



# Luento 15

ITKP102 Ohjelmointi 1

Antti-Jussi Lakanen

# Harkkatyö

# Muuttujien näkyvyys



# Luento 15

- Muuttujien näkyvyys
- Tällä luennolla käsitellään lähinnä paikallisia muuttujia.
- Kuvaus kaikista erilaisista muuttujakategorioista löytyy täältä
- Lisätietoa: <https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/language-specification/variables>



# Muuttujien näkyvyys

- Paikallinen muuttuja: Muuttuja näkyy vain siinä kontekstissa (so. koodilohko, kuten funktio tai luokka) jossa se on luotu.



```
5
6 public class Nakyyvyys
7 {
8     public static void Main()
9     {
10     }
11
12     public static void A(int parametri)
13     {
14         int luku = 5;
15         parametri++;
16     }
17
18     public static void B()
19     {
20         luku++;
21         parametri--;
22     }
```

Cannot resolve symbol 'parametri'



# Muuttujien näkyvyys

- Toisaalta muuttuja näkyy kaikkiin ko. kontekstin sisällä oleviin lohkoihin (tietyin poikkeuksin).

```

1      using ...
2
3
4
5
6      public class Nakyvyys
7      {
8          public static void Main()
9          {
10              int luku = 3;
11              if (luku == 3)
12              {
13                  Console.WriteLine($"{luku} on oikeasti 3");
14              }
15
16
17
18
19
20
21
22
23
24
25      }
26  }
```



```

1      using ...
2
3
4
5
6      public class Nakyyys
7      {
8          public static void Main()
9          {
10              int luku = 3;
11              if (luku == 3)
12              {
13                  Console.WriteLine($"{luku} on oikeasti 3");
14              }
15
16              for (int i = 0; i < 5; i++)
17              {
18                  Console.WriteLine($"Onkos {luku} vieläkin 3?");
19              }
20
21
22
23
24
25      }
26  }
```

```

1      using ...
5
6      public class Nakyyys
7      {
8          public static void Main()
9          {
10             int luku = 3;
11             if (luku == 3)
12             {
13                 Console.WriteLine($"{luku} on oikeasti 3");
14             }
15
16             for (int i = 0; i < 5; i++)
17             {
18                 Console.WriteLine($"Onkos {luku} vieläkin 3?");
19             }
20
21             void SisakkainenAliohjelma()
22             {
23                 Console.WriteLine($"{luku}-muuttujaa voi käyttää jopa täälläkin");
24             }
25         }
26     }

```



# Muuttujien näkyvyys

- Luokan attribuutti (engl. *field*) näkyy kaikille luokan aliohjelmille




1 usage

```
10 10 public class Fysiikkapeli1 : PhysicsGame
11 {
12     private PhysicsObject pallo;
13
14     public override void Begin()
15     {
16         pallo = new PhysicsObject(width:30, height:30);
17         Add(pallo);
18     }
19
20     public void MuutaPalloa()
21     {
22         pallo.Y = 30;
23     }
24 }
```



# Attribuutit = globaalit muuttujat?

- Attribuuttien käyttöä pitäisi kuitenkin välttää, mikäli on tapa tehdä sama asia paikallisia muuttujia käyttäen
- Attribuutit voidaan ymmärtää tällä kurssilla ns. globaaleiksi muuttujiksi
  - **Ylläpidettävyys.** Eri osat tulevat herkästi riippuviksi toisistaan, jolloin (vain) yksittäisten koodin osien muokkaaminen on vaikeaa tai mahdotonta
  - **Testaus ja debuggaus.** Yksikkötestaus voi muuttua hankalaksi mahdottomaksi, jos ohjelman kannalta tärkeää tilatietoa on jatkuvasti ei-paikallisissa muuttujissa.
  - **Luettavuus.** Koodin ymmärrettävyys kärsii, kun tarvittavat muuttujat ovat fyysisesti kaukana.

```
10  public class Fysiikkapeli1 : PhysicsGame
11 {
12
13  public override void Begin()
14 {
15     PhysicsObject pallo = new PhysicsObject(width:30, height:30);
16     Add(pallo);
17     pallo.Position = AnnaUusiPaikka(pallo.Position);
18 }
19
20  1 usage
21 public Vector AnnaUusiPaikka(Vector vanhaPaikka)
22 {
23     Vector uusiPaikka = vanhaPaikka + new Vector(X:0, Y:30);
24     return uusiPaikka;
25 }
```



# Lohkot

- Aliohjelma on oma lohkonsa, kuten myös muun muassa for-lause ja if-lause.

```

1      using ...
2
3
4
5
6      public class Nakyvyys
7      {
8          public static void Main()
9          {
10              int luku = 3;
11              if (luku == 3)
12              {
13                  Console.WriteLine($"{luku} on oikeasti 3");
14                  string jono = "Pertti"; // Tehdään lohkon sisällä uusi muuttuja
15              }
16
17              for (int i = 0; i < 5; i++)
18              {
19                  Console.WriteLine($"Onkos {luku} vieläkin 3?");
20                  jono = "Kalle";
21              }
22          }
23      }

```

Cannot resolve symbol 'jono'





# Lohkot

- Aliohjelma on oma lohkonsa, kuten myös muun muassa for-lause ja if-lause.
- Näiden lohkojen sisällä luodut paikalliset muuttujat eivät näy lohkon ulkopuolelle.
- Lohkoja voi luoda itse lisää.



# Esimerkkejä Riderissa



# Muuttujien näkyvyys ja olioviitteet

- Muuttujan näkyvyydellä on kytkentä parametrin välitykseen:
- Funktiolle voidaan antaa argumenttina
  - arvopohjaisen muuttujan arvo, tai
  - oliomuuttujan arvo, joka siis on viite.



# Esimerkkejä Riderissa