# Toteutusvaihe

- Toteutuksen välineistä
- Ohjelmointikielen valinnasta
- Ohjelmointiparadigmoista
- Yleisohjeita ohjelmoinnista
- Toteutusvaiheen dokumentit OHTU-projekteissa

# Toteutuksen välineistä

- perinteiset ohjelmointikielet
  - eri tyyppisiä
  - eri paradigmat
  - eri tarkoituksiin
- ohjelmointiympäristöt
  - kieli + tukiympäristö (kieliherkkä toimitin, debuggeri, analysoijat, rakenneselaimet, avustus, kirjastotliittymät,…)
- kehittimet
  - kieli + tukiympäristö + sovelluskehys

# Ohjelmointikielen valinnasta

- sovellusalueperustainen
  - tietty kieli havaittu hyväksi ja käytetty sovellusalueella, jolloin on saatavissa valmisosia
  - systeemiohjelmointi <- C
  - tosiaikajärjestelmät <- C, Ada
  - hallinnollinen <- COBOL, 4GL-kehitinkielet
  - matemaattiset <- FORTRAN
  - käyttöliittymät <- Visual Basic
  - henkilökohtaiset <- Visual Basic
  - tekoäly <- Lisp, Prolog

# Ohjelmointikielen valinnasta ...

- erikoissovelluksiin erikoiskieli
  - tietoliikenne <- SDL, ANS.1, Estelle
  - kääntäjät <- YACC, LEX
  - tietokannat <- SQL
  - www-sivut <- HTML, JavaScript

## Ohjelmointikielen valinnasta ...

- ohjelmistotuotantomallia tukeva kieli
  - prototyyppimalli <- 4GL kehitinkielet, Prolog, LISP
  - oliopohjainen kehitys <- C++, Smalltalk, Java, Eiffel
  - kokeileva <- Prolog, Smalltalk, 4GL kehitinkielet

## Ohjelmointikielen valinnasta ...

- tehokkuusnäkökohdat huomioiva valinta
  - C, assembler
- toteutusympäristön mukainen
  - Windows <- Visual Basic, Delphi/Pascal
  - UNIX <- C
  - WWW/Internet <-Java
- toteuttajien osaamisesta johtuva
  - vanhempi polvi <- COBOL, FORTRAN

# Ohjelmointiparadigmoista

- Imperatiivinen ohjelmointi (imperative programming)
  - suoritettavia lauseita (käskyjä), ohjelmoija kontrolloi täysin suoritusjärjestystä
    - alkeelliset -> konekielet, Fortran
    - rakenteiset
      - kootut rakenteet
      - lohko - paikalliset rakenteet ja näkyvyyssäännöt
      - aliohjelma - toiminnallisuuden abstraktiot
      - Algol 60, Pascal, C, (Fortran, COBOL)

# Imperatiivisia kieliä

- modulaariset
  - lohko, joka on mahdollista liittää muihin lohkoihin erillään tuotettuihin lohkoihin
  - julkiset ja yksityiset osat
  - Modula-2. Ada
- oliopohjaiset
  - modulaarinen +
  - perinnän hyväksikäyttö, monimuotoisuus ja dynaaminen sidonta
  - C++, Smalltalk, Java, Eiffel

# Ohjelmointiparadigmoista

- tapahtumaperustainen ohjelmointi (event based programming) / rakenne -- silti imperatiivista ohjelmointia
  - ohjelma = joukko tapahtumankäsittelijöitä, jotka on kytketty reagoimaan tietyn tyyppiseen tapahtumaan.
  - tapahtuma voi käynnistää useita reaktioita
  - tapahtumankäsittelijän aktivointi voi olla automaattinen (SQL:n triggerit ja assertiot) tai ohjelmoijan on tuotettava aktivointimekanismi

# Ohjelmoinnista

- Funktionaalinen ohjelmointi (functional programming)
  - ohjelma = joukko matemaattisia funktioita
  - funktioilla voi olla useita parametreja, mutta se tuottaa yhden tuloksen (voi olla rakenteinen)
  - sivuvaikutoksettomuus - ei tuhoavaa sijoituslausetta
  - funktio kuvataan toisten funktioiden ja niiden tuloksia yhdistävien operaatioiden avulla
  - ohjelman suoritus = funktion evaluointia tietyillä arvoilla
  - evaluoidaan vain se, mikä on tarpeen tuloksen saamiseksi (lazy evaluation)
  - kieliä: LISP, ML, Scheme

## Ohjelmointiparadigmoista

- Logiikkaohjelmointi (logic programming)
  - ohjelma muodostuu joukosta matemaattisen logiigan lauseita (päättelysääntöjä)
  - on_rahaa(X,Y) and Y>100000 => rikas(X).

    ehdot johtopäätös
  - logiikka: lausekkeiden tulkittataan määrittelevän predikaatin
  - proseduraalinen tulkinta: predikaatin määrittevä lauseke tulkitaan predikaatin laskevaksi ohjelmaksi
  - evaluointi perustuu ns samastukseen (unification)

