

Projektowanie obiektowe oprogramowania

Zestaw 13 (ostatni)

Inversion of Control (4)

2012-05-08

Liczba punktów do zdobycia: **6/80**

Zestaw ważny do: 2012-05-22

Uwaga! Kontynuacja pracy nad silnikiem Inversion of Control na identycznych zasadach. W szczególności obowiązkową częścią każdego zadania są testy jednostkowe, nawet jeśli nie wspomina się o tym w treści zadań.

1. (1p) (Łańcuchowanie wywołań metod)

Silnik IoC rozbudować o możliwość łańcuchowania wywołań metod na obiekcie kontenera:

```
SimpleContainer c = new SimpleContainer();  
  
c  
  .RegisterType<Foo>( true )  
  .RegisterType<IBar, Baz>( false )  
  .RegisterInstance<IQux>( qux );
```

2. (3p) (Konfiguracja deklaratywna)

Silnik IoC rozbudować o możliwość konfiguracji deklaratywnej, dostarczanej w postaci plików *.xml:

```
<?xml version="1.0"?>  
<simpleIoCContainer>  
  <resolvableTypes>  
    <add type="Foo" singleton="true" />  
    <add type="IBar" mapsTo="Baz" /> /* brak atrybutu singleton oznacza singleton=false */  
  </resolvableTypes>  
</simpleIoCContainer>
```

Aplikowanie konfiguracji:

```
SimpleContainer c = new SimpleContainer();  
SimpleContainerConfiguration iocConfig =  
  SimpleContainerConfiguration.CreateFromFile( "simpleIoCContainer.xml" );  
  
iocConfig.Configure( c );
```

Uwaga! Pamiętać o tym, że na platformie .NET pełna nazwa typu zapisana literalnie, umożliwiającą tworzenie instancji za pomocą refleksji to zawsze Nazwa.Przestrzeni.Nazw.NazwaTypu, Nazwa.Pliku.Biblioteki.

3. (2p) (Testy wydajności kontenera)

Przygotować testy wydajności w następujących kategoriach:

- Tworzenie instancji typu (niezarejestrowanego)
- Tworzenie instancji typu (zarejestrowanego jako implementacja interfejsu lub klasy abstrakcyjnej)
- Wstrzykiwanie pojedynczego parametru do konstruktora
- Łańcuch co najmniej 3 klas wstrzykujących się do siebie przez konstruktory
- Wstrzykiwanie do właściwości
- Uzupełnianie zależności obiektu skonstruowanego poza kontenerem

Dla każdej kategorii przygotować wskaźnik wydajności dla:

- Wywołania "wprost", bez kontenera IoC
- Własnej implementacji IoC
- Wybranej przemysłowej implementacji IoC (np. Unity, Ninject)

Wyniki interpretować według następujących kluczy:

Jak bardzo zarządzanie obiektami za pomocą kontenera jest wolniejsze od zarządzania nimi wprost? Jak własna implementacja silnika wypada w zderzeniu z implementacją przemysłową?

Wiktor Zychła