

Johdanto kurssin sisältöön



Turun yliopisto
University of Turku

Terminologiaa

- **Algoritmi** on jonkin tehtävän suorittamiseksi tarvittavien toimenpiteiden kuvaus
- Ongelman ratkaisu lasketaan **suorittamalla** algoritmin toimenpiteet
- **Tietokone** suorittaa sopivalla formaalilla kielellä kuvattuja algoritmeja
- **Ohjelmointikielet** ovat näitä sopivia formaaleja kieliä
- Ohjelmointikielellä kirjoitettu algoritmi on **tietokoneohjelma** eli **ohjelma**
- **Sulautetuissa järjestelmissä** tietokone on ”piilotettu” osaksi elektroniikkaa sisältäviä laitteita

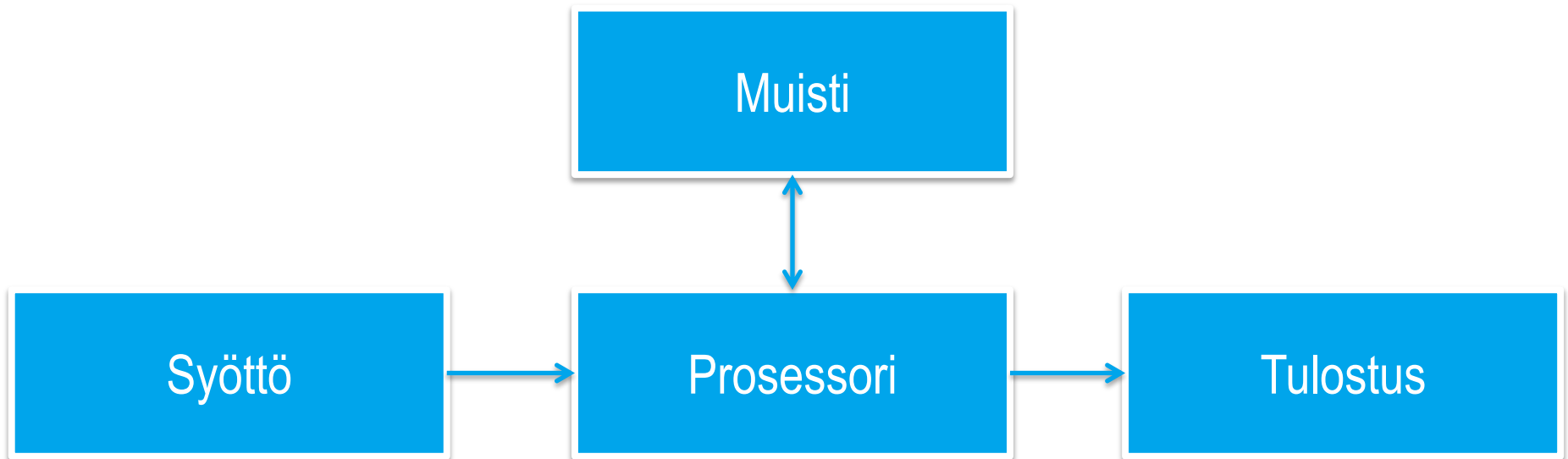


Terminologiaa

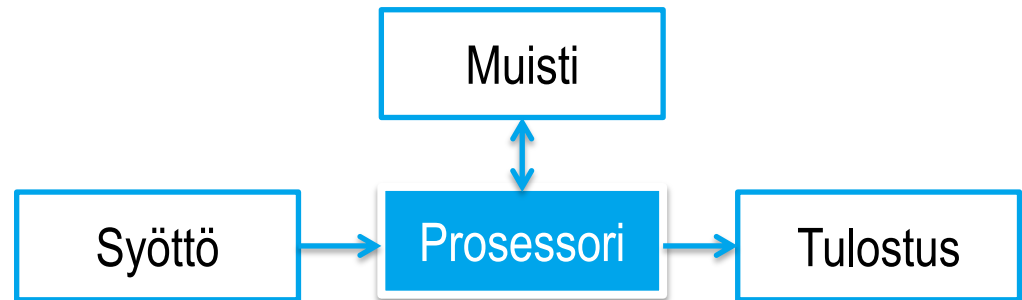
- **Tietojenkäsittelytiede** (computer science) keskittyy **tietojärjestelmän** eksakteihin komponentteihin eli tietokonejärjestelmään
 - Tutkii algoritmeja ja niiden abstrakteja matemaattisia ominaisuuksia matemattis-luonnontieteellisestä näkökulmasta
 - **Tietojenkäsittelytekniikka** (computer engineering) tutkii algoritmeja tietokoneohjelmina, siis tietojärjestelmän osana
- **Tietojärjestelmätiede** (information systems science) laajentaa tutkimuskohdetta hyväksymällä inhimillisestä komponentista johtuvan ”epämääräisyyden”
- Nämä alat yhdessä muodostavat Suomessa **tietojenkäsittelytieteiksi** kutsuttavan kokonaisuuden



