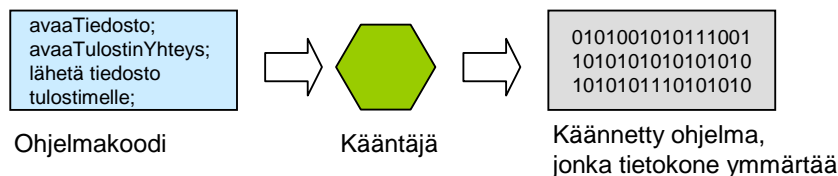


Eri ohjelmointikielissä on erilaisia standardeja siitä, minkälaista ohjelmakoodin on oltava. Java on eräs ohjelmointikieli ja se sisältää tarkat säännöt siitä, minkälaista ohjelmakoodin on oltava, jotta se voitaisiin muuttaa tietokoneen ymmärtämään muotoon.

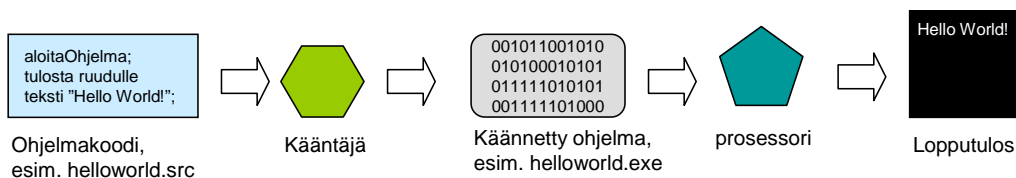
Kääntäjä

Tietokoneiden alkuaikoina ohjelmointi oli sitä, että ohjelmat kirjoitettiin suoraan tietokoneiden prosessoreiden ymmärtämään muotoon. Periaatteessa ohjelmoijat kirjoittivat suoraan ykkösiä ja nollija, jotka voitiin suoraan syöttää tietokoneelle. Kuten arvata saattaa, tämä ei ollut kovin helppoa, ja ohjelmointi vei paljon aikaa.



Ratkaisuksi ongelmaan kehitettiin erilaisia symbolisia ohjelmointikieliä.

Ohjelmointikielten avulla tietokoneohjelmiin voitiin kirjoittaa käskyjä, jotka ovat helpommin ihmisten ymmärrettävässä muodossa. Lisäksi näillä ohjelmointikielillä kirjoitetut ohjelmat voitiin muuttaa automaattisesti tietokoneen ymmärtämiksi ykkösiksi ja nolliksi. Ohjelmasta, joka muuntaa ohjelmoijan kirjoittaman koodin ykkösiksi ja nolliksi, käytetään nimeä kääntäjä (compiler).



Käytännössä kokonaisuudessaan prosessi ohjelmakoodista valmiiksi ohjelmaksi etenee niin, että ensin ihmisen ymmärtämässä muodossa oleva lähdekooditiedosto (esim. helloworld.src) käännetään kääntäjällä tietokoneen ymmärtämään muotoon (esim. helloworld.exe). Tämän jälkeen käännetty ohjelma voidaan antaa tietokoneelle suoritettavaksi. Tietokoneen suorituksen tuloksena voi olla esimerkiksi ruudulle tulostuva teksti "Hello World!". Kun ohjelma on kerran käännetty, ei sitä enää tarvitse käänntää uudelleen, vaan kertaalleen käännetty ohjelma voidaan aina lähettää sellaisenaan tietokoneen suoritettavaksi.

