Ohjelmoinnin perusteet
R0027

5 op

Metodit

Viopen luku 7

Metodit

- Ohjelman monimutkaistuessa päämetodi kasvaa liian suureksi
- Tarkoituksena kokonaisuuden jakaminen osiin
- Metodi on pieni **aliohjelma**
  - Toimenpidekokonaisuus
- Kuin apulainen
  - Tietty tehtävä
  - Tietty nimi
  - Voi kutsua toisia apulaisia
- Metodeita voidaan myöhemmin ryhmitellä luokkiin

Java API:n tarjoamia metodeita

Ei kannata tehdä uudestaan sitä minkä joku toinen on jo tehnyt

PrintStream
- **println(”rivillinen tekstiä”)**

Math
- **random()**

Scanner
- **nextInt()**
- **nextDouble()**

String
- **length()**
- **indexOf**
- **Replace, contains, split ...**

Oma metodi

```
public class Ohjelmani {

   public static void main(String[] args) {

        omaMetodi();
        //jatketaan työskentelyä kutsusta palaamisen jälkeen
   }

   public static void omaMetodi() {
        System.out.println(”Tervetuloa metodiin”);
        System.out.println(”Tämä on oma metodini”);
        System.out.println(”Tervetuloa uudelleen”);
   }


}
```

Lisää Metodien Ominaisuuksia

- Metodien ominaisuuksia ovat:

1. Näkyvyys
2. Parametrit
3. Palautusarvo

Metodin näkyvyys

- Näkyvyys
  - **public**
    - Metodia voidaan kutsua muista luokista ja mistä tahansa muusta ohjelmasta
  - **private**
    - Metodia voidaan kutsua ainoastaan kirjoittamasi luokan sisältä
  - Later when studying Object-Oriented java: protected, package

Metodin parametrit

- Parametrit metodin määrityksessä
  - *public static void* **metodini1()**
  - *public static void* **metodini2(String nimi)**
  - *public static void* **metodini3(int luku, int toinenLuku)**
- Parametrit metodia kutsuttaessa
  - **metodini1();**
  - **metodini2(”Mika”);**
  - **metodini3(54, 23);**

Metodin palautusarvo

- Palautusarvo
  - **void**
    - *Ei palauta mitään arvoa*
    - `System.out.println(”asdf”);`
    - `avaaWebbiselain();`
  - **int**
    - *Palauttaa kokonaisluvun*
    - `lukija.nextInt();`
  - **String**
    - *Palauttaa merkkijonon*
    - lukija.nextLine();
    - munStringini.replace(’a’, ’4’);
- Palautusarvon tallentaminen metodia kutsuttaessa
  - String nimi = lukija.nextLine();
  - int vuoronumero = arvoNumero();

Metodin määrittäminen

näkyvyys ***staattisuus*** **palautusarvo nimi(parametrit) {**
lauseita
}

Esim.

```
public static void main(String[] args) {
        //…
}
private static void tulosta3KrtTervehdys() {
        //…
}
private static String kysyEmail() {
        //…
}

public static int arvoLuku(int mistä, int mihin) {
        //…
}
```

Metodin staattisuus

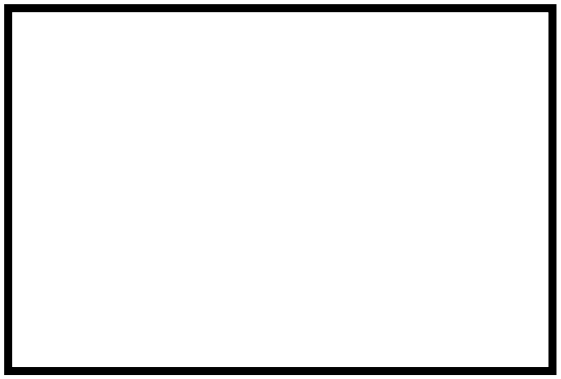
- **static**
  - Metodi on luokkakohtainen eikä kuulu millekään yksittäiselle oliolle
  - Voidaan kutsua suoraan nimensä perusteella