Tietokoneen pääkomponentit



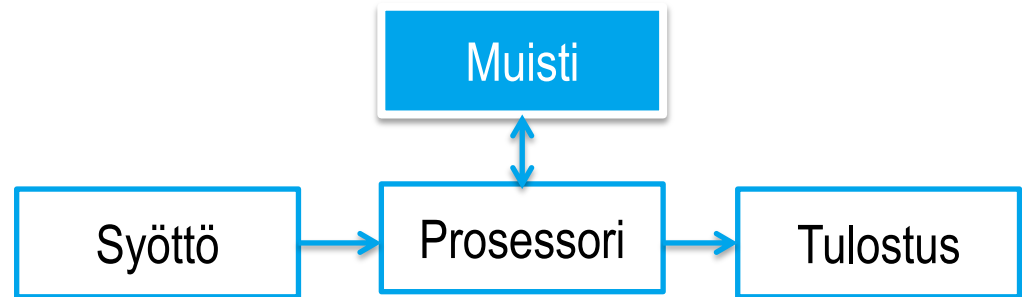
Proessori



- Suorittaa osaamiaan perusoperaatioita
- Proessorin tärkeimmät osat:
 - Aritmeettis-looginen yksikkö (arithmetic and logic unit, ALU)
 - Sisältää dataa käsittelevät ja muokkaavat loogiset piirit
 - Suorittaa esim. laskutoimitukset
 - Rekisterit
 - Sisältävät muistista haettavan datan jota prosessori käsittelee
 - Ovat erityisen nopeita muistipaikkoja
 - Kontrolliyksikkö (control unit, CU) eli ohjausyksikkö
 - Kontrolliyksikön rekisterit sisältävät koneen tarvitsemaa kontrollitietoa
 - Pitää kirjaa missä vaiheessa ohjelman suoritus on



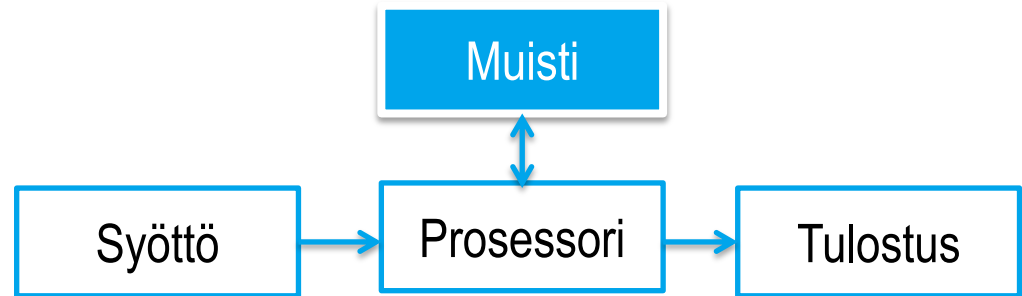
Muisti



- Säilyttää tietoa
- **Keskusmuistissa** säilytetään
 - Suoritettavaa algoritmia
 - Tietoa, johon algoritmin operaatiot kohdistetaan
- **Oheismuisti**
 - Tarvitaan suurten tietomäärien pitkäaikaiseen varastointiin
 - Esim. kiintolevy



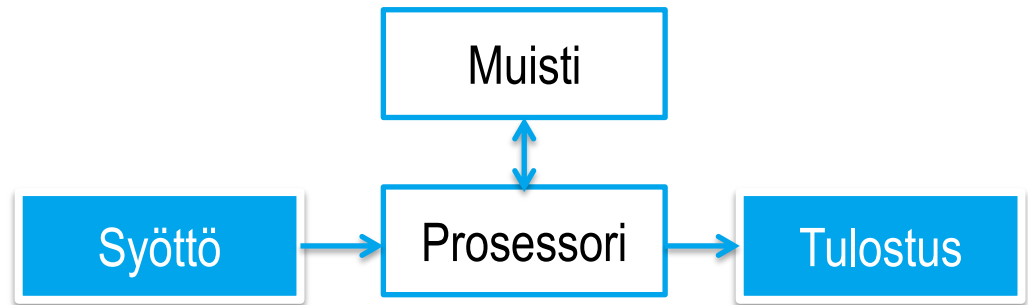
Muisti



- Keskusmuisti jaettu kahteen komponenttiin
 - **Pysyväismuisti** (ROM, Read Only Memory)
 - Sisältää ohjelman, jolla laitteisto alustetaan ja käyttöjärjestelmä ladataan
 - Datan kirjoittaminen hidasta tai mahdotonta
 - **Käyttömuisti** eli **työmuisti** (RAM, Random Access Memory)
 - Sisältää esim. suoritettavan algoritmin ja sen tarvitseman datan
 - Datan kirjoittaminen ja lukeminen nopeaa