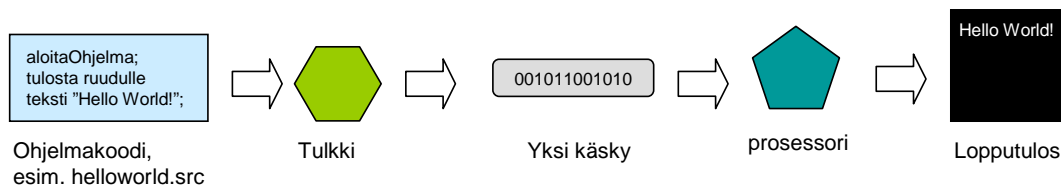
Esimerkki

Liisa on keksinyt nerokkaan algoritmin, joka arpoo lottonumeroita lottoarvontaa varten. Liisa haluaa luoda algoritminsä perusteella tietokoneohjelman, joka luo kerran viikossa Liisalle lottorivin. Käytännössä Liisa kirjoittaa ensin ohjelmansa lähdekoodin

(ohjelmakoodin) ja tallettaa sen johonkin tiedostoon. Tämän jälkeen Liisa kääntää ohjelman kääntäjälle, jonka jälkeen ohjelma on valmiina suoritettavaksi. Tämän jälkeen Liisa suorittaa ohjelman ja tuloksena tietokoneen ruudulle tulostuu seitsemän numeroa, jotka Liisa kirjoittaa lottokuponkiin. Seuraavalla viikolla Liisa haluaa arpoa uuden lottorivin. Nyt Liisan ei tarvitse tehdä muuta kuin suorittaa ohjelman, jonka Liisa viikko loi. Ohjelmaa ei tarvitse enää kääntää uudelleen, kun se on kertaalleen käännetty.

Tulkki

Tulkki (interpreter) toimii periaatteessa samalla tavalla kuin kääntäjä, mutta tekee käännöksen reaaliajassa. Tulkki lukee periaatteessa yhden käskyn lähdekoodista, muokkaa sen tietokoneen ymmärtämään muotoon ja välittää käskyn tietokoneelle suoritettavaksi. Tämän jälkeen tulkki lukee seuraavan käskyn, tekee sille taas muunnoksen ja välittää sen käsiteltäväksi jne.



Tulkki eroaa kääntäjästä siinä, että ohjelmaa ei missään vaiheessa talleteta käännettyssä muodossa tiedostoon samalla tavalla kuin kääntäjän kanssa toimittaessa, vaan käännoistä tehdään periaatteessa sitä mukaa kun ohjelmaa suoritetaan.

Esimerkki

Liisa on keksinyt nerokkaan algoritmin, joka arpoo lottonumeroita lottoarvontaa varten. Liisa haluaa luoda algoritminsä perusteella tietokoneohjelman, joka luo kerran viikossa Liisalle lottorivin. Tällä kertaa Liisa kirjoittaa ohjelmansa niin, että ohjelman suorittamiseen käytetään tulkkiä. Liisa kirjoittaa ensin ohjelmansa lähdekoodin (ohjelmakoodin) ja tallettaa sen johonkin tiedostoon. Liisa antaa tämän jälkeen ohjelmakoodinsa syötteeksi tulkille, joka käy sen läpi rivi kerrallaan, muuttaa lähdekoodissa olevat käskyt tietokoneen ymmärtämään muotoon ja antaa käskyt tietokoneelle suoritettavaksi. Lopputulos on sama kuin kääntäjän kanssa toimittaessa, eli Liisa näkee ruudulla seitsemän numeroa. Kun Liisa haluaa seuraavalla viikolla taas arpoa numerot, antaa Liisa uudestaan kirjoittamansa ohjelman tulkille. Tulkki käy jälleen lähdekoodia läpi rivi kerrallaan, tekee muunnoksen tietokoneen ymmärtämään muotoon ja välittää käskyt suoritettavaksi.

