Pojęcie klasy

Przegląd zagadnień

- Programowanie obiektowe – podstawowe pojęcia
- Definicja klasy w języku C#
- Korzystanie z obiektu
- Słowo kluczowe this
- Ukrywanie informacji – modyfikatory dostępu
- Klasy częściowe
- Metody częściowe
- Struktury – słowo kluczowe struct
- Język UML

Programowanie obiektowe: podstawowe pojęcia

- Klasa i obiekt
  - Definicja obiektu w języku C#
- Stan obiektu
- Abstrakcja
- Hermetyzacja lub enkapsulacja (ang. encapsulation)
- Dziedziczenie (ang. inheritance)
- Polimorfizm (ang. polymorphism)
- Program komputerowy

Definicja klasy w języku C#

```
[modyfikatorDostępu] class NazwaKlasy
{
      // pola i metody
}
```

Składowe klasy

- Pola lub zmienne

```
[modyfikatorDostępu] typPola nazwaPola [= wyrażenie];
```

- Metody składowe (funkcjonalność)

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

Korzystanie z obiektu

- Tworzymy przy pomocy operatora new

```
Klasa1 zmienna1;
zmienna1 = new Klasa1();
```

- lub

```
Klasa1 zmienna1 = new Klasa1();
```

- Odwołanie do składowych

```
zmienna1.X = 20;          //odwołanie się do pola
zmienna1.Metoda1();       //wywołanie metody
```

Słowo kluczowe this

```
class Klasa1{
   public int X;
   public void Metoda1(){
     Console.WriteLine("X = {0}", X);
   }
}
```

```
class Klasa1{
   public int X;
   public static void Metoda1(ref Klasa1 this){
     Console.WriteLine("X = {0}", this.X);
   }
}
```

```
Klasa1 a = new Klasa1();
a.Metoda1();
```

```
Klasa1 a = new Klasa1();
Klasa1.Metoda1(ref a);
```

Ukrywanie informacji: modyfikatory dostępu

- **private**: dostępne tylko z metod danej klasy
- **public**: ogólnie dostępne
- **protected**: wzdłuż ścieżek dziedziczenia
- **internal**: wewnątrz bibliotek (przestrzeni nazw)
- **internal protected**: internal **lub** protected

Klasy częściowe

```
//Plik1.cs
partial class Klasa1
{
      //częściowa definicja klasy
}
```

```
//Plik2.cs
partial class Klasa1
{
      //częściowa definicja klasy
}
```

Metody częściowe - przykład

```
//Pilk A
partial class Osoba
{
   public void f()
   {
      ...
      int a;
      g(a++);
      ....
   }
   partial void g(int a);
}
```

```
//Pilk B
partial class Osoba
{
   partial void g(int a)
   {
      ...
   }
}
```

Metody częściowe: własności

- muszą być definiowane w klasach częściowych (partial)
- typem przekazywanym (zwracanym) musi być void
- nie mogą posiadać modyfikatora dostępu (jak również innych modyfikatorów, takich jak virtual, sealed czy new), domyślnie prywatne
- mogą być metodami statycznymi lub „obiektu"
- można przekazywać do nich parametry poza przesyłaniem argumentu jako parametr wyjściowy „out"
- „nie musi być części implementacyjnej"

Struktury: słowo kluczowe struct

- Typ bezpośredni
- Nie jest w pełni typem obiektowym, ale
  - może zawierać pola oraz metody
  - możemy sterować dostępem do składowych przy pomocy słów kluczowych public i private
  - metodzie zdefiniowanej wewnątrz struktury możemy używać słowa kluczowego this
  - można definiować struktury częściowe i definiować wewnątrz nich metody częściowe.

Język UML

```
class NazwaKlasy{
   public int Pole1;
   private double pole2;
   public int Metoda1(string arg1){}
   private void metoda2(){}
}
```

NazwaKlasy
+Pole1 : int -Pole2 : double
+Metoda1(in arg1 : string) : int -metoda2() : void