

# **Yhd-102.2010**

# **Ympäristösuunnittelun tietotekniikka**

**Luennot ja harjoitustyöohjeet**

**Ari Jolma**

**Hanna Vellonen**

## Yhd-102.2010 Ympäristösuunnittelun tietotekniikka

### Luennot ja harjoitustyöohjeet

### Edition 6.2.2014

Tekijä

Ari Jolma

*[ari.jolma@aalto.fi](mailto:ari.jolma@aalto.fi)*

Tekijä

Hanna Vellonen

*[hanna.vellonen@aalto.fi](mailto:hanna.vellonen@aalto.fi)*

Copyright © 2010-2014 Ari Jolma and Hanna Vellonen This material may only be distributed subject to the terms and conditions set forth in the GNU Free Documentation License (GFDL), V1.2 or later (the latest version is presently available at <http://www.gnu.org/licenses/fdl.txt>).

Tässä kirjassa on esitelty luennoilla käsiteltävät asiat lyhyesti ja annettu ohjeet harjoitustöiden tekemiseksi.

---

<b>1. Esipuhe</b>	<b>1</b>
<b>2. Luennot</b>	<b>3</b>
2.1. Tietokoneesta ja tietojärjestelmistä .....	3
2.2. Tiedosta ja aineistoista .....	5
2.2.1. Peruskäsitteitä .....	6
2.2.2. Paikkatieto .....	7
2.2.3. Paikkatietoinfrastruktuuri .....	10
2.2.4. Talonrakennusalan tieto .....	10
2.2.5. Aineistojen ominaisuuksista .....	10
2.2.6. Esimerkkejä aineistoista .....	11
2.3. Malleista ja menetelmistä .....	12
2.3.1. Tietorakenteista .....	14
2.3.2. Algoritmi .....	14
2.3.3. Simulointi .....	15
2.3.4. Optimointi .....	16
2.3.5. Laskennallisten mallien toteuttamisesta tietokoneessa .....	17
2.4. Tietokannoista .....	18
2.4.1. XML .....	20
2.4.2. Teknisen tietomallin määrittely .....	21
2.4.3. Tiedon muokkaus .....	21
2.4.4. Tiedon haku .....	22
2.5. Paikkatiedosta ja paikkatietomenetelmistä .....	22
2.5.1. Yksinkertaiset paikkatietokohteet .....	23
2.5.2. Spatiaaliset yhteydet .....	24
2.5.3. Spatiaaliset mitat .....	24
2.5.4. Funktiot joiden tulos on geometria .....	24
2.5.5. Rasteritekniikat .....	24
2.5.6. Kolmioverkko ja Voronoi-diagrammi .....	25
2.5.7. Kuvien georeferointi .....	26
2.6. Suunnittelun tietojärjestelmistä .....	26
2.6.1. Tietojärjestelmä .....	26
2.6.2. Tietoturvallisuudesta .....	27
2.6.3. Tietojärjestelmien laadusta .....	27
2.6.4. Tietomallin ja prosessikuvauksen yhteydestä .....	28
2.6.5. Tietojärjestelmätekniikoita .....	28
2.6.6. Paikkatietojärjestelmä .....	28
2.6.7. Päätöksenteon tukijärjestelmä .....	29
2.6.8. Tietokoneavusteinen suunnittelu .....	29
2.7. Tiedonhankinnasta .....	29
2.7.1. Satelliittipaikannus .....	30
2.7.2. Kaukokartoitus .....	30
2.7.3. Maastomittaus .....	31
2.7.4. Muita menetelmiä ja tekniikoita .....	32
<b>3. Aineistoharjoitus</b>	<b>33</b>
3.1. Aineistotuottajia .....	33
3.2. Aineistojen tarkastelu .....	34
3.3. Raportointi .....	36
<b>4. Ohjelmointiharjoitus</b>	<b>37</b>
4.1. Varastoyhtälö .....	37
4.2. Regressioanalyysi .....	37
4.3. Laskenta MATLAB:illa .....	38
4.4. Raportointi .....	39

---

<b>5. Taulukkolaskentaharjoitus 1: simulointi</b>	<b>41</b>
5.1. Varastoyhtälö .....	41
5.2. Altaan toiminnan simulointi .....	41
5.3. Simulointimallin raportointi .....	42
<b>6. Taulukkolaskentaharjoitus 2: optimointi</b>	<b>43</b>
6.1. Verkoston kuvaaminen taulukkolaskentaohjelmassa .....	43
6.2. Lyhin reitti verkostossa .....	44
6.3. Verkoston välityskyky .....	44
6.4. Raportointi .....	45
<b>7. Tietokantaharjoitus</b>	<b>47</b>
7.1. Tietokantayhteys .....	47
7.2. Tietokannan toteuttaminen .....	49
7.3. Tiedon vieminen tietokantaan .....	51
7.4. Vastausten hakeminen tietokannasta .....	52
7.5. Raportointi .....	53
<b>8. Paikkatietoharjoitus</b>	<b>55</b>
8.1. Paikkatietoa sisältävien tietokantataulujen luominen .....	55
8.2. Paikkatiedon avaaminen tietokannasta paikkatieto-ohjelmaan .....	56
8.3. Paikkatietoanalyysit tietokannassa .....	58
8.4. Raportointi .....	61
<b>9. AutoCAD käyttöharjoitus</b>	<b>63</b>
9.1. Käyttöliittymän perusominaisuudet .....	63
9.2. Ohjattu harjoitus .....	65
9.3. Itsenäisesti tehtävä harjoitus .....	84
<b>10. AutoCAD pikaopas</b>	<b>87</b>
10.1. Käyttöliittymä .....	87
10.2. Kommenttien antaminen ja peruminen .....	88
10.3. Zoomaus ja panorointi .....	89
10.4. Tasojen hallinta .....	91
10.5. Snap-asetukset .....	91
10.6. Piirtäminen .....	92

---

# Esipuhe

Ympäristösuunnittelun tietotekniikka on tietojenkäsittelytieteen soveltamista ympäristötieteeseen ja -tekniikkaan siten, että näkökulmana on tekniikka, erityisesti rakennustekniikka (civil engineering).

Tässä oppikirjassa on esitetty ympäristösuunnittelun tietotekniikan perusteet lyhyesti.

Perusteiden kuvauksen jälkeen kirjassa on joukko harjoitustehtäviä.

---

---

# Luennot

Tässä kappaleessa on lyhyehköt tekstit, joissa on luentojen keskeinen sisältö.

## 2.1. Tietokoneesta ja tietojärjestelmistä

### *Tietokonelaitteisto*

Tietokone on ohjelmoitava kone, joka koostuu muistista, prosessorista ja joukosta tiedon syöttö- ja tulostuslaitteita. Tietokoneen muistia käytetään sekä ohjelmien itsensä että ohjelmien käyttämien ja tuottaman tiedon tallentamiseen. Tietokoneohjelma on joukko käskyjä, jotka prosessori toteuttaa. Niiden mukaan tietoa haetaan muistista, viedään tulostuslaitteelle, muokataan prosessorissa jne. Tavallisia syöttö- ja tulostuslaitteita ovat mm. näppäimistö, hiiri, näyttö, paperitulostin, verkkomodeemi.

Tietokoneessa on yleensä pysyvää ja tilapäistä muistia. Kun ohjelma suoritetaan, se haetaan tilapäiseen muistiin ja ohjelma käsittelee yleensä tietoa, joka on tilapäisessä muistissa. Pysyvään muistiin tieto pitää ohjelman käyttäjän usein erikseen tallentaa.

Tietokoneen keskeinen osa on prosessori, joita on jo 1970-luvulta lähtien tehty yhtenä elektronisena komponenttina. Prosessoreita on nykyisin lähes kaikissa laitteissa, joten tietotekniikkaa on kaikkialla.

### *Tietokoneohjelma*

Tietokoneen käyttöjärjestelmä on tietokoneohjelmien tuottama kokonaisuus, jossa ajetaan tietokoneohjelmia, joista osa kommunikoi tietokoneen käyttäjän kanssa. Aiemmin käyttäjä tyypillisesti käynnisti ohjelman käyttöjärjestelmän tuottamalta komentoriviltä (Command Line Interface, CLI). Ohjelma luki lähtötiedon tiedostosta ja tuotti tulostiedon tiedostoon. Nykyisin ohjelmat ovat yleensä vuorovaikutteisia ja lisäksi usein käyttöjärjestelmän luomaa graafista (ikkunoitua) ympäristöä (Graphical User Interface, GUI) hyödyntäviä. Vuorovaikutteiset ohjelmat käsittelevät tietoa, joka on tilapäiseen muistiin luodussa mallissa. Vuorovaikutus voidaan toteuttaa joko komentorivin tai graafisen käyttöliittymän avulla.