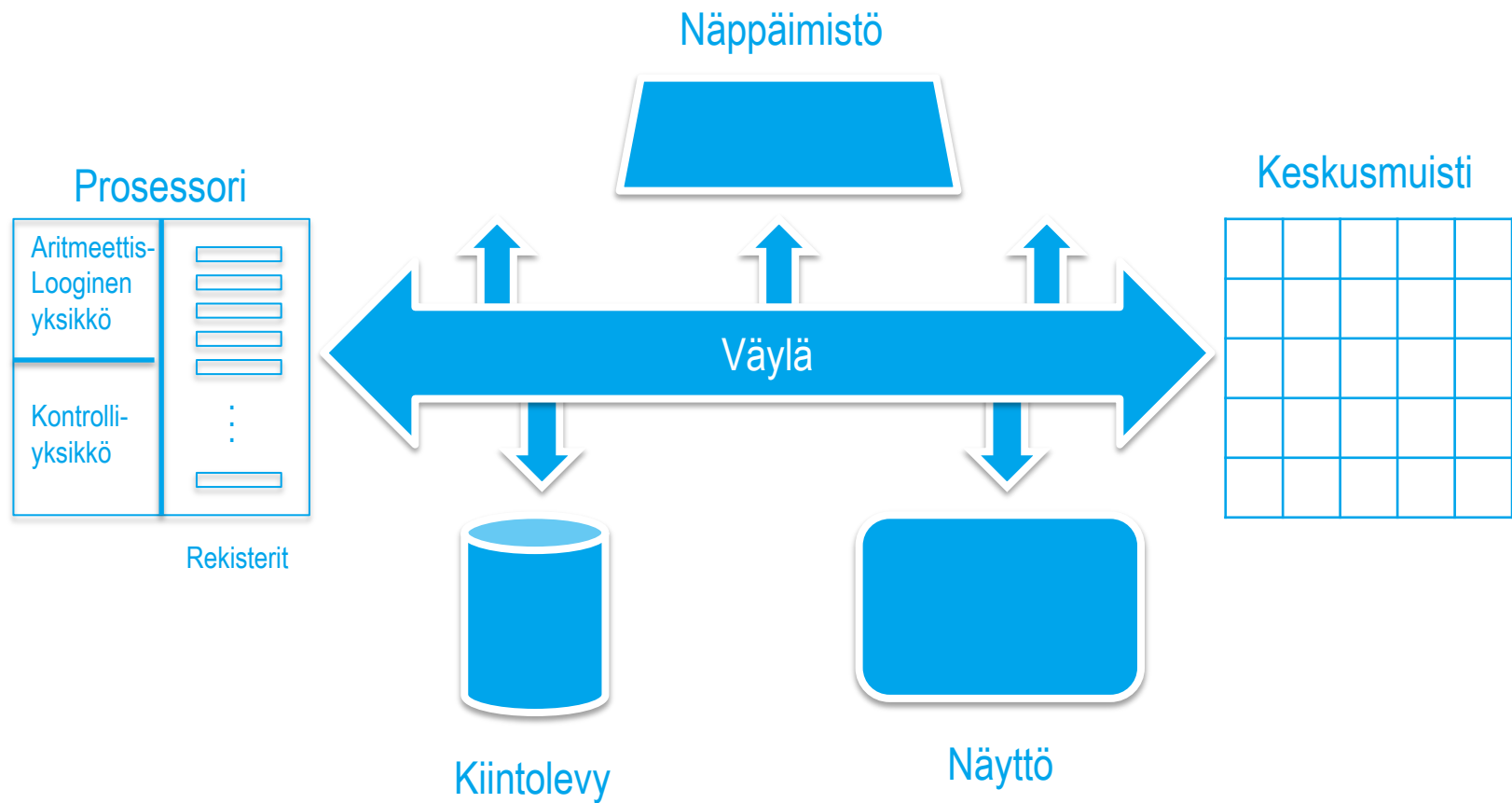
Syöttö & tulostus



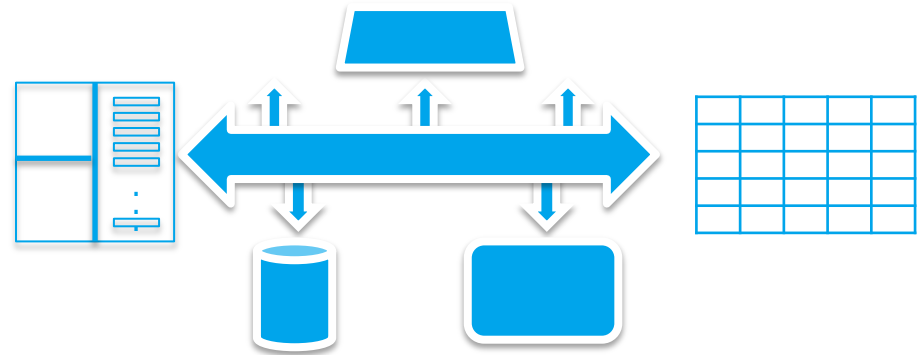
- Syöttölaitteilla
 - Algoritmi ja käsiteltävä tieto syötetään muistiin
- Tulostuslaitteilla
 - Tietokone "tulostaa" käsittelyn tulokset
- Kumpikaan ei ole välttämättömiä tietokoneen toiminnalle
 - Tosin kommunikaatiokyvytön tietokone on melko hyödytön



