

Ohjelmointi 1 / syksy 2007

”Mieleni minun tekevi” (tavoiteasettelua)

Paavo Nieminen

nieminen@jyu.fi

Tietotekniikan laitos

Informaatioteknologian tiedekunta

Jyväskylän yliopisto

Tämän luennon teemat

- Tavoiteasettelu esimerkkiohjelmien kautta: millaisia ohjelmia pitäisi voida tehdä Ohjelmointi 1:ssä hahmottuvien taitojen pohjalta.
- Millaisia ohjelmia voi olla? Digitaalinen järjestelmä mustana laatikkona. Rajapinnan käsite, syöte-vaste -käyttäytyminen, tapahtumapohjaisuus vs. eräajo.
- Suoritusympäristö: argumentit, syötevirta, tulostusvirta, käyttöjärjestelmä, tietokone ("digitaalinen järjestelmä")
- Syntaksi eli kielioppi käsitteenä. Javan syntaksista.

Luennon käsittelyjärjestys

1. Perustyökalujen ja toimintaympäristön esittely:
 - Tiedostojärjestelmä, tiedostot, hakemistot
 - Tekstieditori
 - komentokehote
 - Unix-shell
 - Javan komentorivikääntäjä
 - Javan virtuaalikone
2. Demonpalautusjärjestelmän esittely (koskee suomenkielisiä yliopisto-opiskelijoita; muille ohjeistetaan erikseen).
3. Muut asiat siinä määrin kuin ehditään. (Jatketaan huomenna siitä mihin jää.)

Esimerkkiohjelmia

- Tempolaskuri
- Mandelbrot -fraktaalien piirtäjä
- Kirjoitusnopeustesti
- Termiluettelon WWW-julkaisija
- Kivi-paperi-sakset -peli

Tempolaskuri

- Idea: Muusikko voi taputtaa tempoa, ja ohjelma arvioi mikä on musiikkikappaleen BPM (eli tempo neljäsosanuotteina minuutissa)
- (→ Näytetään videoscreeniltä, miten ohjelma käännetään ja ajetaan)

Tempolaskuri ja tämä kurssi

Ohjelmointi 1:llä on tarkoitus oppia:

- Kaikki tarvittavat ohjelmoinnilliset rakenteet
- Kyky siirtää alkuperäinen toimintaidea ohjelmointikielelle
- Kyky löytää tarvittavat lisäominaisuudet Javasta (tässä tapauksessa millisekuntikello) ja kyky oppia käyttämään niitä dokumentaation avulla.

Ohjelmointi 1:llä ei varsinaisesti opita:

- Ei opita mitään listaa millisekuntikelloista ja miljoonasta muusta asiasta, mitä esim. Javalla voi tehdä vaan opitaan löytämään aina se mitä kulloinkin tarvitaan.

Mandelbrot -visualisointi

- Idea: Halutaan tuottaa kaunis visualisointi eräästä tunnetusta matemaattisesti määritellystä kompleksilukujen joukosta
- (→ Näytetään videoscreeniltä, miten ohjelma käännetään ja ajetaan, ja nähdään tulostettu kuva kuvienkatseluohjelmalla)

Mandelbrot -visuaali ja tämä kurssi

Ohjelmointi 1:llä on tarkoitus oppia:

- Kaikki tarvittavat ohjelmoinnilliset rakenteet
- Kyky siirtää kirjallisuudesta löytyvä matemaattinen algoritmi ohjelmointikielelle
- Keinot tiedostojen käsittelyyn: lataaminen ja tallentaminen

Ohjelmointi 1:llä ei varsinaisesti opita:

- matematiikkaa (kompleksilukuja, fraktaaligeometriaa eikä paljon muutakaan)
- varsinaisesti mitään tiettyjä tiedostoformaatteja (kuten tässä PNM-kuvatiedosto)

Kirjoitusnopeustesti

- Idea: Halutaan testata konekirjoitusnopeutta. Pitäisi mahdollisimman nopeasti kopioida näppäimillä joku merkkijono, ja saada tietää montako merkkiä minuutissa on keskimääräinen kirjoitusnopeus.
- (→ Näytetään videoscreeniltä, miten ohjelma käännetään ja ajetaan)

Kirjoitusnopeustesti ja tämä kurssi

Ohjelmointi 1:llä on tarkoitus oppia:

- Kaikki tarvittavat ohjelmoinnilliset rakenteet
- Kyky siirtää alkuperäinen idea ohjelmointikielelle
- Tiedostojen käyttö (kuten tässä kirjoitettavien tekstirivien lukeminen levyltä)

Ohjelmointi 1:llä ei varsinaisesti opita:

- Ei opita listaa millisekuntikelloista tai muusta, mitä esim. Javalla voi tehdä vaan opitaan löytämään aina se mitä kulloinkin tarvitaan.
- Konekirjoitusta tai kalevalaista runomittaa... (vaikka kymmensormijärjestelmästä on iloa ohjelmoinnissa ja nelipolvisesta trokeesta suomalaisen kansallisidentiteetin kannalta.)