Tietoverkon kautta ohjelma voi olla yhteydessä toiseen ohjelmaan toisessa tietokoneessa. Toisessa tietokoneessa oleva ohjelma voi käyttää suoraan ensimmäisen tietokoneen tulostuslaitteita kuten näyttöä, mutta yleensä yhteyden avulla vain välitetään tietoa ohjelmien välillä. Web-tekniikassa tilanne, jossa verkkoselain esittää tietoa, joka on toisessa tietokoneessa (ns. palvelimessa) olevan ohjelman lähettämää tai tuottamaa, on hyvin tyypillinen. Verkkoselain voi myös näyttää lomakkeen, johon syötetyn tiedon selain lähettää palvelimelle, kun käyttäjä painaa tiettyä käyttöliittymäelementtiä sivulla. Nykyisin web-tekniikassa on yleistä luoda sivustoja, joissa on mukana paljon suoritettavaa ohjelmakoodia. Verkkoselain ajaa kyseisen koodin, joka voi esimerkiksi kommunikoida palvelimen kanssa käyttäjän sitä erityisesti huomaamatta tai muokata käyttöliittymää käyttäjän toiminnan mukaan.

### *Tietojärjestelmä*

Tietojärjestelmä on kokonaisuus, joka koostuu tietokoneesta tai tietokoneista ja muista tietoa tuottavista ja käsittelevistä laitteista, niissä toimivista ohjelmista, tietoverkosta ja/tai muista kommunikaatiojärjestelmistä, ihmisistä ja kommunikaatiosta järjestelmässä ja järjestelmän toimintaa ohjaavista säännöistä. Tietojärjestelmissä on erotettavissa kolme eri tasoa, joissa tietoa on tai sitä käsitellään.

1. Tietoa tallentavat tai tuottavat resurssit
2. Sovelluslogiikka ja sen toteuttava järjestelmä, jossa tietoa muokataan
3. Esitystaso, jonka kautta tietojärjestelmä kommunikoi ulkopuolisen maailman (ihmisten tai toisten tietojärjestelmien) kanssa; tällä tasolla tietojärjestelmän tieto muokataan (esitetään) ulkopuolisen maailman tarpeiden ja mahdollisuuksien mukaan

Eri tasojen erottaminen on järkevää koska usein samaa tietoa käytetään useassa eri laskentasovelluksessa ja useassa eri käyttäjäsovelluksessa.



### Yhteistoiminnallisuus

Yhteistoiminnallisuus on ohjelmakomponenttien, ohjelmien tai tietojärjestelmien kyky toimia yhdessä. Yhteistoiminnallisuus edellyttää yleensä, että järjestelmän osilla on julkaistu ja kohtalaisen stabiili rajapinta, jonka kautta kommunikaatio tapahtuu. Edelleen yhteistoiminnallisuus edellyttää että järjestelmän osat ymmärtävät kommunikaatiossa käytettäviä viestejä ainakin niiltä osin kuin se on niiden itsensä kannalta tarpeellista.

#### Tietoinfrastrukturi

Tietoverkon merkityksen myötä tietojärjestelmien keskeiseksi käsitteeksi ja niiden arkkitehtuurin perustaksi on noussut toisaalta palvelun ja toisaalta viestin käsite. Palvelu tarkoittaa tällöin tietokoneohjelman (palvelimen) toiselle tietokoneohjelmalle (asiakkaalle) tarjoamaa palvelua. Tarjottu palvelu voi olla hyvin yksinkertainen, esim. tieto tämänhetkisestä lämpötilasta Helsingin Kaisaniemessä, tai monimutkainen, esim. lentolipun varaaminen, johon kuuluu monta vaihetta ja jota käyttävältä asiakkaalta edellytetään myös tiettyjä toimenpiteitä. Palvelun käsite siis osittain yhdistää yllä kuvatun tietojärjestelmä-arkkitehtuurin resurssi- ja sovelluslogiikkatasot. Asiakas käyttää palvelua sen rajapinnan kautta ja se, miten palvelu on toteutettu on täysin palvelun tarjoajan asia ja voi esimerkiksi perustua muihin palveluihin. Huomaa myös, että "asiakas" voi olla itsenäisesti toimiva ohjelma tai ohjelma, jota ihminen käyttää.

Viesti on jotain, joka välittää ja sisältää informaatiota. Viestinnässä on oleellista, että ainakin osa viestistä on vastaanottajan tulkittavissa ja että vastaanottaja tunnistaa ja pystyy erottamaan tuon osan. Viestin tulkinnan kannalta on tärkeää, että viestin lähettäjä ja viestin vastaanottaja ymmärtävät viestin mahdollisimman samalla tavalla. Esimerkiksi jos viestissä käsitellään lämpötiloja ilman tarkempaa määrittelyä, on oleellista, että on sovittu yhdessä käytetäänkö Fahrenheit- vai Celsius-asteita.

Tietoinfrastruktuuriksi voidaan nimittää useiden tietojärjestelmien kokonaisuutta, jossa tietoa siirtyy ja on käytettävissä. Tietoinfrastruktuuriin kuuluu tietojärjestelmien lisäksi käytäntöjä, teknologioita, standardeja, resursseja jne. Yksi tietoinfrastruktuurin tarkoitus voi olla parantaa tietojärjestelmissä olevien tietojen löytämistä ja hyödyntämistä erityisesti uusiin tarkoituksiin. Uusia tarkoituksia voi syntyä erilaisen tietojen yhdistämisestä. Tietoinfrastruktuuriin liittyy tietoa, tietoa siinä olevasta tiedosta sekä palveluita, joiden kautta näitä voidaan etsiä ja hyödyntää. Tietoinfrastruktuurin yksi perusajatus on, että tietojärjestelmiä ei yhdistetä vaan että ne toimivat yhdessä. Tähän sisältyy myös ajatus siitä, että tietoa käytetään palveluna, sitä varastoida useisiin paikkoihin.

#### Virtuaalikone ja -palvelin sekä ns. pilvipalvelut

Virtuaalikone on tietokoneen simuloima tietokone. Virtuaalipalvelin on menetelmä, jossa yksi tietokone tuottaa (samantyyppisiä) palveluita eri nimillä. Esimerkiksi web-osoite <http://2014.ogrs-community.org> on virtuaalipalvelin, jonka tuottaa palvelintietokone, joka on varsinaisesti luotu verkkopalvelinta <http://geoinformatics.aalto.fi> varten. Virtuaalipalvelin näyttää ulospäin erilliseltä tietyssä osoitteessa olevalta palvelimelta, mutta ainoastaan palvelufunktio on virtualisoitu. Virtuaalikone sen sijaan virtualisoi tietokoneen funktion, joko rajoitetusti lähinnä tietokoneen prosessorin ja muistin tai laajemmin kokonaisen tietokoneen arkkitehtuurin. Ensinmainitussa tapauksessa virtuaalikoneen tarkoitus on ajaa yhtä ohjelmaa, jälkimmäisessä tapauksessa virtuaalikoneen tarkoitus on tuottaa alusta, jolla ajetaan kokonaista käyttöjärjestelmää. Esimerkki ensinmainitun tyyppisestä virtuaalikoneesta on käännettyjen Java-ohjelmien ajamiseen tarkoitettu Java-virtuaalikone. Esimerkki jälkimmäisestä on edellämainittu



geoinformatics.aalto.fi, joka on erittäin tehokkaan tietokoneen luoma virtuaalikone. Myös esim. Virtual-Box-ohjelmisto luo virtuaalisen x86-prosessoripohjaisen<sup>1</sup> tietokoneen.

Yllä kuvatulle virtuaalipalvelulle ja -koneelle ikäänkuin käänteinen teknologia on sellainen, jossa usea fyysinen tietokone luo yhden, mutta tehokkaamman tietokoneen tai palvelun. Tällaista teknologiaa tarvitaan kun yhden verkko-osoitteen (esim google.com) pitää tuottaa palvelu erittäin suurelle joukolle asiakkaita.

Virtuaalikonetekniikka, tarkemmin sanottuna tietokonearkkitehtuurien virtualisointi, on ns. pilvipalveluiden käytännön perusta. Pilvipalvelulla tarkoitetaan tietoteknisten resurssien tarjoamista Internetin välityksellä. Tietotekninen palvelu voi tarkoittaa yksinkertaisesti virtuaalikoneen tai -koneiden (käytännössä koneissa on valmiina jokin käyttöjärjestelmä) tarjoamista asiakkaalle. Virtuaalikoneteknologia mahdollistaa uusien virtuaalikoneiden luomisen varsinaisesta laitteistosta riippumattomasti. Toisaalta se myös mahdollistaa resurssien joustavan lisäämisen tarpeen mukaan.

