

Toistot



316

Toistot

- Jos tietokone on jossakin hyvä, niin saman asian tekemisessä moneen kertaan
- Kone ei kyllästy, ei valita
- Sitä ei kiinnosta, tehdäänkö asia yhden kerran vai miljoona kertaa



317

Erilaisia toistotapoja

- Määrään perustuva
- Ehtoon perustuva
- "Kaikkien käsittely"



318

For, määrään perustuva toisto

- For-looppia voidaan käyttää kun halutaan suorittaa jokin asia n -kertaa
- Eli johonkin tiettyyn määrään perustuen
 - Viisi kertaa
 - Miljoona kertaa
 - Niin monta kertaa kuin jokin array on pitkä



319

For – Syntaksi

- Varattu sana 'for'
- Sulkujen sisälle, puolipisteellä eroteltuina
 - Alustus
 - Ehto
 - Muutos
- Aaltosulkeiden sisällä (eli for-lohkossa) olevat suoritetaan
- Ehto tarkastetaan jokaisen toiston alussa
 - Jos arvoksi tulee true, lohko suoritetaan

```
for (alustus; ehto; muutos) {
    // sisältö
}
```



320

For, täsmennys

- Sulkujen sisälle määritellään
 - Alustus muuttujalle
 - Ehto, joka määrittää toiston lopetuksen
 - Muuttujan muutos joka toistolla
- Viereinen toisto voidaan lukea
 - Suorita tämä asia niin kauan, kun "i" on pienempi kuin 5. "i":n ensimmäinen arvo on nolla ja sitä kasvatetaan yhdellä joka toiston jälkeen.

```
// alustus; ehto; muutos
for (int i = 0; i < 5; i++)
{
    Console.WriteLine(i);
}
```



321

For – Arrayn looppaaminen

- For-looppia voidaan käyttää arrayn sisällön läpikäymiseen
- Annetaan vain ehdoksi
 - Tee niin kauan kuin "i" on pienempi kuin arrayn pituus

```
int[] nums = { 1, 2, 3 };

for (int i = 0; i < nums.Length; i++)
{
    Console.WriteLine("Arvo indeksissä " + i + " on " + nums[i]);
}
```



322

Loopin lopettaminen kesken suorituksen

- Käyttämällä varattua sanaa 'break' voidaan toistolause lopettaa kesken sen suorituksen
- Breakin kohdalla toiston suoritus lopetetaan heti ja ohjelma jatkaa suoritustaan for-lohkon jälkeen

```
for (int i = 0; i < 5; i++)
{
    Console.WriteLine(i);
    if (i == 3)
    {
        // jos i:n arvo on 3, lopeta looppia
        break;
    }
}
```



323

Lyhenteitä numeroiden käyttöön

- $i = i + 1$
 - $i++$
- $i = i - 1$
 - $i--$
- $i = i + 5$
 - $i += 5$
- $i = i - 5$
 - $i -= 5$
- Nämä voidaan tehdä myös kerto- ja jako-operaatioille



324

While, ehtoon perustuva toisto

- While-loopilla lohko suoritetaan niin monta kertaa kuin määritelty ehto tuottaa tosi-arvon
- Esimerkiksi
 - Lämpötila alle jonkin tietyn
 - Käyttäjä jatkaa arvojen syöttämistä
 - Neljä on suurempi kuin kolme



325

While - Syntaksi

- Varattu sana 'while'
- Sulkujen sisälle looginen lauseke
- Aaltosulkeiden sisällä (eli while-lohkossa) olevat suoritetaan
- Ehto tarkastetaan jokaisen toiston alussa
 - Jos arvoksi tulee true, lohko suoritetaan

```
while (ehto) {
    // toistetaan niin kauan
    // kuin ehto on totta
}
```



326

While - Ehto

- Koodi, joka arvioituu totuusarvoksi
 - true
 - false
- Jos totta, toisto suoritetaan
- Jos epätosi, ohjelma jatkaa suorittamista while-loopin jälkeen

```
true
1 == 1
count < 10
sana != "lopetus"
```



327

While, esimerkki

- Alustetaan apumuuttuja 'count' arvoon 0
- Suoritetaan toistoa niin kauan kuin 'count'-muuttujan arvo on alle 5
- Pitää muistaa kasvattaa 'count' arvoa
 - Ilman muutosta looppia ei päättyisi koskaan, koska 'count' olisi aina pienempi kuin viisi

```
int count = 0;
while (count < 5)
{
    Console.WriteLine(count);
    count++;
}
```



328

For each - kaikkien käsittely

- Foreach-toistoa voidaan käyttää kun halutaan käsitellä kaikki tietyn kokoelman (vaikka arrayn) sisältämät arvot



329

Foreach - syntaksi

- Varattusana 'foreach'
- Sulkujen sisälle
 - Kokoelman sisältävien arvojen tyyppi
 - Muuttujan nimi
 - Varattusana 'in'
 - Käsiteltävän kokoelman nimi
- Lohkossa voidaan käyttää luotua muuttujaa ('item'), joka saa jokaisella toistolla uuden arvon kokoelmasta

```
foreach (tyyppi item in kokoelma)
{
    // käytä item-muuttujaa
}
```



330

Foreach, esimerkki

- Käsitellään kaikki 'nums'-arrayn jäsenet
- Joka kierroksella 'num'-muuttuja saa arvokseen uuden arvon 'nums'-muuttujasta

```
int[] nums = { 1, 2, 3 };
foreach (int num in nums)
{
    Console.WriteLine(num);
}
```



331

Yksittäisen toiston suorituksen keskeytys

- Käyttämällä varattua sanaa 'continue' voidaan yksittäinen toisto lopettaa kesken sen suorituksen
- Continuen kohdalla toiston suoritus lopetetaan heti ja siirrytään loopin alkuun
 - Ja aloitetaan mahdollinen uusi toisto

```
foreach (int num in nums)
{
    if (num == 2)
    {
        continue;
    }
    Console.WriteLine(num);
}
```