Termiluettelon WWW-julkaisija

- Idea: Haluan raapustella kurssilla käsiteltävää terminologiaa nopeasti tekstieditorilla, ja julkaista sanalistan WWW-sivulla.
- (→ Näytetään videoscreeniltä, miten ohjelma käännetään ja ajetaan)

Termiluettelo ja tämä kurssi

Ohjelmointi 1:llä on tarkoitus oppia:

- Kaikki tarvittavat ohjelmoinnilliset rakenteet
- Kyky siirtää alkuperäinen toimintaidea ohjelmointikielelle
- Tiedostojen lukeminen ja kirjoittaminen

Ohjelmointi 1:llä ei varsinaisesti opita:

- WWW:n teknologioita kuten HTML tai CSS

Kivi-paperi-sakset

- Idea: Halutaan aina hävitä todennäköisyydellä $1/3$ ja joutua kirjoittamaan “kivi”, “paperi” tai “sakset” tämän saavuttamiseksi...
- (→ Näytetään videoscreeniltä, miten ohjelma käännetään ja ajetaan)

Ohjelmointi 1:llä on tarkoitus oppia:

- Kaikki tarvittavat ohjelmoinnilliset rakenteet
- Kyky siirtää alkuperäinen toimintaidea ohjelmointikielelle

Ohjelma digitaalisena järjestelmänä

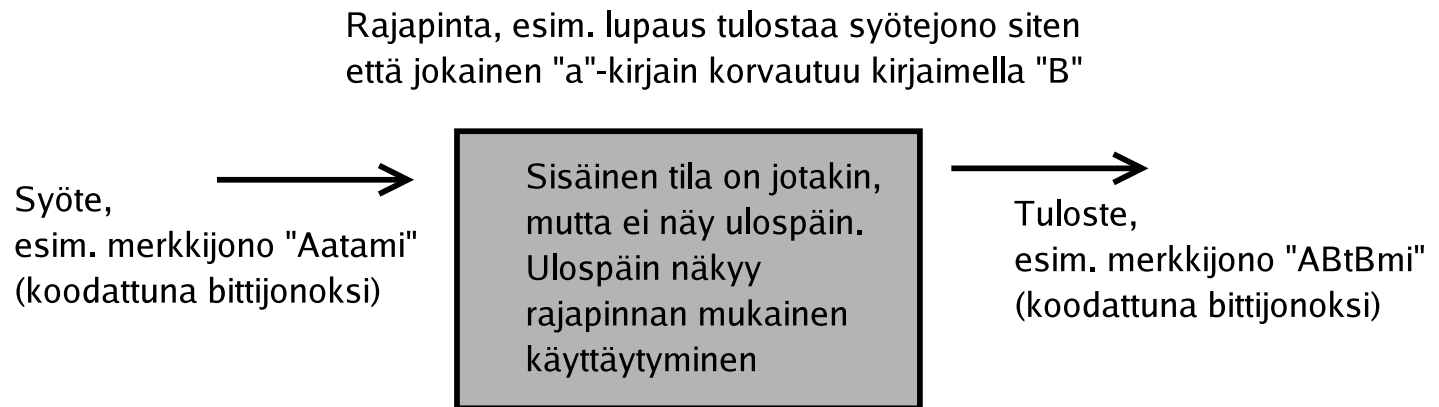
Otetaan aluksi ”ihanan 1920-lukulainen” ajattelumalli tietokoneohjelmasta: Se on digitaalinen (numeroita käsittelevä) järjestelmä, jossa on:

- Sisäinen tila, jota ulkomaailma ei näe
- Kyky vastaanottaa syötteitä (bittijonoja)
- Kyky tulostaa ulkomaailmaan (bittijonoja)
- Käyttäytyminen, eli se miten tuloste (vaste) riippuu syötteestä
- Rajapinta, eli ”lupaus siitä miten järjestelmä käyttäytyy”

Tällaista voi kuvata ”mustalla laatikolla” (kuva seuraavalla sivulla).