## Logiikkaohjelmoinnista

- esim. tietämyskannassa:

  on_rahaa(kalle, 200).

  on_rahaa(liisa, 150000).
  - on_rahaa(X,Y) samastuu kumpaankin, mutta vain yhtä samastuvaa käytetään samanaikaisesti (se kumpaa käytetään valitaan (ainakin teoriassa) satunnaisesti) => epädeterministisyys
- evaluoinnin toteutuksiin liittyy yleensä peruutus, jolloin kokeillaan seuraavaa samastuvaa vaihtoehtoa ellei edellinen johda ratkaisuun.

# Logiikkaohjelmoinnista

– esim. kysymys on_rikas(kalle)
  - evaluoitaisiin evaluoimalla on_rikas predikaatin ehdot on_rahaa(kalle,Y) ja Y>100000,
  - ensimmäinen predikaatti samastuu predikaattiin on_rahaa(kalle,200) eli Y samastuu 200:n, jolloin toinen ehto 200>100000 ei toteudu (eikä koko predikaatti)

# Logiikkaohjelmoinnista

– ratkaisua voidaan kysyä joko sidotuin parametriarvoin (on_rikas(kalle)) tai vapain parametriarvoin (on_rikas(X)), jälkimmäisessä tapauksessa saadaan kaikki samastuvat arvot.

– rekursiiviset määrittelyt ovat yleisiä, samoin listarakenteiden käyttö määrittelyissä

– kieliä Prolog, Goedel

# Ohjelmointiparadigmoista

- rinnakkaisohjelmointi (concurrent programming)
  - ohjelma = joukko rinnakkain toimivia prosesseja
  - prosessilla on oma koodi ja oma paikallinen muisti
  - prosessien välinen kommunikointi perustuu prosessien väliseen viestinvälitykseen (vrt. tapahtumapohjainen)
  - prosesseja voi joutua synkronoimaan
    - prosessi tarvitsee toisen palveluja
    - poissulkeminen: estetään yhteisten tietorakenteiden samanaikainen käsittely
  - kieliä: Modula-2, Ada, Occam, Java

# Yleisohjeita ohjelmoinnista

- Kuvaavat nimet - vaikuttavat ohjelman luettavuuteen
  - ohjelmia luetaan useammin kuin kirjoitetaan: optimoi lukijan työtä (ei kirj.)
  - kieli - ei sekakieltä
  - yhtenäinen nimirakenne
    - jos lyhennetään, niin aina samoin
      - **ei** average_freq, frequency, frequency_maximum, min_fr, frq_totl
    - osasanat aina samassa järjestyksessä
      - **ei** average_freq, frequency_maximum

## Yleisohjeita ohjelmoinnista

- ohjelmassa esiintyvät vakiot on syytä nimetä, mikäli ne voivat jossain ohjelman elinkaaren vaiheessa muuttua
- tietorakenteille ja metodeille tulisi sopia löytämistä helpottava esittelyjärjestys, esim. aakkosjärjestys

## Yleisohjeita ohjelmoinnista

- parametrit tulisi välittää yhdenmukaisesti (ei tarpeetonta vaihtelua osoittimen ja arvon välillä)
- sisäkkäiset if lauseet voivat aiheuttaa ymmärtämisongelmia toisaalta myös monimutkaiset loogiset lausekkeet - näihin on kiinnitettävä erityistä huomiota
- pyrittävä mahdollisimman ymmärrettävään koodin - vältettävä kaikkea kikkailua

## Yleisohjeita ohjelmoinnista

- koodi on kommentoitava
  - otsakekommenttiin
    - käyttötarkoitus
    - liittymän kuvaus: parametrit & globaalit rakenteet
    - miten moduulia käytetään
    - mitä ulkopuolisia moduuleja itse käyttää
    - rajoitukset
    - tärkeimmät rakenteet ja toimintaperiaate
    - kehityshistoria: (laatija, hyväksyntä, muutokset)
    - nykytila

## Yleisohjeita ohjelmoinnista

- kytke koodi vastaavaan moduulisuunnitelmaan
  - ei pseudokooditasolla vaan syitä ja perusteita
- tietorakenteiden (ja kenttien) merkitys
- oikeellisuustarkistusten vaatimat kommentit
- tarvittavien kommenttien määrä riippuu
  - ohjelmointikielestä
  - muuttujanimien valinnasta ym. lisäinformaatiosta
- kommenttejakin täytyy ylläpitää

## Yleisohjeita ohjelmoinnista

- Trace-optioiden käyttö
  - lokitulosteita/ valinnaisia lokitulosteita / ei lokitulosteita lainkaan
  - alkuvaiheen testauksen aputulosteet voi muotoilla trace generating -aliohjelmiksi, jotka voidaan aktivoida
    - esikääntäjän optioilla
    - käynnistysoptioilla
    - komennolla suoritusaikana

