



JYVÄSKYLÄN YLIOPISTO
UNIVERSITY OF JYVÄSKYLÄ

Ohjelmointi 2 - Luento 1

Yleiset tiedot, Java-kielen perusteet

Luennon sisältö



- Henkilökunnan esittely
- Kurssin rakenne ja suorittaminen
- Java-kielen perusteet



JYVÄSKYLÄN YLIOPISTO
UNIVERSITY OF JYVÄSKYLÄ

Kurssin rakenne ja suorittaminen

Kurssin rakenne ja suorittaminen



Opintojaksoilla opit

- oliopohjaisen ohjelmoinnin perusteita ja periaatteita,
- tuottamaan pieniä ja keskisuuria oliopohjaisia ohjelmia,
- graafisen käyttöliittymän suunnittelua ja kehittämistä,
- ohjelman testaamista,
- erilaisia ohjelmoijan työkaluja ja tekniikoita.

Disclaimer: Kurssimateriaalia uudistetaan!

Teemme kokonaisvaltaisen uudistuksen oppimateriaaliin sekä tehtäviin kevään 2026 aikana.

Osa materiaalista julkaistaan kurssin edetessä.

Uudistamisesta johtuen sisällössä voi olla keskeneräisyyksiä ja virheitä.

Ilmoitahan virheistä tai parannusehdotuksista opettajille.

Kurssin rakenne ja suorittaminen

Kurssin sivut ja alustat



- Kurssin kotisivu: <https://ohjelointi2.it.jyu.fi>
- Opettajien sähköposti: ohj2-opet@jyu.onmicrosoft.com
- Opintojakson Teams (liittymiskoodi: nnobn49)
- TIM-kurssialue (vain harjoitustehtävät ja tentti)

Kurssin rakenne ja suorittaminen

Opetus ja ohjaus



- Luennot: <https://ohjelointi2.it.jyu.fi/luennot>
 - Lähtökohtaisesti paikan päällä + livetriimi YouTubessa
 - Nauhoitteet julkaistaan YouTubessa heti ja Moniviestimessä viimeistään seuraavana päivänä
- Ohjaukset: <https://ohjelointi2.it.jyu.fi/#ohjaukset-ja-tuki>
- Kyselytunti: <https://ohjelointi2.it.jyu.fi/luennot#kyselytunnit>

Kurssin rakenne ja suorittaminen

Suorittaminen



Kurssi koostuu 12 osasta:

- Osat 1-6 liittyvät olio-ohjelointiin
- Osat 7-12 painottuvat enemmän harjoitustyöhön

Jokaisessa osassa on:

- kirjallinen oppimateriaali,
- luento oppimateriaalin tueksi,
- harjoitustehtäviä (perustehtäviä ja valinnaisia bonustehtäviä).

Osissa 1-6 harjoitustehtävillä ei ole takarajaa; tekemällä tehtäviä ajoissa voi saada bonuspisteitä (DL-BONUS).

Kurssin lopussa on tentti, josta tiedotetaan tarkemmin myöhemmin.

Suoritustavat: <https://ohjelointi2.it.jyu.fi/suorittaminen>

Java-kielen perusteet

Java-kielen perusteet



- Java on yleiskäytöinen ohjelointikieli alustasta riippumattomien ohjelmien kirjoittamiseen
- Syntaksiltaan lähellä mm. C, C++, C#, D, JavaScript
- Tarvittavat työkalut: <https://ohjelointi2.it.jyu.fi/tyokalut>
 - Java Development Kit (JDK): Javan ohjelmistokehityspaketti
 - Kehitysympäristö (IntelliJ IDEA)

Java-kielen perusteet

Perussyntaksi



```
void main() {  
    IO.println("Hello, World!");  
}
```

Huomaa erityisesti:

- Ohjelman suoritus alkaa `main`-pääohjelmasta
- Aliohjelma alkaa ja loppuu aaltosulkeisiin. Sisennyksellä ei ole merkitystä syntaksin kannalta
- Lause päättyy puolipisteeseen ;

