

Projektowanie obiektowe oprogramowania

Zestaw 5

Wzorce podstawowe i kreacyjne

2012-03-07

Liczba punktów do zdobycia: **5/30**

Zestaw ważny do: 2012-03-20

Uwaga! Obowiązkową częścią każdego rozwiązania są testy jednostkowe.

1. **(1p) (Singleton)** Przygotować implementacje singletonów o następujących politykach czasu życia:

- jedna instancja dla całego procesu
- jedna instancja dla każdego wątku
- jedna instancja na co najwyżej 5 kolejnych sekund

Dostarczyć właściwe testy.

2. **(1p) (Factory)** Zaimplementować fabrykę obiektów dowolnego typu, parametryzowaną nazwą typu i flagą określającą czy obiekt ma być singletonem. Dostarczyć właściwe testy.

```
public class GenericFactory
{
    public object CreateObject( string TypeName, bool IsSingleton );
}

// klient
GenericFactory factory = new GenericFactory();
var s1 = factory.CreateObject( "String", false );
var o1 = factory.CreateObject( "UWr.Examples.ExampleType", true );
```

3. **(1p) (Object Pool)** Zaimplementować klasę `Airport` dostarczającą instance obiektów typu `Plane`. Założyć że lotnisko dysponuje ograniczoną liczbą samolotów. Poprawnie obsłużyć przypadki realizowane przez pulę: tworzenie, zwalnianie, przekroczenie rozmiaru puli. Dostarczyć właściwe testy.

4. **(2p) (Builder)** Przykład z wykładu

<http://netpl.blogspot.com/2012/02/simple-fluent-and-recursive-tag-builder.html>

rozwinąć o obsługę wcięć (z wcięciami/bez wcięć/głębokość wcięć), po to żeby można było napisać:

```
StringWriter writer = new StringWriter(...);
TagBuilder tag     = new TagBuilder( writer );
tag.IsIndented    = true;
tag.Indentation   = 4;
```

```
tag.StartTag( "parent" )
  .AddAttribute( "parentproperty1", "true" )
  .AddAttribute( "parentproperty2", "5" )
  .StartTag( "child1" )
    .AddAttribute( "childproperty1", "c" )
    .AddContent( "childbody" )
    .EndTag()
  .StartTag( "child2" )
    .AddAttribute( "childproperty2", "c" )
    .AddContent( "childbody" )
    .EndTag()
  .EndTag()
.StartTag( "script" )
.AddContent( "$.scriptbody();" )
.EndTag();
```

Dostarczyć właściwe testy.

Wiktor Zychła