Muita tietoteknisiä palveluita edellämainitun infrastruktuuripalvelun (infrastructure as a service, IaaS) lisäksi ovat esimerkiksi alustapalvelu (platform as a service, PaaS) ja ohjelmistopalvelu (software as a service, SaaS). Alustapalvelussa asiakkaalle tarjotaan tietty ohjelmistokokonaisuus valmiina, jonka avulla asiakas pystyy tuottamaan itselleen haluamansa toiminnallisuuden. Ohjelmistopalvelussa asiakkaalle tarjotaan jokin tietty ohjelmisto, esimerkiksi web-pohjaisena toteutettu tietokannan hallintaohjelmisto, jonka avulla asiakas voi luoda ja hallita esimerkiksi jotain aineistoa.

## 2.2. Tiedosta ja aineistoista

”Ympäristösuunnittelu” tämän kurssin nimessä tarkoittaa tavoitteisiin perustuvaa, päätöksentekoon ja rakentamiseen (laajasti käsitettynä) johtavaa toimintaa. Tavoitteena on yleensä ”kehitys” tai kehitykseen reagoiminen, ja muina tavoitteina tai reunaehtoina on luonnonympäristöön, rakennettuun ympäristöön, sosiaaliseen ympäristöön, politiikkaan ja hallintoon sekä taloudellisiin tekijöihin liittyviä asioita. Suunnittelu tapahtuu tiedon pohjalta, tietoa ja tiedonkäsittelymenetelmiä hyödyntäen. Suunnittelun tuloksena on dokumentti, joka kuvaa tavan miten rakentaminen tai toiminta tulee järjestää.

Edelleen, ”ympäristösuunnittelulla” tarkoitetaan tässä suunnittelua joka kohdistuu maankäyttöön, tienrakentamiseen, liikennejärjestelmiin, luonnonvarojen, kuten maa-ainesten ja veden, hyödyntämiseen, yhdyskuntien teknisiin järjestelmiin ja niin edelleen. Ympäristösuunnitteluun liittyy sekä hyödyntämis-, että suojelunäkökulma: ihminen pyrkii hyödyntämään ja muokkaamaan ympäristöä omaksi edukseen mutta toisaalta ihminen myös pyrkii estämään omasta toiminnastaan ja luonnon toiminnasta aiheutuvia haitallisia vaikutuksia. ”Haitallisuus” on luonnollisesti subjektiivinen käsite ja edellyttää yhteisymmärrystä.

Ympäristösuunnitteluun liittyy sekä spatio-temporaalinen- että järjestelmänäkökulma. Spatio-temporaalisuus tarkoittaa sitä, että tarkasteltu ”ympäristö” on olemassa maapallon pinnalla, pääasiassa sen hyvin ohuessa kerroksessa ja että tuo ympäristö muuttuu ajan myötä. Järjestelmänäkökulma tarkoittaa sitä, että ympäristön elementit kytkeytyvät toisiinsa eri tavoilla ja että usein oleellista ei ole esimerkiksi se, miten lähellä joku kohde on jotain toista kohdetta vaan se miten nuo kohteet ovat kytkeytyneet toisiinsa.

Suunnitteluun liittyy syklejä: olemassaolevan kokeminen ja havainnointi → ongelman, puutteen, mahdollisuuden tms. havaitseminen ja analysointi → ratkaisun etsiminen, suunnitelman tekeminen ja päätöksenteko → suunnitelman toteuttaminen → olemassaolevan kokeminen ja havainnointi. Näitä syklejä etenee koko ajan useita rinnakkain ja eri sykleillä on erilaisia ajallisia kestoja.

Kaikki suunnittelu perustuu tietoon. Ympäristöä havainnoidaan ja mitataan koko ajan ja ympäristöön rakennettaessa rakentaminen tapahtuu toisaalta suunnitelman (eli tiedon) pohjalta ja toisaalta rakenta-

<sup>1</sup> x86 on yleinen nimi Intel-yhtiön kehittämälle PC-tietokoneiden käyttämälle prosessoriarkkitehtuurille.

misesta tallennetaan tietoa. Tietoa on hyvin monenlaista. Hyvin yksinkertaiseenkin tietoon (esimerkiksi veden lämpötilan mittaussarvo) liittyy aina monenlaista oheis- eli metainformaatiota (missä mittausta on tehty, milloin mittausta on tehty, kuka mittauksen on tehnyt jne.). Tieto kootaan yleensä aineistoiksi. Aineistoihin, kuten yksittäiseen tietoonkin, liittyy oma informaationsa, joka kuvaa esimerkiksi kuka aineiston on tehnyt ja milloin, minkälaisia oikeuksia aineistoon liittyy jne.

Järjestelmällisen havaintotoiminnan tuloksena syntyvä aineisto on yleensä järjestettävissä hyvin selkeän rakenteen mukaisesti. Toisaalta taas esimerkiksi sanallinen informaatio voi olla hyvinkin huonosti tallennettavissa muuten kuin yhtenäisenä tekstinä. Yksi tiedon tallentamiseen liittyvä ongelma on kohteen, jota tieto kuvaa, monimutkaisuus tai kuvauksen hankaluus. Toisaalta juuri tämä kohteiden, niiden välisten suhteiden, prosessien jne kuvaaminen on keskeistä pyrittäessä hallitsemaan tietoa organisaatiossa ja organisaation tietoa. Edelleen on erotettava toisistaan tiedon tallentaminen ja hallinta tiedon kokoamisesta käytettäväksi, dokumenteiksi jne (vrt. tietojärjestelmien perusrakenne yllä).

Tieto on yhteiskunnan hallinnan perusväline. Tietoon liittyy suuria talouteen, valtaan ja politiikkaan liittyviä intressejä. Jokainen järjestäytyneen yhteiskunnan kerää itsestään, alueestaan ja järjestelmänsä tietoa. Käytännössä tietoa kerätään erilaisiin tietojärjestelmiin tietoresursseiksi. Näitä tietojärjestelmiä ylläpitävät yhteiskunnan eri organisaatiot.



### Avoin tieto

Avoin tieto (open data) on ajatus, että tietyn tiedon (tässä viitataan nimenomaan dataan eli aineistoihin ja niiden osiin) tulisi olla kaikkien käytettävissä ja uudelleenjulkaitavissa (vrt. tietoinfrastruktuurin ajatus). Käytännössä ajatus on, että avoin tieto olisi saatavilla tietyssä verkko-osoitteessa kohtuullisen pieninä tai helposti hyödynnettävänä paloina. Helsinki Region Infoshare <http://www.hri.fi/fi/> on esimerkki avoimesta datasta.

### 2.2.1. Peruskäsitteitä

*Ontologia* filosofian osana on tiede, jossa kysytään mitä tarkoittaa olemassaolo, mitä on olemassa ja miten jonkun olemassaolo liittyy jonkin toisen olemassaoloon. Moderni filosofia Kantista lähtien tuo ontologiaan mukaan tarkastelijan: olemassaolo ja sen ominaisuudet ovat oleellisesti tarkastelijasta ja tarkastelun näkökulmasta määräytyviä. Tästä seuraa se, että on käytännössä mahdotonta luoda kaikenkattavaa ontologista kuvausta maailmasta.

Tietoteknisenä terminä ontologia tarkoittaa systemaattista ja johonkin esitysmuotoon ja tarkastelun syyhyn nojaavaa olemassaolevan kirjaamista ja kategorisointia kyseisen olemassaolevan hallitsemiseksi tietokoneohjelmissa tai tietojärjestelmissä.

*Semanttinen* tarkoittaa merkitykseen perustuvaa. Asioiden esityksellä on *syntaktinen* eli muotoon perustuva ja semanttinen puoli. Muodon ja rakenteen oikeellisuuden tarkastus on tietoteknisillä välineillä melko helppoa, mutta semantiikkaan liittyvät ongelmat ovat vaikeita havaita tietoteknisesti ilman ihmisen suoraa apua. Viestillä on aina tietty syntaksi ja tietty semantiikka. Käytännössä semanttinen tieto on entiteeteistä (kohteista, resursseista) olevaa kuvailevaa tietoa tai metatietoa, joka on yleensä tallennettu käyttäen jotain sovittua teknologiaa.

*Entiteetti* on jotain, jolla on oma itsenäinen olemassaolonsa. Olemassaolon ei tarvitse olla materiaalista tai edes millään lailla "todellista", määrittely riittää. Voidaan esimerkiksi määrittellä velho-oppilas nimeltä Harry Potter, joka on olemassa vain koska me sanomme niin. Samanlaiset (jollain lailla samanlaiset) entiteetit muodostavat joukon, joka voidaan käsittää yksinkertaisesti joukkona tällaisia entiteettejä tai sitten tällaisten entiteettien ideana. *Idean* ajatus on houkutteleva, mutta johtaa käytännössä helposti ongelmiin: milloin voidaan sanoa, että jokin kohde on oikeasti esimerkiksi talo eikä jokin muu rakenne? Eri aloilla entiteeteistä puhutaan erilaisilla nimillä. Ohjelmointitekniikassa entiteettiä kutsu-

taan yleensä olioksi ja olioiden joukkoa tai mallia (ideaa) luokaksi. Tietomallinnuksessa entiteettiä kutsutaan yleensä kohteeksi.