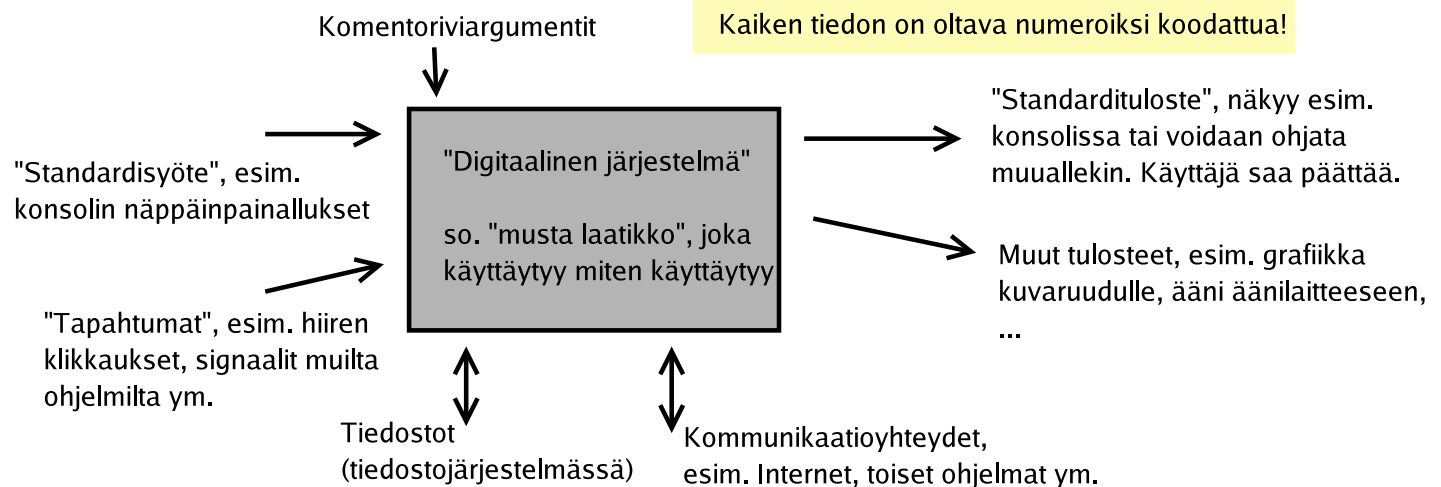
Digitaalinen järjestelmä, ”musta laatikko”

Karkea kuva syötteistä, vasteista ja käyttäytymisestä:



Tietokoneohjelma hieman tarkemmin

Hieman tarkemmin katsottuna tieto kulkee eri kanavia pitkin, ja eri kanavilta tulevien syötteiden ajankohdilla ja järjestyksellä voi olla vaikutus:



Erilaisia ohjelmatyyppejä

Erilaiset ohjelmat yksinkertaisimmasta monimutkaisempaan:

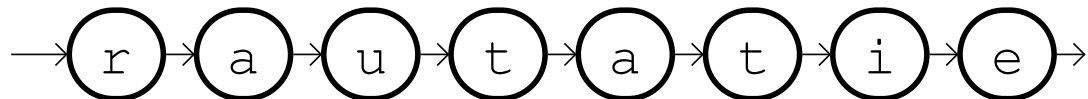
- Tulostavat aina samoin (turhalta kuulostaa)
- Syötteestä riippuva tulostus (esim. kääntäjä!)
- Tulostus riippuu syötteestä ja sen ajankohdasta
- Kommunikoi myös tiedostojen kautta ja tietoliikenneyhteyksillä (Ohj1:llä raja tähän!)
- Sisältää käyttöliittymäkomponentteja ja vahvasti muuttuvaisen, näkyvän tilan (so. graafinen, tapahtumapohjainen ohjelma)
- Järjestelmä, jossa on mukana useita keskenään kommunikoivia erilaisia ohjelmia ja käyttäjiä.

Syntaksi

- Syntaksi eli kielioppi on säännöstö, joka määrää merkkijonojen joukon.
- Syntaksi on välttämätön asia ohjelmoinnissa.
- Ohjelmointikielen syntaksi määrää, onko joku merkkijono ohjelma ollenkaan vai ei.
- Vain kieliopin mukainen ohjelmakoodi on mahdollista kääntää ja suorittaa.
- Kielioppi ei sinänsä kerro mitään merkkijonon merkityksestä!

Rautatie eli syntaksikaavio

rautatie

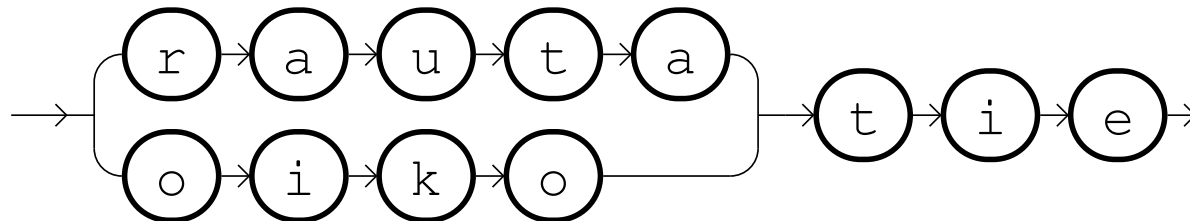


Ajattele rautatietä: Juna lähtee yhdeltä asemalta ja sen tulee päästä perille määräasemalle. Matkalla on asemia, joissa tulee kirjoittaa joku merkki. Kun juna kulkee ylläkuvatus raiteen läpi, on kirjoitettu täsmälleen merkkijono `rautatie`.

Tämä on kielioppi, joka määrää sallittujen merkkijonojen joukoksi tasan yhden sanan eli "rautatie". Ei kovin hyödyllistä vielä ...

