

Składowe statyczne

Przegląd zagadnień

- Pole statyczne
- Metody statyczne
- Konstruktor statyczny
- Klasy statyczne
- Metody rozszerzające (Extension Methods)
- Wzorec projektowy Singleton

Pole statyczne

- Wspólne dla wszystkich obiektów danej klasy
- Atrybut klasy, nie obiektu
- Można się do niego odwołać
 - przed utworzeniem obiektu
 - bezpośrednio po nazwie (w metodach danej klasy)
 - przy pomocy nazwy klasy (nie wolno nazwy zmiennej i this)

`NazwaKlasy.nazwa_pola`

```
[modyfikatorDostępu] static typPola nazwaPola
    [=wyrażenieInicjalizujące];
```

Metody statyczne

- Służą do dostępu do pól statycznych
- Nie można odwołać się do zwykłych składowych
 - Nie jest przekazywany argument this
- Można się do nich odwołać
 - przed utworzeniem obiektu
 - bezpośrednio po nazwie (w metodach danej klasy)
 - przy pomocy nazwy klasy (nie wolno nazwy zmiennej i this)

```
[modyfikatorDostępu] static typZwracany
    nazwaMetody([lista_argumentow])
{ //ciało metody
}
```

Konstruktor statyczny

- Służy do inicjalizacji pól statycznych
- Nie wolno odwołać się do zwykłych składowych
- Jest wywołany
 - w sposób niejawny
 - przed odwołaniem się do składowej statycznej danej klasy
 - przed utworzeniem obiektu danej klasy

```
static nazwaKlasy()
{
    //ciało konstruktora
}
```

Klasy statyczne

- Może posiadać tylko składowe statyczne
- Nie możemy tworzyć obiektów tej klasy, a nawet deklarować zmiennych
- Substytut funkcji i zmiennych globalnych
- Przykład klasy statycznej: System.Math

```
[modyfikatorDostępu] static class NazwaKlasy
{
    //definicja klasy
}
```

Metody rozszerzające (Extension Methods)

```
static class RozszerzenieDouble{
    public static double Potega(this double x, int n){
        double p = 1;
        for (int i = 0; i < (n < 0 ? -n : n); i++){
            p *= x;
        }
        return n < 0 ? 1 / p : p;
    }
}
```

```
class Program{
    static void Main(string[] args){
        double a = 2;
        Console.WriteLine(a.Potega(2));
        Console.WriteLine((2.0).Potega(-2));
    }
}
```

Metody rozszerzające: ograniczenia

- Nie można rozszerzyć istniejącej metody - o takiej samej sygnaturze
 - nowo zdefiniowana metoda nie będzie wywołana
- Idea metod rozszerzających nie dotyczy pól, właściwości i zdarzeń
- Nie ma dostępu do prywatnych składowych rozszerzanej klasy
- Podsumowując: zwykła metoda statyczna (stodzik)

Wzorzec projektowy Singleton

- Stworzenia klasy, która posiadałaby co najwyżej jeden swój obiekt oraz dostarczenie łatwego sposobu dostępu do tego obiektu

```
public class Singleton
{
    private static Singleton obiekt = new Singleton();
    private Singleton()
    { }
    public static Singleton ZwrocObiekt() {
        return obiekt;
    }
}
```