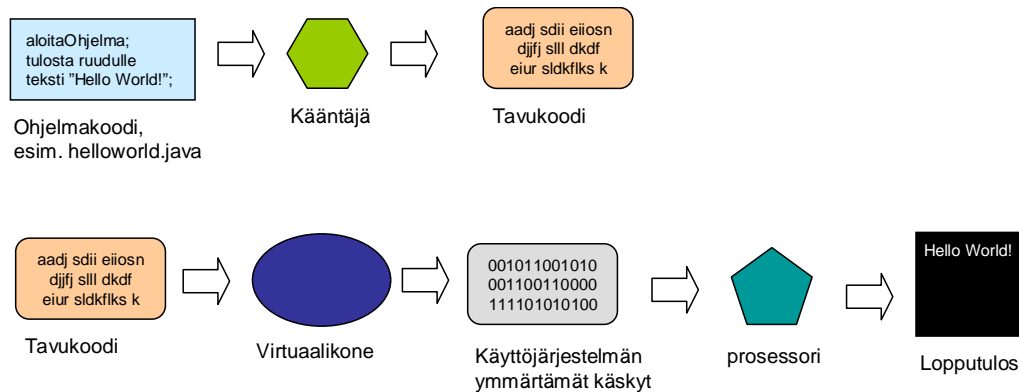
Kääntäjää ja tulkkiä voi verrata hyvin normaalin puhutun kielen kääntämiseen ja tulkkaamiseen. Kääntämisessä esimerkiksi englanninkielinen teksti käännetään suomenkieliseksi. Lopputuloksena on papereita, joissa lukee teksti suomeksi. Tulkkauksessa taas tulkki kuuntelee englanninkielistä tekstiä, kääntää tekstin omassa päässään suomeksi, ja puhuu saman tekstin suomenkielisenä. Suomenkielistä käännettyä tekstiä voidaan hyödyntää sellaisenaan uudelleen niin monta kertaa kuin halutaan ilman, että tekstiä tarvitsee kääntää uudelleen. Sen sijaan tulkkaus pitää tehdä aina uudelleen, jos englanninkielinen tulkattu teksti halutaan suomeksi.

Osa ohjelmointikielistä on käännettyjä kieliä, osa taas tulkattuja. Molemmilla lähestymistavoilla on omat hyvät ja omat huonot puolensa. Tässä vaiheessa on tärkeintä ymmärtää, mitä käsitteet tulkkaus ja kääntäminen tarkoittavat ja mikä niiden ero on.

Virtuaalikone

Kun ohjelmakoodi käännettään, sitä ei oikeataan käännetä suoraan prosessorin ymmärtämään muotoon, vaan pikemminkin käyttöjärjestelmän ymmärtämään muotoon. Käyttöjärjestelmä vastaa vasta varsinaisesta eri käskyjen välittämisestä prosessorille. Tästä on seurauksena käytännössä se, että käännetyt ohjelmat toimivat vain sen käyttöjärjestelmän kanssa, jolle ne on käännetty. Jos ohjelma on esimerkiksi käännetty Windows-käyttöjärjestelmälle, se ei toimi Linux-käyttöjärjestelmän kanssa.

Java-ohjelmointikieli pyrkii käyttöjärjestelmäriippumattomuuteen. Tavoitteena Javassa olisi, että kerran käännetty Java-ohjelma toimii millä tahansa käyttöjärjestelmällä. Käytännössä tähän päästään virtuaalikoneen avulla.



Virtuaalikoneen ajatuksena on, että se ottaa vastaan tavukoodimuodossa olevan ohjelman ja tulkaa sen käyttöjärjestelmän ymmärtämissä käskyiksi. Tämän jälkeen käyttöjärjestelmä välittää käskyt suoritettavaksi prosessorille, joka huolehtii lopputuloksesta, esimerkiksi tekstin tulostumisesta ruudulle. Tällä ratkaisulla päästään siihen lopputulokseen, että ohjelmoijan ei oikeastaan tarvitse tietää mille käyttöjärjestelmälle ohjelma pitäisi kirjoittaa ja kääntää. Ohjelmoijan vastuulla on ainoastaan kääntää ohjelma tavukoodiksi, joka on virtuaalikoneen ymmärtämä muoto. Nyt ohjelmoijan kerran kirjoittamaa ja kääntämää ohjelmaa voidaan käyttää millä tahansa käyttöjärjestelmällä, jolle on saatavilla virtuaalikone.