Väylä



Väylä



- Väylä (bus) on kokoelma johtimia
- Hoitaa tiedonsiirron
- Prosessorin, muistin ja laitteiden välinen data kulkee aina väylän kautta
- Tietokoneen toimintaperiaate (yksinkertaistettuna):
 - Prosessori noutaa suoritettavan ohjelman (konekieliset) käskyt ja tarvittavan datan keskusmuistista
 - Prosessori suorittaa käskyissä määritellyt toiminnot
 - Prosessori siirtää käsitellyn datan takaisin keskusmuistiin (tai muihin muisti-/tulostuslaitteisiin)



Data, tieto ja informaatio

- Usein käytetään synonyymeinä, vaikka tarkoittavat eri asioita
- Data:
 - Asian säännönmukainen esitys viestittävässä tai käsittelykelpoisessa muodossa, joka on koneellisesti luettavissa
- Informaatio:
 - Datan ihmiselle tuottama mielle tai merkitys
- Tieto:
 - Ihmisen ajattelun kohde tai tulos
- Tietokone suorittaaakin siis datan käsittelyä eikä tietojen käsittelyä



Kahden olotilan periaate - bitit

- Tietokoneen toiminta perustuu ***kahden olotilan periaatteeseen***:
 - Kaksi vaihtoehtoa, esim. ”johdossa” on (arvo=1) tai ei ole (arvo=0) jännitettä
- Kaikki tieto tietokoneessa esitetään arvoilla 0 ja 1, eli biteillä
- Kaikki data esitetään bittijonona
 - Kokonaisluvut käyttäen 2-järjestelmän esitystä (binäärilukuja)
 - Desimaaliluvut tallennetaan ”eritavalla” (esim 1.0 tallennetaan eritavoin kuin kokonaisluku 1)
 - Kuvan ja äänen käsittely tietokoneessa vaatii tiedon muuttamisen bittijonoksi (siis 0 ja 1)
- Erilaisiin bittiesityksiin palataan kurssin II osassa



Tietokoneet kaikkialla yhteiskunnassa – miksi?

- Nopeus
 - Tietokoneen prosessori voi suorittaa miljardeja yksinkertaisia operaatioita sekunnissa, aivojen ”laskentanopeus” on paaaljon hitaampi
 - Nopeus ei kuitenkaan yksin riitä, ja monet ihmiselle helpot tehtävät ovat koneille vaikeita (esim. motoriset toiminnot)
 - Monet ongelmat niin vaikeita, ettei tietokoneidenkaan nopeus riitä
- Luotettavuus
 - (Viattomat) tietokoneet toimivat kuten algoritmi kääkee eivätkä tee virheitä
 - Virheet johtuvat yleensä virheistä algoritmeissa (jotka ihminen ohjelmoi)
 - Tietokone ei kykene arvioimaan tulosten järkevyyttä



Tietokoneet kaikkialla yhteiskunnassa – miksi?

- Muisti
 - Tietokoneet kykenevät varastoimaan ”rajattomasti” tietoa, ihmisen muistikapasiteetti on rajallinen
- Kustannukset
 - Tietokoneen käytöllä voidaan saavuttaa merkittäviä kustannussäästöjä verrattuna tehtävien suorittamiseen ihmistyövoimalla



Ohjelmointikielet

- Tietokone ei pysty suorittamaan luonnollista kieltä
- Luonnollisissa kielissä algoritmin kannalta ongelmallista:
 - Laaja sanasto
 - Monimutkaiset kielioppisäännöt
 - Lauseiden merkitys riippuu kontekstista
 - Perustuu viime kädessä puhuttuun kieleen
 - Käyttö vaatii maailmankuvaa
- Algoritmit esitetään siis formaalilla kielellä, eli ohjelmana



Ohjelmointikielet

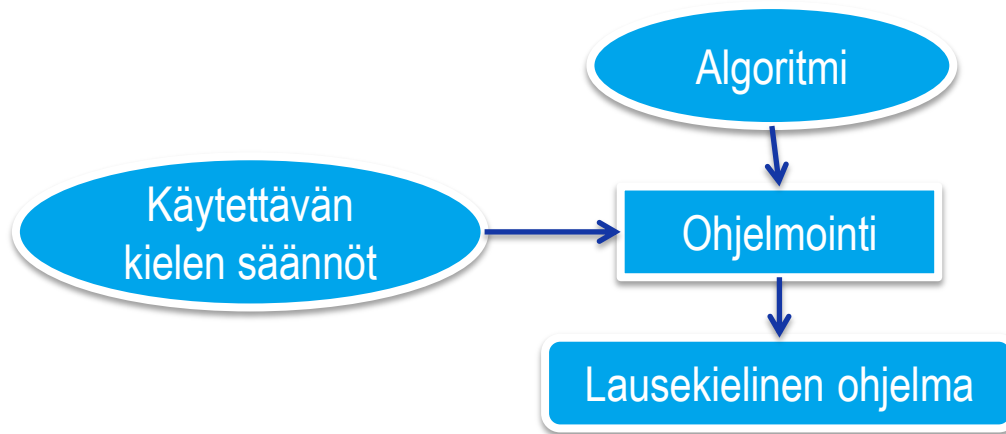
- Ohjelmointikielen suunnittelussa huomioitava
 - Helpotettava algoritmien ilmaisemista tietokoneen ymmärtämässä muodossa
 - Helposti tietokoneen ymmärrettävissä
 - Helposti ihmisen ymmärrettävissä
 - Minimoitava virhemahdollisuudet ohjelmaa kirjoitettaessa
 - Ohjelmaa tutkimalla pitäisi olla helppo vakuuttua sen oikeellisuudesta
- Lukuisia ohjelmointikieliä suunniteltu
- Kurssilla käytetään **pseudokieltä**, jossa oikeista ohjelmointikielistä ovat rakenteet
 - Kurssin lopussa Python-ohjelmointikieltä