Entiteettien välillä voi olla erilaisia yhteyksiä. Kukin yhteys on periaatteessa oma entiteettinsä. Esimerkiksi tietyllä oppilaalla on tietty opettaja ja näiden välillä on oppilas-opettaja -suhde. Tällä tietyllä suhteella on omia ominaisuuksiaan, esimerkiksi alku- ja loppupäivämäärä, joka tekee siitä itse asiassa oman entiteettinsä.

Yhteydet voivat olla kahden entiteetin välisiä tai yksi yhteys voi kytkeä useampia entiteettejä toisiinsa (esimerkiksi kirja, sen kirjoittaja, sen kustantaja ja sen myyjä). Yhteen entiteettiin voi liittyä useampia toisentyypisiä entiteettejä (yhdessä kaupungissa asuu useita ihmisiä) tai kaksi entiteettijoukkoa voi olla yhteydessä toisiinsa siten että kukin entiteetti kummassakin joukossa voi olla yhteydessä useaan entiteettiin toisessa (ajattele esim. yliopiston kurssien ja opiskelijoiden muodostamia joukkoja).

Entiteeteillä voi olla ominaisuuksia. Ominaisuuden arvon voi käsittää entiteetiksi ja "olla ominaisuus" tarkoittaa, että entiteetillä ja ominaisuuden arvolla on yhteys. Sen miettiminen onko ominaisuus entiteetti vai ei voi vaikuttaa hiusten halkomiselta mutta tällä erolla voi olla teknisen toteutuksen ja sovelusten toiminnallisuuden ja jopa luotettavuuden kannalta tärkeä merkitys. Asian problematisointi myös osoittaa tarkastelukulman merkityksen olemassaolevasta olevan tiedon rakenteen määrittelyssä.

Ominaisuuksia voidaan luokitella.

*Nominaalinen* ominaisuus on ainoastaan entiteettejä luokitteleva tai identifioiva.

*Ordinaalisen* ominaisuuden avulla entiteettejä voidaan laittaa järjestykseen (esim. huono, hyvä, erinomainen).

*Intervalliominaisuuden* arvojen erotukset ovat mielekkäitä (esim. Celsius-asteet).

*Rationaalisten* ominaisuuksien suhteet ovat mielekkäitä (esim. kappaleen massa kilogrammoina; voidaan sanoa, että kappaleen massa on kaksi kertaa suurempi kuin jonkin toisen kappaleen massa).

*Syklinen* ominaisuus saa arvoja esim. 0:sta 360:een jonka jälkeen arvot kasvavat jälleen 0:sta, eli 0 ja 360 ovat ikäänkuin sama arvo. Data-analyysissä, jossa tarkastellaan muuttujien arvojen välisiä samankaltaisuuksia matemaattisesti, tästä seuraa ongelmia koska etäisyys 350:n ja 10:n välillä on yhtäsuuri kuin 10:n ja 30:n. Yksi ratkaisu tähän ongelmaan on muuntaa syklinen ominaisuusarvo kahdeksi arvoksi esimerkiksi sini- ja kosini-funktioilla.



## UML

Unified Modeling Language (UML) on graafinen kieli järjestelmien määrittelyyn, suunnitteluun ja dokumentointiin. UML sisältää erilaisia kaaviotyyppejä järjestelmien rakenteen, toiminnan ja vuorovaikutuksen kuvaamiseen. UML:n on standardoinut ja sitä ylläpitää Object Management Group -niminen tietotekniikka-alan konsortio.

### 2.2.2. Paikkatieto

Paikkatieto on tietoa, josta osa on maantieteellinen sijainti tai kohde, jolla on maantieteellinen sijainti, tai joukko näitä. Tietoon voi kuulua sijainnin laatuun tai muotoon, maantieteellisiin prosesseihin, joukon sisäisiin yhteyksiin, havaintoihin, suuntiin tms. liittyviä seikkoja.

Maantieteellinen sijainti ilmaistaan maantieteellisten koordinaattien, karttakoordinaattien tai jonkin muun referenssijärjestelmän kuten esimerkiksi postiosoitteen avulla. Yleinen tapa määrittellä maantieteellinen sijainti on käyttää leveys- ja pituusasteita. Näiden lisäksi korkeusasemalla voi olla merkitystä.

## Luku 2. Luennot

Maantieteellinen pituusaste on lukuarvoltaan 0–180 astetta, joko itään tai länteen Greenwichin (Lontoossa) kautta kulkevasta nollameridiaanista. Maantieteellinen leveysaste taas on lukuarvoltaan 0–90 astetta, joko pohjoiseen tai etelään päiväntasaajasta.

Korkeusasema ilmoitetaan yleensä suhteessa merenpinnan korkeuteen. "Merenpinnan korkeus" voidaan määrittellä *geoidin* avulla. Geoidi on se pinta, jolla Maan painovoimakenttä saa vakioarvon ja johon meren pinta asettuisi, jos ilmanpaineen eroja tms. ei olisi. Geoidi on hyvin epätasainen kappale ja vaikea mitata, jonka takia käytetään idealisoidumpia matemaattisia muotoja. Yleisesti käytetty maan muotoa kuvaava malli on geoidia approksimoiva ellipsoidi. Maantieteelliset koordinaatit ovat johonkin referenssijärjestelmään (ellipsoidi, sopimus nollameridiaanista, merenpinnan korkeuden määrittely jne) tukeutuvia koordinaatteja (astelukuja). Esimerkiksi GPS-järjestelmä (ks. alla) käyttää WGS 84 -referenssijärjestelmää, joka on maailmanlaajuinen.

Karttaprojektio on tapa kuvata (Maan) käyrä pinta tasolle. Jokainen kuvaustapa vääristää pinnan jollain tavoin, josta seuraa mm. se, että joko kohteiden pituudet tai pinta-alat muuttuvat kuvauksessa. Erilaisia karttaprojektioita on teoriassa ääretön määrä mutta käytännössä kuvattava alue, sen koko ja konventiot määräävät käytettävän karttaprojektion. On esimerkiksi sovittu, että nykyisin Suomessa koko maan kattavien paikkatietoaineistojen tapauksessa käytetään nk. Transverse Mercator -karttaprojektiota kun taas eurooppalaisten aineistojen tapauksessa käytetään nk. Lambert azimuthal equal area -karttaprojektiota.



### EPSG

EPSG on tietokanta maantieteellisistä koordinaattijärjestelmistä ja ne määrittelevistä tiedoista ja parametriarvoista. EPSG-tietokanta määrittelee kullekin järjestelmälle kokonaislukuindeksin, jolla järjestelmään voidaan viitata. EPSG tulee sanoista European Petroleum Survey Group mutta EPSG-tietokantaa pitää nykyisin yllä öljyn ja kaasuntuottajien kansainvälisen yhdistyksen (OGP) geomatiikkakomitea.

Esimerkiksi EPSG:3067 viittaa ETRS-TM35FIN järjestelmään, joka on käytössä Suomessa. Koodissa ETRS viittaa tiettyyn Maan muodon malliin, TM karttaprojektioon ja 35 on k.o. projektion parametri. Useat paikkatietoja käsittelevät ohjelmat hyödyntävät EPSG-tietokantaa ja tunnistavat siten EPSG-indeksit.

Paikkatieto on tietoa jostain ilmiöstä tai kohteesta. Ilmiö liittyy silloin joko tiettyyn paikkaan tai sillä on arvo kaikkialla tai lähes kaikkialla (esimerkiksi ilman lämpötila, maanpinnan korkeus, asukastiheys) (ns. *maantieteellinen kenttä*). Arvo voi olla todellinen (kuten esimerkiksi lämpötila) tai laskennallinen (kuten esimerkiksi asukastiheys). Kohde tai sijainti on yleensä jollain tavoin maantieteellisesti rajattu. Maantieteellisesti rajatut ilmiöt tai kohteet voivat olla pistemäisiä, viivamaisia, aluemaisia tai näiden kombinaatioita (esim. pistejoukko).