Rautatie eli syntaksikaavio 2

rautatie2

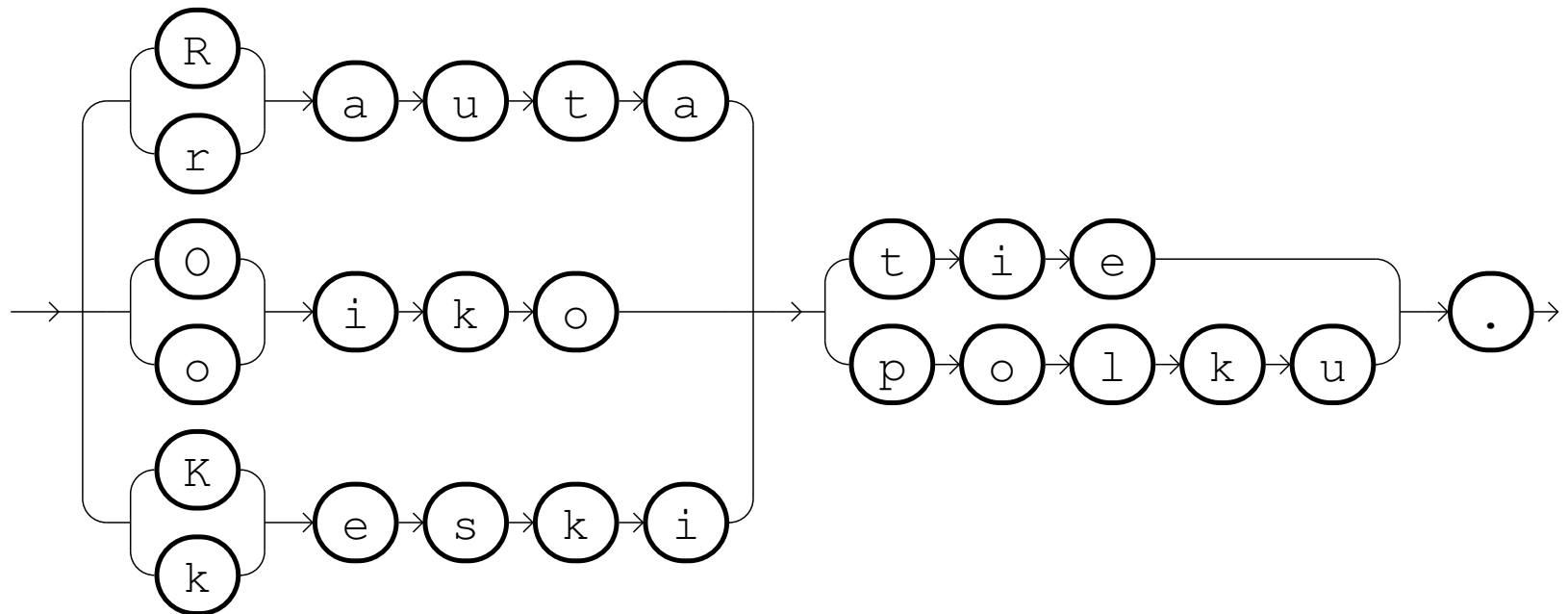


Ajattele edelleen rautatietä: Raidetta voi vaihtaa, jos kiskoissa on sopiva vaihde. Reitti on vapaasti valittavissa, kunhan juna lähtee alusta ja päätyy loppuun. Tämä on kielioppi, joka määrää sallittujen merkkijonojen joukoksi tasan kaksi erilaista: joko `rautatie` tai `oikotie`, riippuen raiteen vaihdosta.

Rautatie eli syntaksikaavio 3

Raiteenvaihtopaikkoja voi olla monta; aina on vapaa valinta. Seuraavassa sallittuja merkkijonoja on jo monta vaihtoehtoa, mm. "Rautatie.", "rautatie.", "Keskitie.", "Oikopolku.", "keskipolku.":