Ohjelman kirjoituksesta suoritukseen

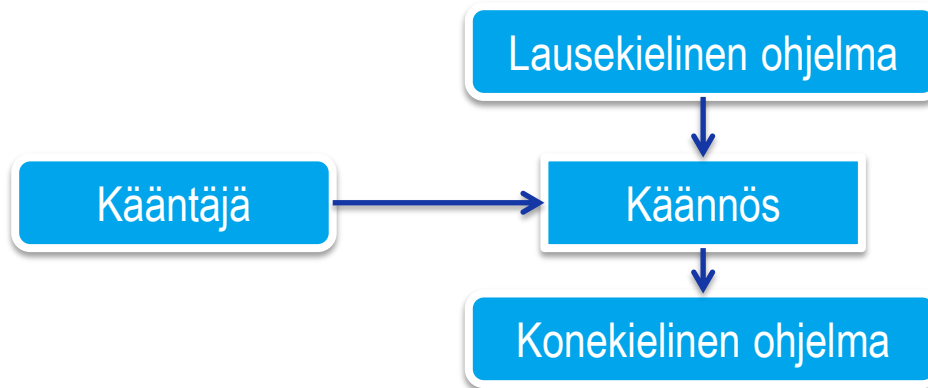
- <https://www.youtube.com/watch?v=GZpcwKEIRCI>





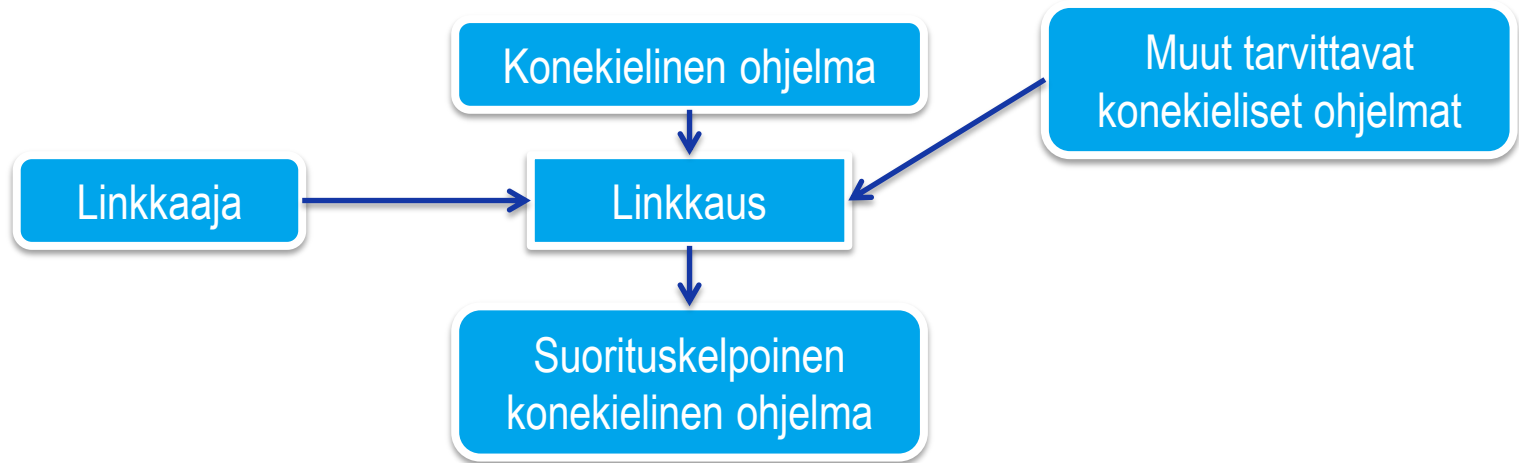
- Ongelman ratkaiseva algoritmi ohjelmoidaan käyttäen jonkin ohjelmointikielen sääntöjä
- Jokainen algoritmin askel esitetään ohjelmassa käskynä, lauseena
- Ohjelma muodostuu lausejonosta, jotka määrittävät tietokoneen suorittamat perusoperaatiot
- Kirjoitettua ohjelmaa kutsutaan usein ***lausekieliseksi ohjelmaksi***
 - Koneista riippumattomia ohjelmointikieliä
 - Suunniteltu ihmisille ”helpoksi” ja ”luontevaksi”
 - Tietokone ei ymmärrä korkean tason lausekieltä suoraan