### Huomaa

Geoinfomatiikan termistö on koottu geoinfomatiikan sanastoon: <http://www.tsk.fi/tiedostot/pdf/GeoinfomatiikanSanasto.pdf>

Teknisesti paikkatietoaineisto on käytännössä joko rasteri- tai vektoriaineisto. Rasteri tarkoittaa käytetyssä koordinaatistossa suorakulmaisen alueen täyttävää säännöllisistä suorakulmaisista soluista

muodostuvaa matriisimaista alkiojoukkoa (tai säännöllisessä hilassa olevia pisteitä, riippuen aineiston tulkinnasta; tulkintaan liittyy vielä se, tarkoittaako piste solun keskipistettä vai jotain muuta pistettä). Kuhunkin soluun (tai pisteeseen) liittyy numeroarvo, joka voi olla kokonaisluku, reaalityyppinen tai imaginääriluku. Numeroarvo viittaa rasterin kuvaaman paikkatiedon arvoon, eli se voi sinällään tarkoittaa jotain (esim. sademäärää millimetreinä keskimäärin solun alueella) tai sitten se voi olla viittaus johonkin taulukkoon, jossa arvo on kuvattu sanallisesti (esim. arvo tarkoittaa maankäyttötyyppiä ja arvo 12 tarkoittaa lehtimetsää). Voidaan myös asettaa, että tietty arvo (esim. -9999) tarkoittaa, että kyseisen solun alueelta ei ole tietoa. Rasteriaineisto voi sisältää useita kaistoja (kanavia) tietoa, jolloin samasta kohdasta on useita arvoja. Usempikaistainen aineisto syntyy esimerkiksi monikanavaisen kaukokartoitusinstrumentin tuottamana. Myös tavallinen värikuva on usein kolmikanavainen (RGB eli punaisen, vihreän ja sinisen kanavan sisältämä) rasteri. Myös vektorikenttiä (esim. tuulen nopeus ja suunta) kuvaavissa aineistoissa kuhunkin rasterin soluun liittyy moniulotteinen (esimerkiksi tuulen tapauksessa 2-ulotteinen: suunta ja voimakkuus) tieto. Rasteriaineistossa kohde (esimerkiksi järvi) on käytännössä yhtenäinen solujoukko, joka eroaa ympäristöstään solujen arvojen perusteella.



### Pikseli ja solu

Pikseli on pienin yksittäinen elementti rasterikuvassa tai kuvantamis- tai näyttölaitteessa. Solu taas viittaa struktuurin pienimpään yksittäiseen osaan. Paikkatietorasterin solu voi olla lähtöisin kuvantamislaitteen pikselistä mutta katseltaessa rasteria kuvaruudulta yksi solu ei välttämättä eikä useinkaan ole yhden pikselin avulla esitetty.

Vektorianeisto koostuu kohteista, joilla on jokin geometrinen muoto ja muita, ei-spatiaalisia ominaisuuksia. Geometrietieto on siten kohteen yksi ominaisuus. Geometrietietoon liittyy yleensä aina maantieteellinen/kartografinen referenssijärjestelmä eli geometrietiedon koordinaatit ovat k.o. referenssijärjestelmän koordinaatteja. Yleensä paikkatietokohteet on paikkatietoaineistoissa esitetty pisteiden, murtoviivojen, polygonien tai niiden yhdistelmien avulla. Näitä muotoja käytetään yksinkertaisuuden vuoksi, koska ne voidaan esittää pelkästään pisteiden ja niiden järjestyksen avulla ilman funktioita. Vektorianeistolla voidaan kuvata maantieteellisiä kenttiä esim. pistemäisten kohteiden (kentän arvo k.o. pisteissä) tai tasa-arvokäyrien (esim. maastokartan korkeuskäyrät) avulla.

Huomaa, että rasteriaineisto voidaan muokata vektorianeistoksi tietoa hävittämättä, kun taas vektorianeiston kuvaaminen rasteriaineistoksi yleensä hävittää tietoa.

Yleensä teknisissä vektoriformaateissa kohteet ovat toisistaan täysin erillisiä ja kunkin geometria on sen oma ominaisuus. Kohteiden *geospaatialinen topologia*, eli niiden spatiaaliset yhteydet on kuitenkin tärkeä osa todellisuutta. Esimerkiksi kahdella vierekkäisellä tontilla on osa rajapyykeistä samoja ja mikäli tontit tallennetaan kahtena polygonina, on vaara, että yhdellä rajapyykillä on kahdet, toisistaan poikkeavat koordinaatit. Kohteiden spatiaalisten yhteyksien yksityiskohtainen kuvaaminen kun niiden geometriat ovat ominaisuuksia (ja siksi erillisiä) on vaikeaa. Yksi vaihtoehto on käyttää kohteiden muita (lisä-) ominaisuuksia topologisten yhteyksien kuvaamiseen. Toinen vaihtoehto on luoda formaatti, jossa aineiston kattama alue jakaantuu täysin osa-alueisiin, jolloin saavutetaan esim. se etu, että kukin rajapiste on tallennettu vain kerran. Kolmas vaihtoehto on tehdä geometrioista omia entiteettejä (vrt. rajapyykki edellä olevassa esimerkissä) ja muuttaa kohteen ja geometrian välinen yhteys kahden entiteetin väliseksi yhteydeksi.

Paikkatietoaineistoon liittyy yleensä tieto siitä missä koordinaatistojärjestelmässä se on. Rasteriaineistossa on lisäksi solukoordinaattien ja karttakoordinaattien välisen muunnoksen

$$x_p = a_x + b_x X + c_x Y \quad (2.1)$$

$$y_p = a_y + b_y X + c_y Y \quad (2.2)$$

parametrit. Ylläolevassa kaavassa  $x_p$  ja  $y_p$  ovat solukoordinaatteja (sarake ja rivi) ja  $x$  ja  $y$  ovat kartta-koordinaatteja (itäinen ja pohjoinen; pituuspiiri ja leveyspiiri). Mihin kohtaan solua solukoordinaatti viittaa, on sopimuksenvarainen asia (ks. yllä).

Mittakaava on suhdeluku, joka kertoo kartan tai muun mallin ja sen esittämän todellisen kohteen mittojen välisen suhteen. Paikkatiedon (joka ei siis ole kartta vaan dataa, josta kartta voidaan tehdä) tapauksessa mittakaava kuvaa tallennetun tiedon sijainnin tarkkuutta, esityksen (esim. käytettyjen pisteiden määrä) tarkkuutta ja esitettyjen kohteiden laatua (esim. onko aineisto rakennuksia vai kaupunkeja).

### 2.2.3. Paikkatietoinfrastrukturi

Paikkatietoinfrastrukturi on yhteiskunnan tietoinfrastrukturi, jonka avulla paikkatietojärjestelmät (tai ohjelmat) voivat viestiä toistensa kanssa tai käyttää toistensa palveluita. Paikkatietoinfrastruktuurit on nykyisin usein laeilla määritelty.

Paikkatietoinfrastrukturiin liittyy palveluita, joiden kautta paikkatietoa voi etsiä (aineistoluettelopalvelu), katsella (katselupalvelu), käyttää ohjelmassa tai ladata jollekin omalle tallennusvälineelle (aineistopalvelu), muokata tai analysoida (prosessointipalvelu) ja ladata omalta tallennusvälineeltä palveluun ja tallentaa sinne (tallennuspalvelu). Kaikista palveluista voi periaatteessa olla kaksi eri versiota: ihmisille ja tietokoneohjelmille tarkoitetut.



#### INSPIRE

INSPIRE on Euroopan parlamentin ja komission vuonna 2007 asettama direktiivi, joka määrittelee Euroopan Yhteisön paikkatietoinfrastruktuurin. Direktiivi on toteutettu Suomessa lakina paikkatietoinfrastruktuurista (<http://www.finlex.fi/fi/laki/alkup/2009/20090421>), joka tuli voimaan 17.6.2009. INSPIRE nojaa vahvasti paikkatietopalvelustandardeihin, kuten WMS (Web Map Service eli karttapalvelu), WFS (Web Feature Service eli paikkatietokohdepalvelu) ja WCS (Web Coverage Service eli paikkatietorasteripalvelu) sekä Catalog Service (eli aineistoluettelopalvelu). Nämä standardit on tuottanut OGC (Open Geospatial Consortium <http://www.opengeospatial.org/>).

### 2.2.4. Talonrakennusalan tieto

IFC (Industry Foundation Classes) on talonrakennusalan tietomallistandardi ja tiedostoformaatti, jota kehittää buildingSMART-allianssi. Talonrakennus- ja kiinteistöalalla (AEC/FM Architecture, Engineering, Construction and Facility Management) tästä tietotekniikan alasta käytetään usein termiä rakennustietomallinnus (BIM, Building Information Modeling).

IFC sisältää kuvaukset ja määrittelyt sadoista erilaisista kohteista ja osista, joita rakennuskohde voi sisältää (kuten palkit tai ovet), sekä rakennusprosessiin ja -projektiin liittyvistä erilaisista *entiteeteistä* ja näiden välisistä yhteyksistä.

### 2.2.5. Aineistojen ominaisuuksista

Aineistoja tallennetaan, arkistoidaan, siirretään ja hyödynnetään. Aineistoihin liittyy monia institutionaalisia ominaisuuksia kuten tekijänoikeus, tietokantaoikeus, mahdollisesti jopa patenttioikeuksia jne. Aineiston käytön kannalta keskeisiä ominaisuuksia ovat mm. sen sisältämien tietojen tarkkuus ja se mitä tietoa aineistossa on. Kun aineistoa hankitaan tai sen käyttöä harkitaan täytyy nämä asiat selvittää aiotun käytön kannalta.