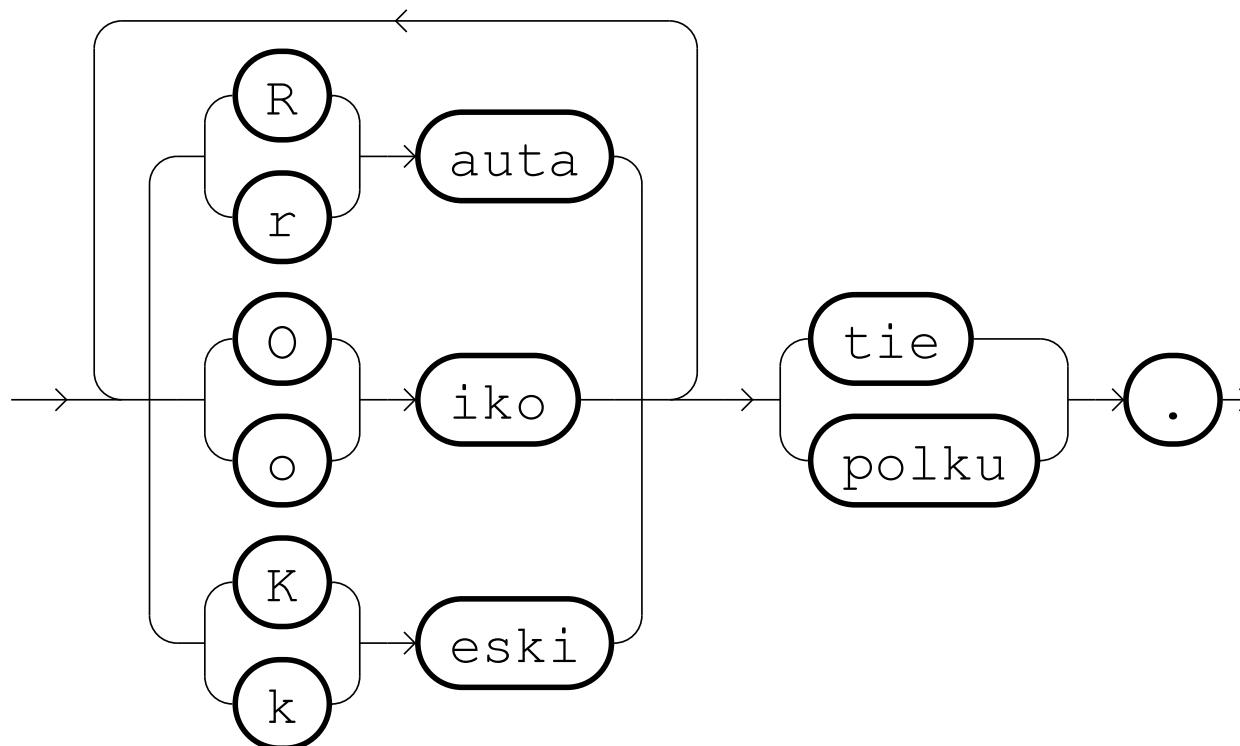
rautatie3



Rautatie eli syntaksikaavio 4

Pakosti peräkkäin tulevat merkit voi kirjoittaa samaan pylpyrään. Raide voi kiertää takaisin alkuun. Seuraavassa sallittuja merkkijonoja on äärettömän paljon, mm. "Keskikeskitite.", "Rautaoikokeskipolku."

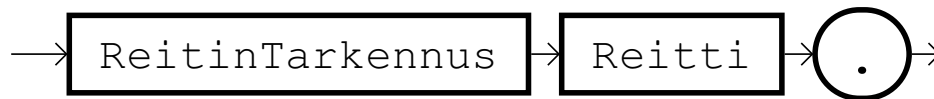
rautatie4



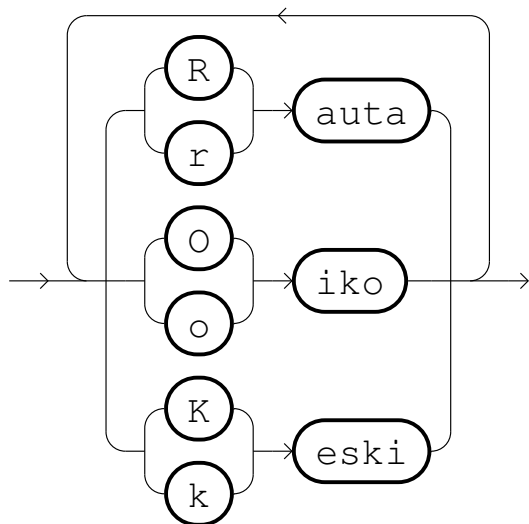
Rautatie eli syntaksikaavio 5

Rautatie voi olla pitkä ja monimutkainen. Sitä kannattaa katsoa lintuperspektiivistä yksi "rataosio" kerrallaan. Laatikko avautuu aina tarkemmiksi ja tarkemmiksi. Vasta pyöreät pylpyrät ovat kirjoitettavia:

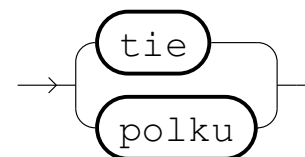
rautatie5



ReitinTarkennus



Reitti



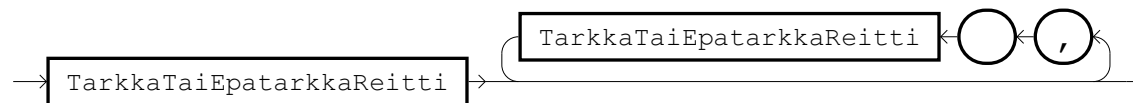
Rautatie eli syntaksikaavio 6

Seuraava mahdollistaa esim. merkkijonon "polku, Rautaoikotie, keskitie, oikoOikopolku, tie, tie, tie."

