liczba iteracji: 100 i 1000

<b>NiquloCFull</b>	<b>38</b>
LightInject	44
Dryloc	47
SimpleInjector	49
Grace	73
<b>NiquloCPartial</b>	<b>74</b>
StructureMap	181
Autofac	231
Unity	318
Windsor	608
Ninject	3511

<b>NiquloCFull</b>	<b>82</b>
LightInject	86
Dryloc	99
SimpleInjector	116
Grace	155
<b>NiquloCPartial</b>	<b>690</b>
StructureMap	1540
Autofac	2288
Unity	3015
Windsor	6037
Ninject	37642

# Podsumowanie