Tekniset ominaisuudet ovat usein ensimmäisiä vastaantulevia asioita aineistoa hankittaessa tai tarkasteltaessa. Jos aineisto on tallennettu formaattiin, jota ei tunnista tai pystytä millään käytössä olevalla ohjelmalla avaamaan, niin aineisto on käytännössä hyödytön.

Tiedostoja on kahta tyyppiä: teksti- ja binääritiedostoja. Tekstitiedostojen etu on se, että ne ovat yleensä ihmisen ja tietokoneohjelman helposti luettavissa ja sikäli lähtökohtaisesti ainakin jollain tavalla tulkittavissa, kun taas binääritiedostot on luotu esim. tallentamalla tietokoneen muistissa olevaa tietoa sellaisenaan, joten tieto on käytännössä pakko lukea kyseisen formaatin lukemiseen tarkoitetulla ohjelmalla. Tekstitiedostojenkin käytössä on omia hankaluuksiaan. Rivinvaihdon tunnistin on eri eri käyttöjärjestelmissä ja kirjaimet voi olla koodattu eri tavoin. Nykyisin käytetään usein nk. Unicode-standardia, jonka avulla voidaan esittää käytännössä mitä tahansa tekstiä millä aakkostolla tahansa. Unicoden yleisimmin käytetty tekninen toteutus on UTF-8, joka on taaksepäin yhteensopiva aieman ASCII-standardin kanssa.

Koodauksen lisäksi tiedosto voi olla pakattu eli tallennettu kompressoituna. Useita tiedostoja voidaan liittää yhteen paketiksi. Nykyisin käytetään usein XML-pohjaisia tallennustekniikoita (ks. kappale 2.4.1). Esimerkiksi toimisto-ohjelmien tuottamat dokumentit ovat nykyisin usein paketoitu joukko XML-tiedostoja.

Aineistojen tietojen tarkkuus eli aineistoon liittyvät epävarmuudet on monitahoinen asia. Esimerkiksi paikkatiedon tapauksessa tiedon epävarmuuteen liittyy mm. miten kohdeluokat määriteltä (esim. rakennukset vs. asuinrakennukset ja liikerakennukset), tieto luokitteluun liittyvistä epävarmuuksista (esim. kuinka iso osa havupuiksi määritellyistä kohteista on todennäköisesti kuitenkin lehtipuita), ja yksittäisen arvo tarkkuus (esim. mikä on pistemäisten kohteiden koordinaattien keskivirhe).

## 2.2.6. Esimerkkejä aineistoista

Maankäyttötietoaineisto on joko rasteri- tai vektoriaineisto, joka kuvaa maa-alueiden käyttötavat. Maanpeitetietoaineisto on sukua maankäyttötietoaineistolle ja eroaa siitä lähinnä siinä mielessä, että maankäyttö-sana viittaa siihen, että on olemassa jonkinlainen päätös siitä mihin eri maa-alueita käytetään. Maanpeitetieto on usein alunperin kaukokartoituksen avulla tuotettua. Maanpeitetieto voi olla rakenteellinen, esim. metsä -> lehtimetsä -> lehtimetsä, jonka puustotilavuus on 10-20 m<sup>3</sup>/aari. Samaa maankäyttöä (esimerkiksi pysäköintialue) voi vastata usea erilainen maanpeitetieto (pysäköintialue voidaan toteuttaa esimerkiksi asfaltoituna tai sorapäälysteisenä).

Valuma-alueet ja niiden osa-alueet on joko manuaalisesti maastokartalta tulkittu tai sitten hydrologisten algoritmien avulla (ks. kappale Rasteritekniikat alla) korkeusmallista laskettu.

Järvet, joet tai yleisesti vesialueet voivat olla maanpeitetietoaineistosta erotettuja tietoaineistoja.

Liikenneverkkoaineisto (tiet, kadut, kevyen liikenteen väylät, vesiliikennereitit) on yleensä vektoriaineisto, joka on joko erotettu yleisestä paikkatietoaineistosta, tai erikseen tuotettu. Tällaisessa aineistossa on yleensä katujen ja teiden sijaintien ja geometrioiden lisäksi paljon liikenne- ja osoitejärjestelmiin liittyvää tietoa. Liikenneverkkoaineisto on navigointisovellusten ja -laitteiden kannalta keskeistä, joten niiden rahallinen arvo ja käyttöarvo voi olla suuri.

Säättietoja tuotetaan säänmittausasemilla ja kaukokartoitusinstrumenttien avulla. Säättietojen yleinen ominaisuus on niiden muita aineistoja yleensä nopeampi päivittyminen.



### Shapefile

Shapefile on yleinen tekninen formaatti vektoripaikkatietoaineistoille. Shapefile on yhdysvaltalaisen yrityksen, Esrin, määrittelemä. Yksi syy sen yleisyyteen on se, että se on pääosin julkisesti kuvattu<sup>2</sup>. Toinen syy on, että se on suosittu paikkatietoohjelmiston ArcView:n (joka on nyt jo poistunut markkinoilta) pääasiallisesti käyttämä formaatti. Kolmas syy on se, että se on yksinkertainen formaatti, paikkatietokohteet ovat yksinkertaisia kohteita, joilla on geometrian lisäksi joukko muita ominaisuuksia. Shapefilessä on aina vähintään kolme tiedostoa: shp, shx ja dbf. dbf-tiedosto on dBase-formaatissa oleva tietokantataulutiedosto, jossa on kohteiden ei-spatiaaliset attribuutit. shp-tiedostossa on kohteiden geometriat ja shx-tiedostossa on kohteiden geometrioihin liittyvä indeksi. Shapefile-aineistoon voi näiden kolmen pakollisen tiedoston lisäksi kuulua muitakin tiedostoja, kuten käytetyn koordinaattijärjestelmän määrittelevä prj-tiedosto ja spatiaalinen indeksi (sbn ja/tai sbx -tiedostot)<sup>3</sup>.

## 2.3. Malleista ja menetelmistä

Mallilla tarkoitetaan tässä todellisuuden yksinkertaistettua kuvausta. Todellisuus on luonnollisesti loputtoman monimutkainen, mutta sitä voidaan ymmärtää ja hallita (tiettyynajaan asti) siitä saatavan tiedon avulla. Todellisuuden luonteen analysointi on perimmältään filosofinen ongelma, mutta tieteen ja tekniikan eri alueilla on kehitetty menetelmiä ja tekniikoita todellisuuden osa-alueiden kuvaamiseksi.

Mallin tarkoitus voi olla kuvata todellisuutta, jotta sitä voitaisiin ymmärtää paremmin tai saavutettu ymmärrys voitaisiin jakaa. Malli voi olla myös suunnitelma, kuinka tekninen järjestelmä tai joku sen osarakennetaan. Mallin avulla voidaan pyrkiä myös ennustamaan, selittämään, suunnittelemaan tai säätämään järjestelmiä tai niiden toimintaa. Yksi yleinen mallityyppi on fysikaalis-kemiallisen prosessin matemaattinen malli, jolla simuloidaan prosessin toimintaa. Toinen yleinen mallityyppi on dataa jollain matemaattisella konstruktiolla kuvaava malli (esim. keinotekoisia hermoverkkoja - jotka ovat matemaattisia konstruktioita - käyttäen luotu kuvaus tai ns. aikasarjamalli, joka kuvaa aikasarjan tilastollisia ominaisuuksia).

*Hahmontunnistus* perustuu malliin, joka tuottaa tietylle (monimutkaiselle) syönteelle tietyn (yksinkertaisen) tuloksen. Tulos voi olla nominaali- tai reaaliluku. Hahmontunnistus voi olla luokittelua, esimerkiksi sähköpostin luokittelua roskapostiksi ja asialliseksi postiksi. Luokittelumallin rakentaminen voi perustua opettamiseen (tunnetaan joukko syöte-tulos -pareja) tai ohjaamattomaan oppimiseen (jolloin mallin tuottamien luokkien merkityksen tulkinta jää mallin käyttäjälle). Hahmontunnistusta voidaan käyttää esim. ennustamiseen tai päätöksenteon tukemiseen.

*Tietomalli* on kuvaus jossain yhteydessä (organisaatiossa, projektissa, tietyssä suunnittelussa) käsiteltävästä tiedosta. Tietomalli muodostuu kohteiden ja kohteiden välisten yhteyksien kuvauksista. Kohteet määritellään käytännössä niiden ominaisuuksien avulla. Samanlaiset kohteet muodostavat kohdejoukon eli kategorian. Kahden kohteen välillä voi olla yhteys sen takia, että niiden kategorioilla on yhteys (esim. "moottoritie" on tietyn tyyppinen "tie" samoin "paikallistie" on "tie") tai jostain muusta syystä (esimerkiksi tie numero 12 risteää tien numero 4 kanssa joten niillä on spatiaalinen yhteys, tai esimerkiksi Pakila on osa Helsinkiä joten niillä on on-osa -yhteys). Tietomallin rakenteen määrää toisaalta käsiteltävä tieto, mutta toisaalta myös tiedon käyttötarkoitus.