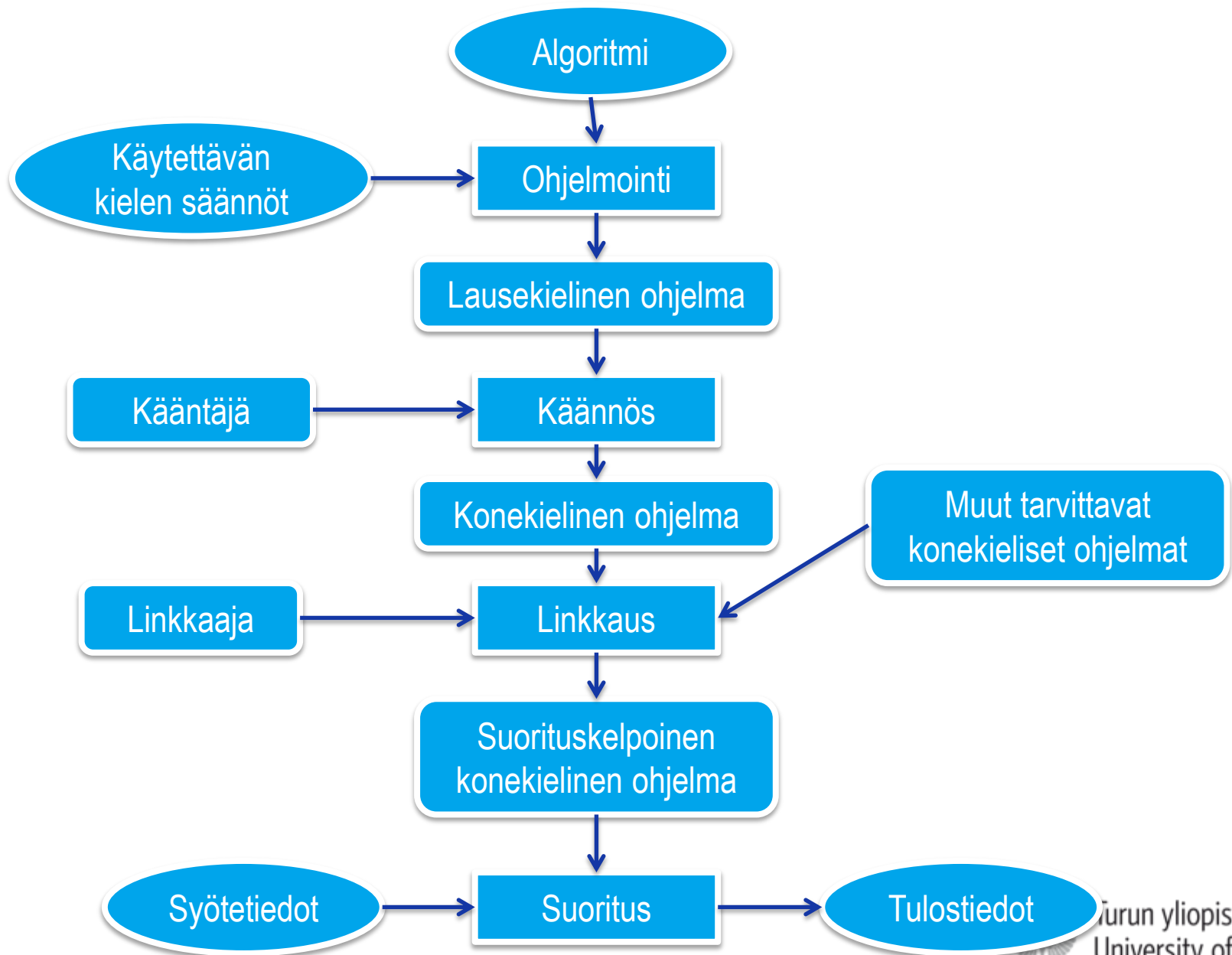
- Kullakin prosessorilla on sille ominainen **konekieli** (machine language)
 - Kieli joka on suunniteltu toimimaan yhdessä prosessorin kanssa
 - Voidaan kuvata vain hyvin primitiivisiä operaatioita, jotka kone osaa suorittaa
 - Konekielellä ohjelmointi on työlästä, mutta lausekielten ansiosta harvoin tarpeellista (aiheeseen palataan kurssin II osassa)
- **Kääntäjä** (compiler) tai **tulkki** (interpreter) muuntaa lausekielisen ohjelman konekieliseksi ohjelmaksi
 - Kääntäjä muuntaa koko ohjelman kerralla, tulkki lause kerrallaan sen suorituksen aikana (aiheeseen palataan kurssin II osassa)





- Käännetty konekielinen ohjelma ei usein ole suorituskelpoinen
- Toimiakseen se tarvitsee muiden jo käännettyjen ohjelmien apua
- Suoritetaan vielä **linkkaus**
 - Liitetään tarvittavat muut ohjelmat käännettyyn konekieliseen ohjelmaan
- Linkkaaja on liittämisen suorittava ohjelma
- Monissa järjestelmissä linkkaus ja suoritus voidaan tehdä yhdellä komennolla