Java-kielen perusteet

Muuttujat ja tietotyypit



- Muuttuja tulee määritellä ennen sen käyttöä
- Määrittelyn jälkeen sijoitus onnistuu yhtäsuuruusmerkillä (=)

```
double muuttujanNimi;  
muuttujanNimi = 10.0;
```

- Monesti määrittely ja ensimmäinen sijoitus yhdistetään samaan lauseeseen

```
double muuttujanNimi = 10.0;
```

Java-kielen perusteet

Muuttujat ja tietotyypit



- Jokaiselle muuttujalle on määriteltävä *tietotyppi*
- Tietotyppi kertoo, mitä tietoa muuttuja esittää ja paljonko muistia se tarvitsee
- Javassa on 8 sisäänrakennettua alkeistietotyyppiä: `byte`, `short`, `int`, `long`, `float`, `double`, `char`, `boolean`

```
// 8-bittinen kokonaisluku
byte tavu = 8;
// 16-bittinen kokonaisluku
short pieniKokonaisluku = 300;
// 32-bittinen kokonaisluku
int kokonaisluku = 200000;
// 64-bittinen kokonaisluku, huomioi L-merkki loppussa
long suuriKokonaisluku = 9000000000L;

// 32-bittinen liukuluku, huomioi f-merkki loppussa
float liukuluku = 5.75f;
// 64-bittinen liukuluku
double tarkkaLiukuluku = 19.99;

// Merkki (16-bittinen), huomioi yksinkertaiset lainausmerkit
char merkki = 'a';

// Totuusarvo
boolean totuusarvo = true;
```

Java-kielen perusteet

Muuttujat ja tietotyypit

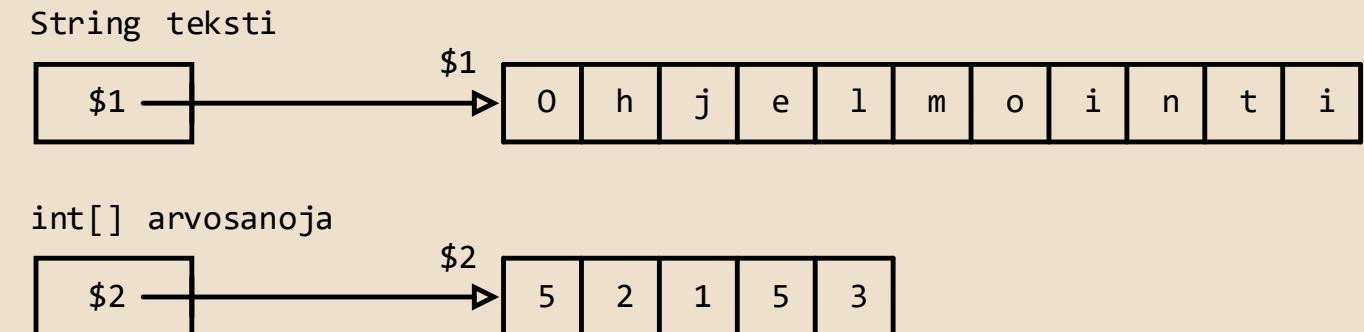


- Javassa kaikki muut tyypit kuin alkeistietotyypit ovat *viitetietotyyppejä*
- Viitetietotyyppi sisältää arvon sijan *viitteen*, jonka avulla ohjelma pääsee varsinaiseen dataan
- Vertaus: muuttuja on “laatikko”; alkeistietotyppillä laatikko sisältää itse arvon, viitetietotyppillä laatikko sisältää “lapun”, joka kertoo, mistä arvo tulee etsiä

```
int luku = 100;
double liukuluku = 1.0;
boolean totuus = true;
```

int luku	double liukuluku	boolean totuus
100	1.0	true

```
String teksti = "Ohjelmointi";
int[] arvosanoja = {5, 2, 1, 5, 3};
```