<sup>2</sup> Shapefilen kuvaava dokumentti löytyy esim. osoitteesta: <http://shapelib.maptools.org/dl/shapefile.pdf>.

<sup>3</sup> Spatiaalinen indeksi mahdollistaa tietyn spatiaalisen predikaatin (ks. kappale Tietokannoista) mukaisten geometrioiden nopean poiminnan tietokannasta.



Kappaleiden tai kohteiden (kolmiulotteista) muotoa voidaan mallintaa erityyppisillä menetelmillä. Nämä mallit on tarkoitettu tuotantoa, visualisointia, analysointia jne varten. Nämä geometrisen mallinnuksen menetelmät muodostavat CAD:in eli tietokoneavusteisen suunnittelun (piirtämisen) perustan.



## Ontologia

Ontologia tietoteknisenä mallina on tietyn aihealueen perustermien ja niiden välisten yhteyksien määrittelydokumentti, joka sisältää myös säännöt, joiden mukaan sanaston termejä voidaan käyttää. Ontologia on tarkoitettu useiden tahojen hyväksymäksi ja käyttämäksi. Ontologia on tarkoitettu tietokoneohjelmien käyttöön. Ontologioita voidaan hyödyntää luonnollisen kielen käsittelyssä, tiedonhallinnassa, sähköisessä kaupankäynnissä, tietojärjestelmien integroinnissa jne.

Järjestelmää voi kuvata *arkkitehtuurimallilla*, joka kuvaa järjestelmän sisäisen rakenteen ja miten vastuut tai toiminnot jakautuvat niiden kesken. *Tietovirtakaavio* kuvaa kuinka tieto kulkee järjestelmän eri osien välillä.

*Kausaalinen malli* on kuvaus siitä miten jotkin asiat vaikuttavat toisiin asioihin (esim. kuinka tietyt elintavat vaikuttavat ihmisen terveyteen). *Prosessimalli* voi olla kuvaus siitä miten asiat tapahtuvat jossain prosessissa (esim. rakennuslupapäätöksen tekeminen), miten niiden pitäisi tapahtua (esim. kuinka kehityskeskustelu esimiehen ja alaisen välillä käydään jossain organisaatiossa) tai miten ne saattavat tapahtua (esim. kuinka ihmisen toiminta aiheuttaa ilmastonmuutoksen).



## DPSIR

DPSIR (Driving forces, Pressures, States, Impacts, Responses) on Euroopan ympäristöviraston omaksuma kausaalinen mallikehikko yhteiskunnan ja ympäristön välisen vuorovaikutuksen ja ympäristömuutoksen kuvaamiseen. Ajava voima on esim. teollinen tuotanto, paine on esim. jäteveden johtaminen ympäristöön, tila on esim. veden laatu vesistöissä, vaikutus on esim. virkistyskäytönmahdollisuuden heikkeneminen ja vastine on esim. vesistöjen suojeleminen.

*Matemaattinen malli* on matemaattinen kuvaus jostain järjestelmästä. Matemaattisia formalismeja, joilla kuvata todellisuutta, on useita. Lähtökohdaksi voidaan ottaa havaintoaineisto ja matemaattiseksi formalismiksi esimerkiksi tilastomatematiikka ja todennäköisyyslasku. Lähtökohdaksi voidaan ottaa myös luonnontieteellinen teoria havaintoaineiston tuottamasta järjestelmästä ja rakentaa matemaattinen kuvaus kyseeseen tulevien luonnontieteellisten teorioiden pohjalta. Tyypillinen matemaattinen malli on järjestelmän tilan muutosta ajassa kuvaava differentiaaliyhtälö. Lähtökohdaksi voidaan ottaa myös kokemuspohjaiseen tietoon perustuvia matemaattisia kuvauksia.

*Päätöksentekomalli* on menetelmä, jolla valitaan yksi tarjolla olevista vaihtoehdoista. Päätöksentekomalli voi olla kuvaileva malli, eli teoria tai kuvaus siitä miten päätöksentekijä todellisuudessa tekee päätöksensä tai se voi olla ohjaava, eli se voi antaa tietyssä mielessä parhaan (esim. olettamusten valossa todennäköisesti suurimman taloudellisen tuloksen tuottavan) vaihtoehdon.

Asiantuntijajärjestelmä perustuu malliin, jonkin aihealueen (asiantuntija)tiedosta. Tällainen malli perustuu usein asioiden totuusarvoihin ja niiden sekä asiantuntijalta saatujen sääntöjen perusteella päättelyyn. Yksinkertaisimmillaan totuusarvo on kaksiarvoinen: asia on joko tosi tai epätosi. Esimerkki yksinkertaisesta päättelystä on: "Jos asia on tosi, niin se ei ole epätosi". Loogisilla operaattoreilla (esim. "ei", "ja", "tai") voidaan rakentaa yksittäisistä proposiioista eli asioiden (lauseiden) totuusarvoista mo-

nimutkaisempia ilmaisuja (jotka ovat myös propositioita). Esim. ilmaisun "A tai B" totuustaulu on oikean taulukon mukainen. Huomaa, että logiikan "tai" ei ole sama kuin puhekielen "tai".

Taulu 2.1.

A	B	A tai B
epätosi	epätosi	epätosi
epätosi	tosi	tosi
tosi	epätosi	tosi
tosi	tosi	tosi

Kaksiarvoinen malli asioiden totuusarvosta on yleensä riittämätön ja tarvitaan ainakin kolmas vaihtoehto *määrittelemätön* tilanteelle, jossa asiaa ei yksinkertaisesti tunneta tms. *Sumea logiikka* on formalismi, jossa totuusarvo on reaalityttö nollan ja yhden välillä. Esimerkiksi lauseen "Pekka on pitkä mies" totuusarvo voi olla tarkastelijasta riippuen vaikkapa 0.1 tai 0.2.

Asia, joka joko tapahtuu tai ei tapahdu (kaksiarvoinen totuus, esimerkiksi auton käynnistyminen), voi kuitenkin tapahtua jollain todennäköisyydellä. Asioiden riippuvuus toisistaan voidaan kuvata mallina (esim. kausaalimallina, ks. yllä) ja siten rakentaa todennäköisyyksiin perustuva päättelymalli. Esimerkiksi auton käynnistymättömyys voi perustua useisiin syihin ja päättelymallia ja sen todennäköisyyksien päivittämistä vian selvittämisen yhteydessä voidaan hyödyntää ongelmanratkaisussa.

Nykyisin malleja pyritään usein integroimaan isommiksi kokonaisuuksiksi koska halutaan ymmärtää ja hallita suurempia ja/tai monimutkaisempia kokonaisuuksia. Integrointi edellyttää usein siihen sopivaa analyttistä menetelmää (esim. kausaalimalli) ja/tai tietoteknistä menetelmää (esim. tietojärjestelmien rakentaminen useita malleja yhdistelemällä).

### 2.3.1. Tietorakenteista

Tietorakenne on tapa tallentaa ja organisoida tietoa tietokoneeseen, jotta sitä voitaisiin käyttää tehokkaasti. Tietorakenne on tekninen ratkaisu tiedon tallentamisen ja käytön ongelmaan. Tietomallin tekninen toteutus hyödyntää jotain tai joitain tietorakenteita, mutta tietomallin rakenne on motivoitu sen perusteella miten se kuvaa ulkoista todellisuutta kun taas tietyn tietorakenteen käyttö motivoituu sen perusteella miten se sopii tiedon muotoon ja ajateltuun käyttöön. Usein yksi keskeinen tavoite tietorakenteiden (ja niihin liittyvien algoritmien) määrittelyssä on halutun tiedon nopea löytyminen.

On olemassa abstrakteja malleja ja tietorakenteita, jotka tunnetaan hyvin ja joihin liittyviä ongelmia ratkaisemaan on kehitetty algoritmeja. Yksi keskeinen abstrakti malli/tietorakenne on graafi eli verkko. Graafi muodostuu joukosta solmuja ja niitä yhteen liittäviä kaaria. Kaari voi olla suunnattu eli kaarilla voi olla suunta ja kaareen voi liittyä kustannus, jota voidaan käyttää kuvaamaan esim. etäisyyttä. Graafi on erityisen käyttökelpoinen abstrakti malli esim. liikennejärjestelmien, kausaalimallien tai projektien oleellisten piirteiden kuvaajana.