Ohjelmistot

- Tietokoneen ohjelmistot koostuvat
 - **Sovellusohjelmistosta** (application software)
 - Valmisohjelmia ja käyttäjien tekemiä ohjelmia
 - **Systeemiohjelmistosta** (system software)
 - Systeemitason ohjelmia kuten esim. kääntäjät ja tulkit
 - **Käyttöjärjestelmiä**, jotka huolehtivat mm. syöttö- ja tuloslaitteiden hallinnasta ja tietojen pitkäaikaissäilytyksestä

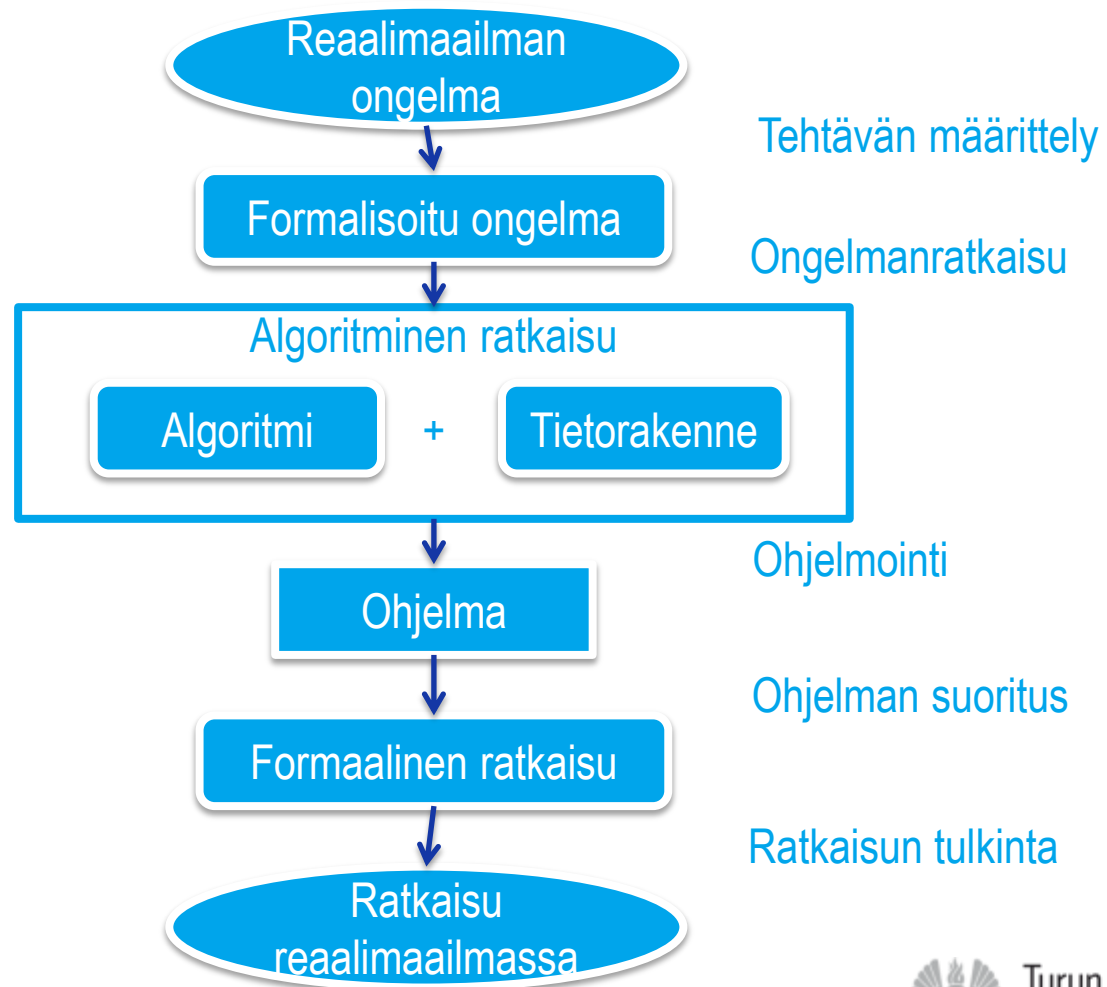


Algoritminen ongelmanratkaisu

- Ongelman ratkaisu algoritmisesti tietokoneella voidaan kuvata algoritmilli:
 1. Esitä algoritmi ohjelmana sopivalla ohjelmointikielellä
 2. Suorita ohjelma tietokoneessa
- Vaihe 2 on usein helppo
- Vaiheen 1 algoritmin muodostaminen ei
 - Vaatii luovuutta ja älyllistä ponnistelua
 - **Ei ole mahdollista kirjoittaa algoritmia algoritmin muodostamiseksi**



Algoritminen ongelmanratkaisu



Algoritmin perusvaatimukset

- Algoritmi
 - On täsmällinen ja yksityiskohtainen kuvaus tai ohje tehtävän suoritukseen
 - Perustuu yksikäsitteiseen ja tarpeeksi täsmälliseen tehtävän määrittelyyn
 - Syötteiden ja tulosteiden tarkka kuvaus tärkeä osa
- Mikä tahansa ohje ei siis ole algoritmi