## Yleisohjeita ohjelmoinnista

- **Sijoittelu (layout)**
  - useimmissa kielissä sijoittelu ohjelmoijan päätettävissä (vs sarakesidonnaiset kielet)
  - usein standardoitu, jopa automatisoitu
    - tyhjä tila on halpaa, mutta sen puute voi tulla kalliiksi! Huomattava vaikutus luettavuuteen
  - koko ohjelmistossa yhdenmukaista
    - sisennykset, tyhjät rivit
    - lauseiden sijoittelu riveille, alku ja loppusulut
      - begin … end, { …}
    - kommenttien sijoittelu

# Toteutusdokumentit

- Toteutusdokumentti
- Asennusohje
- Ylläpitäjän käsikirja
- Käyttöohje

- Demonstraatio

# Toteutusdokumentti

- OHTU-projekteissa
  - Johdanto
  - aikaisempien dokumenttien tietoihin tehdyt täsmennykset
  - toteutustason ratkaisut
    - periaatteet, ohjelmointimenetelmät, kielet, kirjasot, sovelluskehittimet, sovelluskehykset, aputyökalut, kokemukset
    - ohjelmiston sijainti, versionhallinta

- Moduulitestaus
  - testausperiaate, testiaineisto, kattavuusarviot
- Järjestelmätestaus
  - testausperiaate, aineisto, kattavuusarviot
- Suorituskykyanalyysi
  - ajan ja tilan käyttö erilaisissa tätä varten suunnitelluissa testiajoissa, käyrät, johtopäätelmät

- Jatkokehitys
  - mitä suunniteltua ei toteutettu, miksi?
  - Laajennusehdotuksia
- Liitteet
  - koodausstandardi (jatkokehittäjille)
  - moduulien lähdekoodi
  - kirjastojen otsikkotiedot
  - testitapaukset, testitulokset, testilokit

# Asennusohje

- Tuotteen identifiointi
- Toiminnallinen yleiskuvaus
- Ohjelmiston toimintaympäristö
- Asennuskomennot ja niihin liittyvät konfigurointitiedostot, muutokset näihin ympäristön mukaan
- Ohjelman käytössä tarvittava apumateriaali

# Käyttöohje

- Johdanto tuotteeseen
  - tuotteen käyttötarkoitus
  - oletettu käyttäjäprofiili ja käyttötapa
  - käyttöympäristö
  - viitteet apumateriaaliin
- Yleiset rajoitteet
  - kuka saa käyttää
  - tekniset rajoitteet (muistimäärät, jne)

- Takuut ja sitoumukset
- Tuotteen käyttö
  - lähdetään liikkeelle onnistuneen asennuksen jälkeisestä tilanteesta; viite asennusohjeeseen
  - yhteenveto tyypillisimmistä käyttömuodoista ja toimintosarjoista (vrt käyttötapaukset)

---

- ohjeet toimintojen suorittamiselle
  - kuvia käyttöliittymistä
  - toimenpidevalinnat ja niiden semantiikka
  - syötteiden/tulosteiden semantiikka ja syntaksi
  - konkreettisia esimerkkejä käyttötilanteista
- käytettävissä olevat optiot
- toimintojen peruutusmahdollisuudet
- suorituksen tilan tallennus ja uudelleenkäynnistys

- Virheilmoitukset
  - mitä, miksi, kuinka toimia
- Toiminnalliset erot eri käyttöympäristöissä
  - esim käyttöliittymän ulkoasu vaihtelee ikkunointijärjestelmästä toiseen
- Liitteet
  - komentokielien syntaksit

# Ylläpitäjän käsikirja

- Johdanto
  - tuotteen identifiointi
- Toiminnallinen yleiskuvaus
  - tarkoitus, tavoitteet
  - yleisarkkitehtuuri
  - looginen tieto(kanta)ratkaisu
  - käyttöliittymäkuvaus
  - liittymät muihin järjestelmiin

- Toimintaympäristö
  - Laitteisto
  - KJ
  - Ohjelmointikielet, kääntäjät, versioineen
  - CASE-työkalut

- Toteutuskuvaus
- Ylläpidossa tarvittavat tukiohjelmat
  - testaustyökalut
  - ylläpitotyökalut
  - sovelluskehittimet, skehykset, kirjastot
  - metriikkatyökalut
  - dokumentointityökalut

- Noudatettavat standardit
- Ylläpito-ohjeita
  - Keskeneräisten asioiden osoittaminen
  - Moduulin korvaaminen uudella versiolla käytännössä -> ohjelmiston toteutuksen hakemistorakenne ja versionhallinta, Makefilet yms täytyy selostaa

- Luettelo kaikkien ohjelmiston, apuohjelmien ja dokumentaation osista, niiden säilytyspaikoista ja formaateista