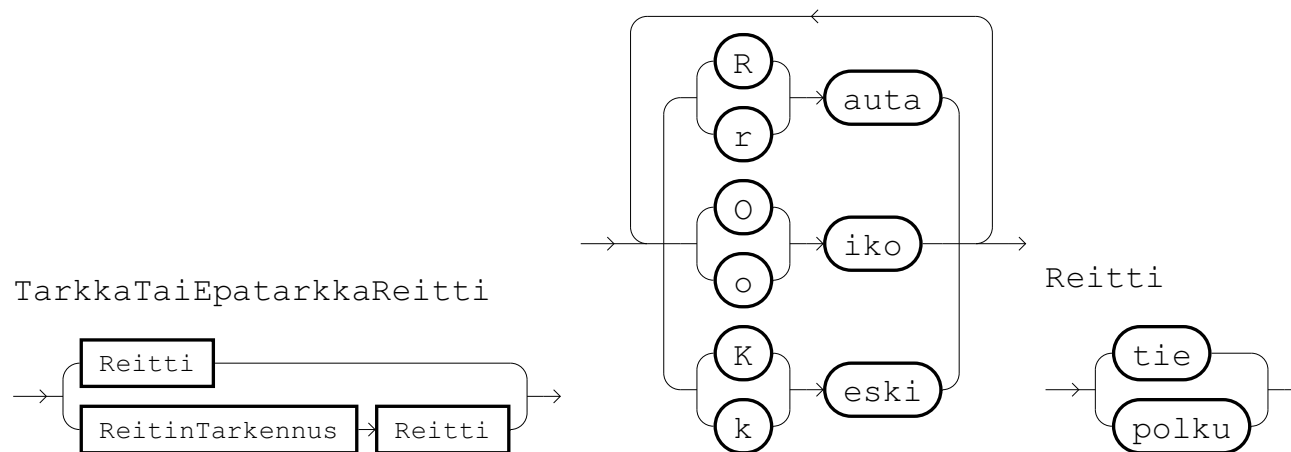
rautatie6



ReittiLista



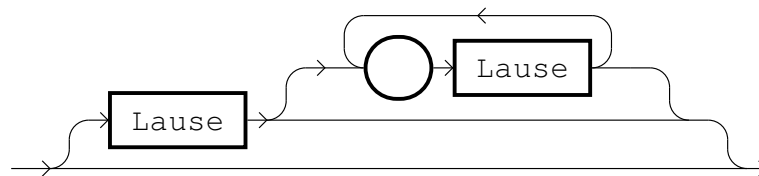
ReitinTarkennus



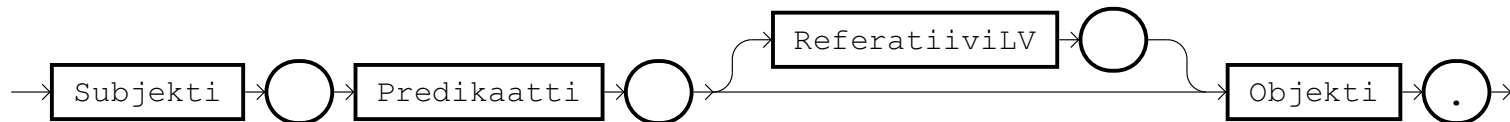
Turha yritys mallintaa suomea

Luonnollinen kieli on eri asia kuin täsmällinen formaali kielioppi, mutta lähestytään vielä arkihavainnon kautta:

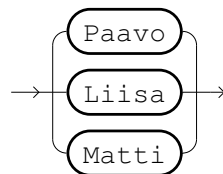
Tarina



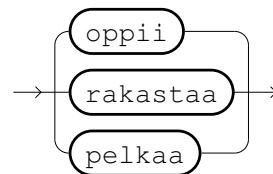
Lause



Subjekti



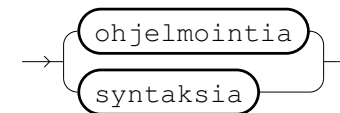
Predikaatti



ReferatiiviLV



Objekti



Millaisen tarinan voi kirjoittaa?

Turha yritys mallintaa suomea

Kelpo, kieliopin mukainen tarina:

- "Liisa pelkää ohjelmointia. Liisa oppii syntaksia. Liisa pelkää oppivansa ohjelmointia. Liisa oppii ohjelmointia. Liisa pelkää rakastavansa ohjelmointia. Liisa rakastaa ohjelmointia."

Toinen tämän kieliopin mielessä täysin kelpo tarina:

- "Paavo pelkää rakastavansa syntaksia. Matti rakastaa oppivansa syntaksia."

(Ei oikein toimi suomalaisen kielikorvaan! Onneksi ohjelmoinnissa ei tarvitse miettiä niin monia sääntöjä. Ohjelmointikielten syntaksit ovat suoraviivaisia!)

Javan syntaksi

Korkeimman tason syntaktinen elementti, eli Javan ”tarina” on nimeltään käännösyksikkö eli Compilation Unit. Yksi lähdekooditiedosto on aina yksi käännösyksikkö! Katsotaanpa miten käännösyksikkö rakentuu, että se on kieliopin mukainen:

CompilationUnit

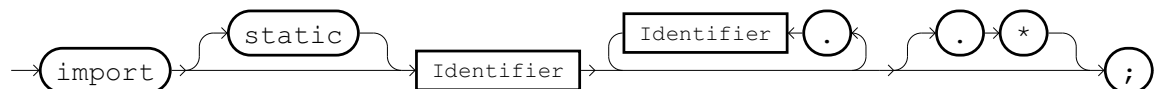


Se on Javan käännösyksikkö! Näin se lähtee avautumaan:

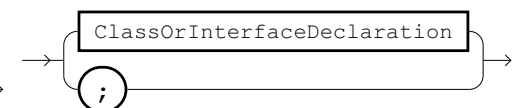
PackageStatement



ImportDeclaration



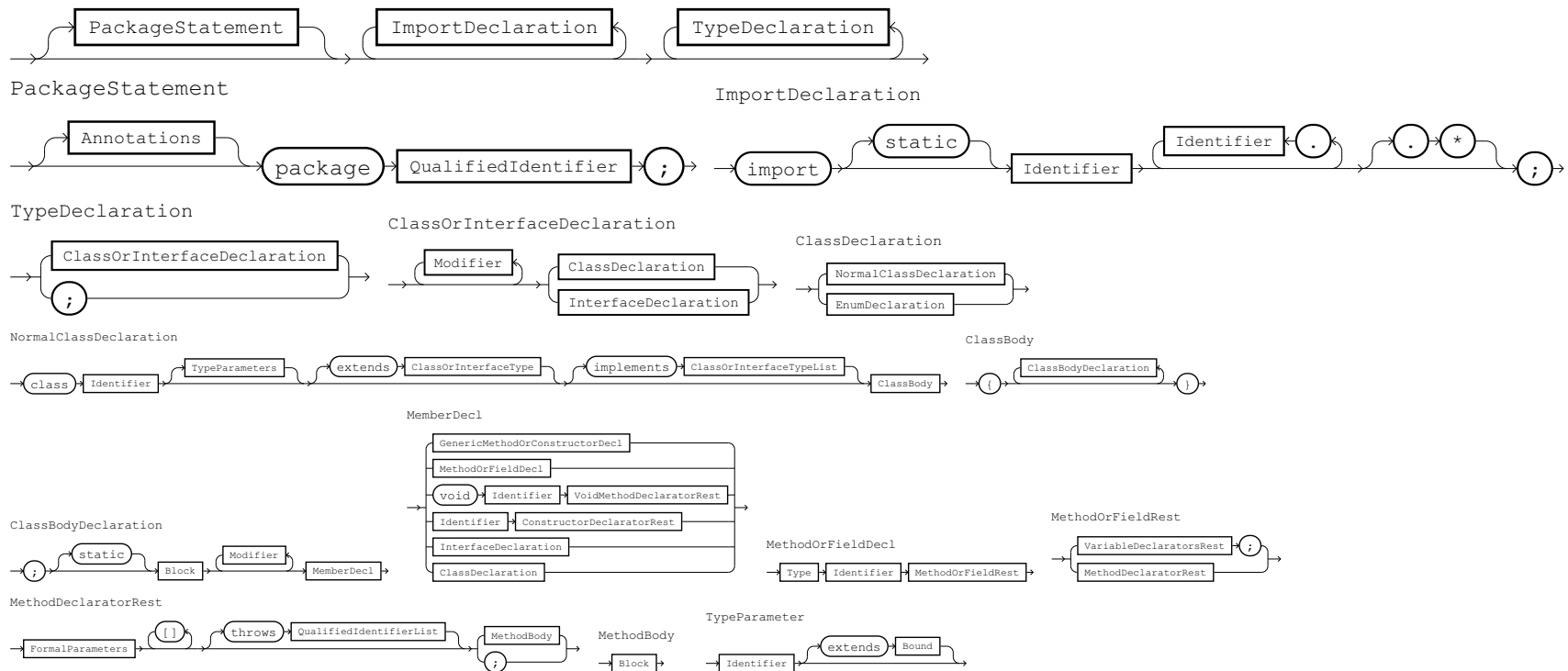
TypeDeclaration



Javan syntaksia pari pykälää lisää

Jokainen toimiva Java-ohjelma rakentuu näin.
Syntaksi avautuu niin suureksi, ettei siitä mahdu luentokalvolle kuin pieni osa:

