

Programowanie pod Windows

Zestaw 5

Język C#4, C#5

2024-03-19

Liczba punktów do zdobycia: **8/44**

Zestaw ważny do: 2024-04-09

1. (**3p**) Przeprowadzić testy porównawcze szybkości kodu, w którym metoda będzie miała parametr raz typu konkretnego, a drugi raz - dynamicznego. Jak bardzo wolniejsze jest wykonywanie kodu dynamicznego w tym konkretnym przypadku i czy w ogóle jest? Jak najlepiej wytłumaczyć różnice?

```
public int DoWork1( int x, int y )
{
    // jakieś obliczenia na x i y, np. x + y
}

public dynamic DoWork2( dynamic x, dynamic y )
{
    // te same obliczenia na x i y
}
```

Do przeprowadzenia testów użyć biblioteki **BenchmarkDotNET** (<https://benchmarkdotnet.org/>) którą proszę zainstalować w projekcie za pomocą menedżera pakietów NuGet.

Czy i jak wyniki zmieniają się w zależności od tego jak bardzo złożone jest wyrażenie używające zmiennych?

2. (**3p**) Zademonstrować w praktyce możliwość dynamicznej implementacji operacji na obiekcie dziedziczącym z **DynamicObject**. Konkretnie - pokazać jak używać

- TryGetMember, TrySetMember
- TryGetIndex, TrySetIndex
- TryInvoke, TryInvokeMember
- TryUnaryOperation, TryBinaryOperation

3. (**1p**) Powtórzyć przykład z wykładu, w którym zaprezentowano w jaki sposób można osiągnąć efekt, w którym "oczekiwalny" (ang. *awaitable*) może być dowolny obiekt. Konkretnie, chodzi o przykład w którym można napisać

```
Console.WriteLine( "1" );

await 2000; // (1)

Console.WriteLine( "1" );
```

gdzie instrukcja **(1)** powoduje dwusekundowe oczekiwanie przed wykonaniem kolejnej linii.

4. (**1p**) Na podobieństwo poprzedniego zadania - dostarczyć takiego rozszerzenia "oczekiwanego" napisu, które spowoduje, że

```
Console.WriteLine( await "https://www.google.com" ); //
```

spowoduje pobranie zawartości witryny spod wskazanego adresu i zwrócenie napisu - zawartości witryny. Formalnie, chodzi o implementację metody

```
public static TaskAwaiter<string> GetAwaiter( this string url )
{
    // uzupełnić
}
```

Do pobrania zawartości witryny o wskazanym adresie użyć obiektu **HttpClient**. Uwaga! Użyć go poprawnie, w szczególności:

<https://www.aspnetmonsters.com/2016/08/2016-08-27-httpclientwrong/>

Wiktor Zychła