- Vertaa
  - Staattisen metodin kutsuminen
    - luku = **Math.random();** //Math is a class
  - Ei-staattisen eli olion metodin kutsuminen
    - Scanner lukija = new Scanner(System.in);
      luku = **lukija.nextInt();** //Lukija is an object

Oma metodi

- Julkinen näkyvyys
- **Ei oteta vastaan parametreja**
- Ei anneta takaisin paluuarvoa

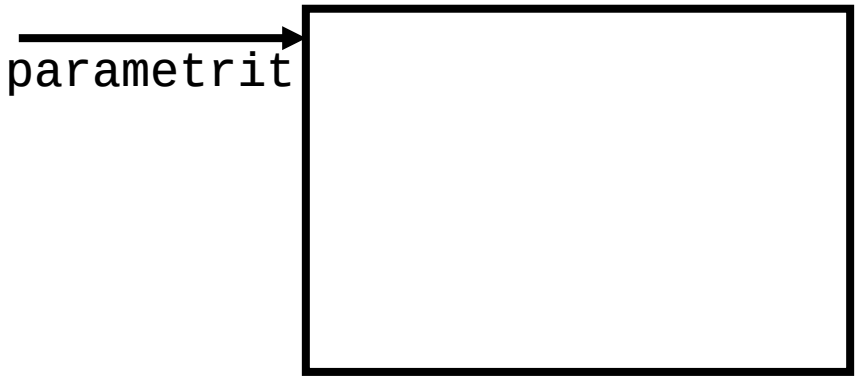
```
public static void tulostaTervehdys() {
        System.out.println(”Tervehdys”);
}
```



Oma metodi parametrillä

- privaatti näkyvyys
- **otetaan vastaan parametri**
- ei anneta takaisin paluuarvoa

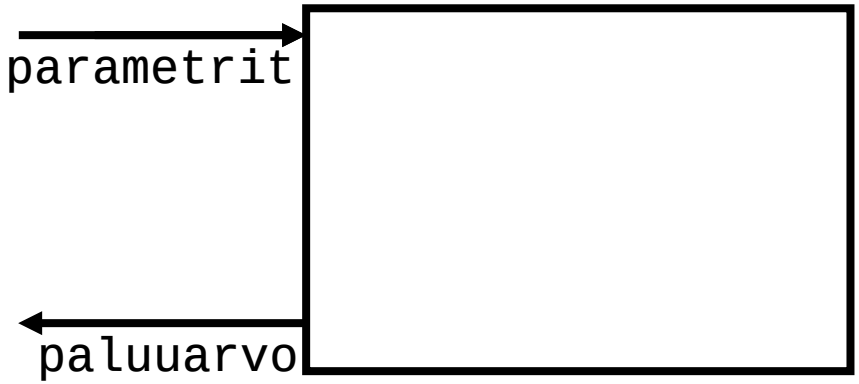
```
private static void tulostaMontaTervehdysta(int lkm) {
        for (int i=0; i<lkm; i++){
                System.out.println(”Tervehdys”);
        }
}
```



Oma metodi parametreillä ja palautusarvolla

- Julkinen
- **Otetaan vastaan parametreja**
- Annetaan takaisin paluuarvo

```
public int laskeSumma(int eka, int toka) {
        int yhteensa;
        yhteensa = eka + toka;
        return yhteensa;
}
```



Method overloading (Metodien kuormitus)

- Metodien kuormituksella tarkoitetaan sitä, että metodeilla voi olla sama nimi, jos niiden parametrit eroavat toisistaan, joko lukumäärän, parametrien tyypin tai molempien osalta.
- Eron on oltava parametreissa, ei paluuarvossa

Metodien kuormitus - esimerkki

```
public class ItseisarvotMetodeilla {

   public static void main(String[] args) {
       int a = itseisarvo(584);
       double b = itseisarvo(-2.1);
   }

   public static int itseisarvo(int luku) {
       //…
   }
   public static double itseisarvo(double luku) {
       //…
   }
}
```