CompilationUnit



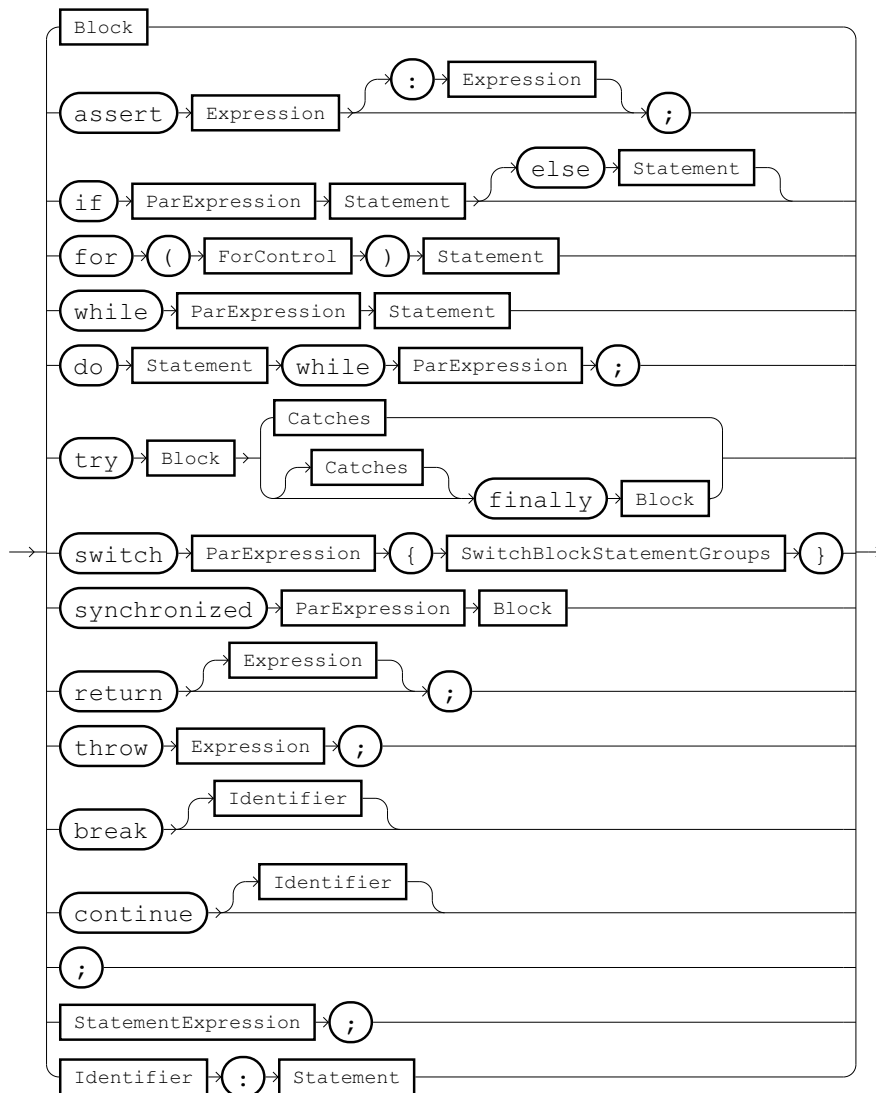
Tarpeellisinta on ymmärtää kieliopin perusidea!

Ohjelmointi on kommunikointia ohjelmointikielellä!

- Jotta voit ohjelmoida, pitää osata ohjelmointikieli (tai useampia). Ensimmäisen opetteleminen on vaikeinta. Seuraava tulee jo melkein itsestään! Kaikilla kielillä voi kuvata tietyt perusrakenteet.
- Ohjelmointikielessä puhutaan peräkkäin kirjoitetuista "lauseista", "määrittelyistä" ja "toteamuksista" aivan kuten esim. suomi.
- Edellämainitut sijoittuvat korkeamman tason rakenteisiin, joita sanotaan olio-ohjelmoinnissa "luokiksi", "rajapinnoiksi" ja "metodeiksi". Suoraan mallinnuskäsitteistä!
- Tällä kurssilla tullaan oppimaan rakenteiden MERKITYS sekä kirjoittaminen Javalla.

Java-kielen lauserakenteet

Statement



(Javassa mahdolliset lauseet (engl. "statement"). Kaikki laatikot avautuvat omalla tavallaan; pylpyrät on kirjoitettava merkistä merkkiin noin.) Voi kirjoittaa vain hyvin tarkoin määrättyjä, melko yksinkertaisia lauseita, mutta niitä peräkkäin laittamalla saadaan aikaan kaikki se, mitä tietokoneilla voidaan tehdä: pelit ja pankkijärjestelmät.

Tarkennuksia ja iskulauseita

- Edellä johdateltiin syntaksiin käsitteenä
- Perusidea on varsin yksinkertainen. Muista rautatie ja raiteet!
- Javan kieliopista on tullut muutosprosessien kautta monimutkainen. Sen syntaksi kannattaa opetella pienenpieni pala kerrallaan!
- Syntaksi ei ole juuri mitään; MERKITYS ON KAIKKI!
- Syntaksivirhe on pienin mahdollinen virhe, ja helppo korjata!
- Haluamme, että ohjelmaan kirjoitettu merkitys on oikeellinen ohjelmointiongelman ratkaisemiseen.

Mihin tästä edetään?

- Ryhdytään ohjelmoimaan
- Opetellaan ohjelmoinnin rakenteet yksi kerrallaan — sekä merkitys että syntaksi.
- Ryhdytään myös hahmottamaan syntaksia tärkeämpiä rajoitteita ohjelman tekemisessä: tiedon esitysmuotoja, suoritusta tietokonelaitteessa ja käyttäjän huomioimista.
- Ryhdytään löytämään tietä ongelmakuvauksesta täsmälliseksi algoritmiksi ja lopulta toteutukseksi ohjelmointikielellä.

Huomenna nähdään!

Silloin aiheena ohjelmoinnin perusrakenteita:

- Peräkkäissuoritus, lauseita ja lausekkeita
- Aliohjelmat ja niiden aktivointi
- Muuttujat, tyyppijärjestelmä, esitysmuodot, operaattorit, sijoituslauseet
- Parametrinvälitys ja paluuarvon sijoittaminen
- Tulostaminen konsoliin