

Funktiot

KOODIN UUSIOKÄYTTÖÄ



348

Funktio

- Funktio on sellainen osa ohjelman lähdekoodia, jota voidaan kutsua useaan kertaan, useassa eri paikassa.
- Englanniksi function



349

Metodi

- Funktio, mutta oliomaailmassa
- Englanniksi method



350

Funktio

- Funktioilla pyritään saavuttamaan:
 - Parempi koodin luettavuus
 - Ylläpidettävyys paranee
 - Virheiden esiintymisen riski pienenee



351

Funktio - Perussyntaksi

- Paluuarvon tyyppi
 - void (ei mitään)
 - string, int, bool...
- Funktion nimi
 - Uniikki
 - Kuvaava (mitä tämä koodi tekee)
- Kaarisulut
 - Olennainen osa funktio määrittelyä
- Aaltosulut
 - Kaikki näiden sisällä oleva suoritetaan kun funktiota kutsutaan/suoritetaan
- FunktioNimi()
 - Kutsutaan määriteltyä funktiota

```
paluuArvonTyyppi FunktioNimi()
{
    // koodia
}

FunktioNimi();
```



352

Yksinkertainen funktio

- Määritellään TulostaSana-funktio, joka ei palauta mitään
- Funktio tulostaa aina saman tekstin
- Kutsutaan määriteltyä funktiota

```
void TulostaSana()
{
    string sana = "sana";
    Console.WriteLine("Tulostetaan: " + sana);
}

TulostaSana();
```



353

Argumentit 1

- Funktiolle voidaan antaa sulkujen sisään argumentteja
- Argumentteja käytetään funktiossa kuten muuttujia
- Määritetään argumentin tyyppi
 - Esim. int, string, bool, decimal...
- Määritetään nimi, jolla sitä halutaan käytettävän
- Useammat argumentit erotetaan toisistaan pilkulla

```
paluuArvonTyyppi FunktioNimi(argumentinTyyppi argNimi)
{
    // koodia, joka käyttää argNimi-argumenttia
}
```



354

Argumentit 2

- Muutetaan TulostaSana-funktiota
- Funktiolla on yksi string-tyyppinen argumentti
- Funktio tulostaa argumenttina syötetyn merkkijonon yhdistettynä toiseen merkkijonoon
- Kutsutaan määriteltyä funktiota argumentilla "moi"

```
void TulostaSana(string sana)
{
    Console.WriteLine("Tulostetaan: " + sana);
}

TulostaSana("moi");
```



355

Funktion paluuarvo 1

- Funktio voi palauttaa jonkin arvon
- Jos funktio ei palauta mitään määritellään paluuarvon tyyppi void
- Jos funktio palauttaa jonkin arvon, paluuarvon tyyppi määritellään ennen funktion nimeä
- Varattusana 'return' suorittaa ja palauttaa sen jälkeen kirjoitetun koodin

```
int FunktioNimi()
{
    return paluuArvo;
}
```



356

Funktion paluuarvo 2

- TulostaSanaJaPalautaSenPituus ottaa vastaan yhden string argumentin ja palauttaa kokonaisluvun
- Palautettu arvo otetaan talteen pituusnimiseen kokonaislukumuuttujaan

```
int TulostaSanaJaPalautaSenPituus(string sana)
{
    Console.WriteLine("Tulostetaan: " + sana);

    return sana.Length;
}

int pituus = TulostaSanaJaPalautaSenPituus("moikka");
Console.WriteLine($"Tulostettu sana oli {pituus} merkkiä pitkä");
```



357

Muuttujien näkyvyys

- Funktion sisällä määritellyt muuttujat ovat käytettävissä vain siinä funktiossa, eikä missään muualla
- Näkyvyys, eng. scope
- Viereinen koodi ei ole laillinen C# ohjelma, koska alimmalla rivillä sanan_pituus-nimistä muuttujaa yritetään käyttää ilman, että sitä on määritelty kysytyssä scopessa

```
int TulostaSanaJaPalautaSenPituus(string sana)
{
    Console.WriteLine("Tulostetaan: " + sana);
    int sanan_pituus = sana.Length;

    return sanan_pituus;
}

int pituus = TulostaSanaJaPalautaSenPituus("moikka");
Console.WriteLine($"Tulostettu sana oli {sanat_pituus} merkkiä pitkä");
```



358

Funktion nimeäminen

- sum(a, b)
- getDatabaseRecordById(id)
- calculateProbabilities(events)
- run()



359

DRY

Don't Repeat Yourself

- Jos huomaat kirjoittavasi samaa koodin pätkää toista kertaa, on hyvin mahdollista, että sen voit paketoida funktioksi
 - Muutokset tulevaisuudessa vain yhteen paikkaan
 - Virheiden metsästäminen helpottuu