Java-kielen perusteet

Muuttujat ja tietotyypit



Hyödyllisiä viitetietotyyppejä

- Merkkijono (`String`)
- Muokattava merkkijono (`StringBuilder`)
- Taulukot (`Tyyppi[]`)
- Listat (`ArrayList<Tyyppi>`)
- Alkeistietotyyppien käärijälkuotat
(esim. `Integer`, `Double`)

```
// Huomaa lainausmerkit
String jono = "Opiskelen ohjelmointia!";

StringBuilder muokattava = new StringBuilder("Opiskelen");
muokattava.append(" ohjelmointia!");

// Taulukko: kokoelma samantyyppisiä alkioita
// Huom: luotuun taulukkoon ei voi lisätä alkioita
int[] taulukko = new int[] {12, 42, 16, -2};
taulukko[0] = -1;

// Lista: kokoelma samantyyppisiä alkioita
// "Taulukko, jota voi muokata"
List<Integer> lista = new ArrayList<>(List.of(12, 42, 16));
lista.add(-2);
```

Java-kielen perusteet

Ohjausrakenteet, ehdot



```
int luku1 = 10;
int luku2 = 20;

if (luku1 == luku2) {
    IO.println("Luvut ovat yhtä suuret.");
} else if (luku1 < luku2) {
    IO.println("Luku1 on pienempi kuin Luku2.");
} else {
    IO.println("Luku1 on suurempi kuin Luku2.");
}
```

Java-kielen perusteet

Ohjausrakenteet, ehdot



- Huom! Yhtäsuuruusoperaattori (==) vertaa muuttujien sisältöjä ("mitä laatikossa on")
 - Alkeistietotyypeillä vertaa, ovatko arvot samat
 - Viitetietotyypeillä vertaa, viittaavatko kaksi muuttujaa samaan dataan
- Viitetyppien arvojen vertailu tapahtuu (yleensä) equals-metodilla

```
// "new String" pakottaa uuden viitteen luomisen
String jono1 = new String("OHJ2");
String jono2 = new String("OHJ2");

IO.println(jono1 == jono2); // Tulostaa "false"
IO.println(jono1.equals(jono2)); // Tulostaa "true"
```

Java-kielen perusteet

Ohjausrakenteet, silmukat



```
// while-silmukka
// Soveltuu tilanteisiin, kun toistojen määrää ei
// tiedetä etukäteen
int i = 0;
while (i < 10) {
    IO.println("i = " + i);
    i++;
}
```

```
// for-silmukka
// Soveltuu tilanteisiin, kun toistojen määrä on tiedossa
for (int j = 0; j < 10; j++) {
    IO.println("j = " + j);
}
```

```
// do-while-silmukka
// Vähintään yksi toisto taattu
String salasana;
do {
    salasana = IO.readln("Anna salasana > ");
} while (!salasana.equals("Ohj2"));
```

```
// for-each-silmukka
// Kätevä taulukoiden ja listojen läpikäyntiin
int[] numerot = {10, 20, 30, 40, 50};
for (int numero : numerot) {
    IO.println("Numero: " + numero);
}
```

Java-kielen perusteet

Aliohjelmat



```
// void = ei palauta arvoa
void aliohjelma() {
    IO.println("Aliohjelman runko.");
}

void main() {
    // Aliohjelman kutsu
    aliohjelma();
}
```

```
// Aliohjelma, joka palauttaa int-tyyppisen arvon
int aliohjelma() {
    return 42;
}

void main() {
    // Aliohjelman kutsu ja palautetun arvon sijoitus muuttujaan
    int vastaus = aliohjelma();
    IO.println("Vastaus on: " + vastaus);
}
```

Java-kielen perusteet

Aliohjelmat



```
// Aliohjelma, joka palauttaa String-tyyppisen arvon
// ja ottaa kaksi parametria: String ja int-taulukon
String hienoAliohjelma(String jono, int[] taulu) {
    return jono + taulu.length;
}

void main() {
    String tulos = hienoAliohjelma("Ohjelmointi ", new int[] {1, 2});
    IO.println(tulos);
}
```

Mitä seuraavaksi



- Tutustu kurssin sivuihin
- Liity kurssin Teamsiin
- Asenna tarvittavat työkalut. Kysy tarvittaessa apua ohjauksissa.
- Tee osan 1 harjoitustehtävät

Ensi viikosta alkaen luentotunnit ovat maanantaisin klo 14:15.