Virtuaalikoneen käyttämisellä on monia hyötyjä. Ensimmäinen on luonnollisesti se, että ohjelmoijan ei tarvitse tuntea eri käyttöjärjestelmiä ja niiden erikoispiirteitä. Riittää, että ohjelmoija tuntee Java-kielen. Ohjelmia ei lisäksi tarvitse jaella lähdekoodimuodossa, vaan ne voidaan kääntää ja jaella käännettyssä muodossa. Lisäksi kun ohjelma on kerran kirjoitettu, sitä ei tarvitse kääntää enää uudelleen, vaan ohjelmaa voidaan sellaisenaan käyttää millä tahansa käyttöjärjestelmällä, jolle on saatavilla virtuaalikone.

Jos asiaa tarkastellaan kääntäjien ja tulkkien näkökulmasta, Java on sekä käännetty että tulkattu kieli. Java-ohjelma käännettään aluksi, mutta käännettyä ohjelmaa ei

ole vielä valmis käyttöjärjestelmän ymmärtämä ohjelma, vaan eräänlainen välimuoto, josta käytetään nimitystä tavukoodi. Tämän jälkeen kun tavukoodi-muodossa oleva ohjelma suoritetaan, suorittaa virtuaalikone tulkkauksen, eli muuttaa suorituksen aikana tavukoodi-muodossa olevat käskyt sen käyttöjärjestelmän ymmärtämiksi käskyksi, jolla ohjelmaa suoritetaan.

Esimerkki

Liisa on keksinyt nerokkaan algoritmin, joka arpoo lottonumeroita lottoarvontaa varten. Liisa haluaa luoda algoritminsa perusteella tietokoneohjelman, joka luo kerran viikossa Liisalle lottorivin. Liisa päättää kirjoittaa ohjelmansa Javalla.

Liisa kirjoittaa ensin ohjelmakoodin, tallettaa sen tiedostoon ja kääntää tiedoston Javan kääntäjälle tavukoodiksi. Tämän jälkeen Liisa antaa tavukoodimuodossa olevan tiedoston virtuaalikoneelle, joka käy tavukoodissa olevia käskyjä läpi yksi kerrallaan, tulkaa käskyt käyttöjärjestelmän ymmärtämään muotoon ja välittää käyttöjärjestelmälle suoritettavaksi. Lopputuloksena on ruudulle tulostetut seitsemän lukua. Seuraavalla viikolla Liisa haluaa arpoa uuden lottorivin. Nyt Liisan ei tarvitse enää kääntää ohjelmaansa uudelleen, vaan Liisa voi taas antaa tavukoodimuodossa olevan ohjelman virtuaalikoneelle. Virtuaalikone tulkaa jälleen käskyt käyttöjärjestelmän ymmärtämään muotoon ja välittää käskyt käyttöjärjestelmälle suoritettavaksi. Tuloksena on jälleen ruudulle tulostuvat seitsemän numeroa. Vaikka Liisa olisi kirjoittanut ohjelman alun perin Windows-käyttöjärjestelmällä, voi Liisa suorittaa ohjelman seuraavalla viikolla esimerkiksi Linux-käyttöjärjestelmässä ilman, että Liisan tarvitsee kääntää ohjelmaa uudelleen.

Kertaus: tarkista osaamisesi

Tässä osuudessa käytiin läpi yleisesti ohjelmoinnin teoriaa ja peruskäsitteitä tavoitteena ymmärtää, mitä tietokoneohjelman kirjoittaminen yleisesti karkealla tasolla tarkoittaa. Opiskeltuasi tämän kokonaisuuden sinun tulisi osata vastata seuraaviin kysymyksiin:

- Mistä eri vaiheista tietokoneohjelman luominen koostuu?
- Mitä ohjelmointikielet ovat ja miksi niitä tarvitaan?
- Mitä kääntäjä ja tulkki ovat ja mikä niiden olennaisin ero on?
- Mikä virtuaalikone on ja mikä sen suhde on kääntäjään ja tulkkiin?

Jos et keksinyt vastausta johonkin kysymykseen, kouluttaja selittää mielellään asiaa tarkemmin. Saat nopeimmin vastauksen kysymykseesi kirjoittamalla sen keskusteluryhmään.