Algoritmin perusvaatimukset

- Algoritmilta vaadittavat ominaisuudet:
 - **Yleisyys**: algoritmin sovelluttava määritellyn tehtävän kaikkiin tapauksiin
 - **Deterministisyys**: tehtävän ratkaisu on oltava yksikäsitteinen, ja joka vaiheessa tiedettävä mitä seuraavaksi tehdään
 - **Tuloksellisuus**: algoritmin on aina annettava oikea tulos, jaettavissa:
 - **Oikeellisuus**: algoritmin antama tulos on aina oikein
 - **Terminoituvuus**: algoritmin suoritus päättyy, eli se antaa tuloksen



Algoritmien asteittainen tarkentaminen

- Algoritmien suunnittelu usein vaikeaa
 - Yleinen virhe, että algoritmi tuottaa oikean tuloksen, mutta ei juuri annettuun tehtävään -> **varattava riittävästi aikaa tehtävän määrittelylle**
- Algoritmien suunnitteluun kehitetty menetelmiä
- Yksi sellainen on ***asteittain tarkentava menetelmä*** (stepwise refinement)
 - Algoritmi suunnitellaan kokonaisuudesta osiin edeten (top-down design)
 - Jaetaan alkuperäinen prosessi pienempiin osiin, joille voidaan laatia oma algoritminsa
 - Jaetaan alialgoritmit jälleen pienempiin ja pienempiin osiin
 - Riittävän pienet alialgoritmit ratkaistaan
 - Lopullinen ratkaisu yhdistetään alialgoritmeista



Algoritmien asteittainen tarkentaminen

- Algoritmin abstraktiotaso laskee jokaisella tarkentamisella
- Tehtävän ratkaisu suunnitellaan ensin korkealla abstraktiotasolla
 - Konkretisoidaan asteittain, kunnes abstraktiotaso on sopiva suorittajalle (tietokoneelle)
- Ohjelmointi on siis ihmisen korkean abstraktiotason ajattelun muuntaminen koneelle soveltuvalle matalammalle abstraktiotasolle



Algoritmien asteittainen tarkentaminen

- Tarkentava menetelmä liittyy läheisesti reduktiiviseen ongelmanratkaisuun
- **Reduktio**: ongelman jakaminen hyvin määriteltyihin osiin, jotka ratkaistaan erikseen, ja osaongelmien ratkaisusta kootaan koko ongelman ratkaisu
- Lähestymistapaa voidaan myös pitää **modulaarisena**:
 - Kunkin osaongelman ratkaisu kootaan itsenäiseksi osa-algoritmiksi eli algoritmimoduuliksi
 - Modulaarisuudella merkittävä asema algoritmien suunnittelussa
 - Modulaarisuuteen palataan myöhemmissä luennoissa



Algoritmien asteittainen tarkentaminen

Esimerkki: pikakaakaon keittoalgoritmi robotille.

1. Keitä vettä
 2. Pane kaakaojauhoja kuppiin
 3. Kaada vettä kuppiin
- Ei tarpeeksi tarkka robotin suoritettavaksi -> tarkennetaan



Algoritmien asteittainen tarkentaminen

Esimerkki: pikakaakon keittoalgoritmi robotille.

1. Keitä vettä
 1. Täytä vesipannu
 2. Aseta pannu liedelle
 3. Lämmitä kunnes vesi kiehuu
 4. Siirrä pannu liedен sivuun
2. Pane kaakaojauhoja kuppiin
...
3. Kaada vettä kuppiin
...
- Edelleen kaipaa tarkennusta -> tarkennetaan



Algoritmien asteittainen tarkentaminen

Esimerkki: kaakaon keittoalgoritmi robotille.

1. Keitä vettä
 1. Täytä vesipannu
 1. Aseta vesipannu vesihanan alle
 2. Avaa vesihana
 3. Odota, kunnes pannu on täynnä
 4. Sulje vesihana
 2. Aseta pannu liedelle
 3. Lämmitä kunnes vesi kiehuu
 4. Siirrä pannu lieden sivuun
- Tarkennusta jatketaan kunnes suorittaja (robotti) voi tulkita ja suorittaa ne



Imperatiivinen paradigma

- Perinteisesti algoritmit ymmärretään koneelle annettavina komentosarjoina
 - Algoritmit kirjoitettu käskymuotoisina
- Tätä ohjelmoinnin suuntausta (eli paradigmaa) kutsutaan ***imperatiiviseksi*** tai ***proseduraaliseksi***
 - Algoritmit kuvaavat määrätyssä järjestyksessä suoritettavia toimenpidesarjoja



Ensi kerralla – torstaina 4.9. klo 14:15

- Lisää ohjelmointikielistä
- Ohjelmoinnin peruskäsitteitä
 - Muuttujat
 - Aritmeettiset- ja ehtolausekkeet
 - Valinta