### 2.3.2. Algoritmi

Algoritmi on hyvin määritelty toimintatapa jonkin ongelman ratkaisemiseksi tai tehtävän suorittamiseksi. Tietotekniikassa algoritmeja kehitetään, jotta tehtäviä voitaisiin tehdä mahdollisimman nopeasti (esim. piirtää yksittäinen suora viiva tietokoneen näytölle) tai jotta niitä voitaisiin ylipäänsä (käytännössä, hyväksyttävässä ajassa) tehdä (esim. löytää jokin tietty tieto isosta tietojoukosta).

Tietorakenteisiin liittyy algoritmeja, jotka kuvaavat kuinka k.o. tietorakenteeseen lisätään tai siitä haetaan tietoa. Esimerkiksi yksinkertainen tietorakenne linkitetty lista koostuu joukosta tietoalkioita, jotka sisältävät tietokentän ja viittauksen joko listan seuraavaan alkioon tai sekä listan edelliseen että seu-

raavaan alkioon. Linkitettyyn listaan päästään käsiksi esimerkiksi sen ensimmäisestä alkioista. Linkitetyn listan tietyn alkion etsivä algoritmi lähtee liikkeelle listan ensimmäisestä alkioista ja käy läpi alkioita kunnes haluttu löytyy. Jos lista on järjestetty esimerkiksi aakkosjärjestykseen voidaan etsiminen lopettaa, kun tullaan kohtaan, jossa halutun alkion tulisi olla, muutoin koko lista täytyy käydä läpi.

Malleihin liittyy algoritmeja, jotka kuvaavat kuinka malleihin liittyviä lasketatehtäviä ratkaistaan. Esimerkiksi verkkoon liittyviä ongelmia ja niitä ratkaisevia algoritmeja on useita (ks. alla optimointi). Myös differentiaali- ja osittaisdifferentiaaliyhtälöiden numeeriseen ratkaisuun on useita menetelmiä.

### 2.3.3. Simulointi

Simulointi on todellisuuden jäljittelyä. Koska malli on todellisuuden kuvaus, simulointi perustuu yleensä simuloitavasta järjestelmästä tehtyyn malliin. Usein simulointiin liittyy aika eli tapahtumisen jäljittely, mutta ei välttämättä, voidaan esimerkiksi simuloida erilaisia maankäyttötilanteita. Simulointimalli voi pyrkiä kuvaamaan järjestelmän toimintaa tietyistä alkutilasta lähtien, kun ulkoinen pakote tunnetaan. Simulointia voidaan käyttää järjestelmästä olevan mallin testaamiseen vertaamalla simuloinnin antamia tuloksia havaittuihin arvoihin. Vuorovaikuteista simulointia voidaan käyttää todellisen järjestelmän sijasta opetuksessa, suunnittelussa tai vaikkapa viihteessä.

Simulointiin liittyy simulointimallin rakentaminen, mallin hyvyyden tarkistaminen ja mallin käyttö. Mallin rakentamiseen liittyy simuloitavasta järjestelmästä olemassaolevan tiedon (erityisesti havaintotiedon) ja tietämyksen (kokemusperäinen tieto, teoriat) kokoaminen, matemaattisen mallin rakenteen ja ominaisuuksien valitseminen, mallin parametrien arvojen valitseminen tai estimoiminen havaintotiedon perusteella ja mallin toteuttaminen tietokoneessa. Mallin hyvyyden tarkastelu perustuu siihen, että osa havaintotiedosta jätetään käyttämättä mallia rakennettaessa ja mallin antamia tuloksia verrataan niihin.

Simulointimallin rakentaminen voi perustua joko alhaalta ylöspäin (bottom-up) -lähestymistapaan tai ylhäältä alaspäin (top-down) -lähestymistapaan. Alhaalta ylöspäin -lähestymistavassa pyritään etsimään järjestelmän toiminnalle keskeisiä perussääntöjä ja malli rakennetaan usein koostumaan lukuisasta joukosta perusyksiköitä, jotka toimivat näiden sääntöjen mukaisesti (vrt. SimCity). Tällainen malli on usein kokeellinen ja mallinnuksessa on tällöin tutkimuksellinen ote. Ylhäältä alaspäin -lähestymistavassa pyritään kokonaisuuden ymmärtämiseen ja mallin mahdollisimman yksinkertaiseen (vähän parametrejä) rakenteeseen. Tällaisessa mallissa tärkeää on mallin validisuus eli oikeellisuus ja luotettavuus.



#### Occamin partaveitsi

Occamin partaveitsi viittaa periaatteeseen, jonka mukaan ilmiötä selitettäessä on pyrittävä mahdollisimman yksinkertaiseen kuvaukseen. Mallinnukseen sovellettuna tämä tarkoittaa pyrkimystä mahdollisimman yksinkertaiseen malliin, jossa on mahdollisimman vähän parametreja.

Mallia ei oikeastaan koskaan voida todistaa "oikeaksi". Malli voidaan ainoastaan osoittaa vääräksi tai todeta hyödylliseksi.



## Karl Popper ja falsifioitavuus

Karl Popper esitti, että kaikki inhimillinen tieto, tieteellinen tieto mukaan luettuna, on aina hypoteettista, josta syystä mitään teoriaa tai mallia ei voida koskaan todistaa lopullisesti oikeaksi, mutta toisaalta yksikin teorian tai mallin vastainen esimerkki riittää osoittamaan sen vääräksi. Toisaalta myöskään mitään teoriaa, jota ei edes teoriassa voida osoittaa vääräksi, ei voida pitää tieteellisenä.

### 2.3.4. Optimointi

Optimointi tarkoittaa parhaan vaihtoehdon valitsemista. Optimointiin voidaan joutua turvautumaan jos joukko, josta valittava vaihtoehto on valittava on äärettömän suuri tai rajoitettu, mutta käytännössä liian suuri, jotta kaikki vaihtoehdot voitaisiin yksityiskohtaisesti tutkia esim. simuloimalla.

Matemaattisessa optimoinnissa vaihtoehdon hyvyys määrittyy matemaattisen lausekkeen arvona. Tuon lausekkeen (kohdefunktion) kirjoittaminen on osa optimointimallin rakentamista. Kohdefunktion lisäksi optimointimalli voi sisältää mahdollisesti suurenkin joukon yhtälöitä ja epäyhtälöitä (rajoitusehtoja), joiden tulee olla voimassa, jotta vaihtoehto on käypä. Vaihtoehto on yksinkertaisesti päätösmuuttujalle (tai päätösmuuttujille) annettava arvo (arvojoukko).

Yksinkertaisessa tapauksessa optimointi on funktion ääriarvojen hakemista. Jos kohdefunktion  $f(x)$  lisäksi tehtävässä on yhtälörajoituksia  $g(x) = 0$  ja funktio  $f$  ja  $g$  ovat derivoituvia, voidaan käyttää Lagrangen kertojatekniikkaa. K.o. tekniikassa muodostetaan funktio  $L(x, \lambda) = f(x) + \lambda(g(x))$ , jonka jokin kriittinen piste (pisteet, joissa funktion osittaisderivaatat ovat nolliä) on alkuperäisen tehtävän ratkaisu.

Varsinaisia optimointimenetelmiä on lähinnä kaksi: lineaarinen optimointi ja dynaaminen optimointi. Näiden lisäksi on iteratiivisia menetelmiä, jotka etsivät parasta vaihtoehtoa generoimalla aina uusia vaihtoehtoja, jotka pyrkivät olemaan parempia kuin edellinen. Iteraatio voi perustua funktion gradienttiin (uusi vaihtoehto haetaan suunnasta, jossa kohdefunktion arvo nopeimmin kasvaa tai vähenee), johonkin heuristiikkaan (kokemusperäinen sääntö) tai vaikkapa luonnonvalintaa jäljittelevään laskentaan. Iteratiivinen menetelmä ei välttämättä takaa parhaan mahdollisen ratkaisun löytymistä vaan se voi pysähtyä esimerkiksi johonkin paikalliseen ääriarvoon.

Lineaarisen optimoinnin suosio (ja osittain koko menetelmä) perustuu alkujaan George Dantzigin vuonna 1947 kehittämään Simplex-algoritmiin. Lineaarisessa optimoinnissa kohdefunktio on muotoa  $c^T x$ , jossa  $c$  on tunnettu kerroinvektori ja  $x$  on päätösmuuttujat sisältävä vektori (molemmat pystyvektoreita, joten kohdefunktion arvo on skalaari), ja rajoitusehdot ovat muotoa  $Ax \leq b$ , jossa  $A$  on tunnettu kerroinmatriisi ja  $b$  on tunnettu vakiovektori. Usein lineaarisen optimointitehtävän ratkaiseva ohjelmakoodi pystyy ratkaisemaan myös ongelman, jossa jokin tai jotkin päätösmuuttujista on rajoitettu (pakotettu) kokonaisluvuiksi. Käytännössä lineaarista optimointia käytettäessä ongelma kuvataan ratkaisun etsivälle ohjelmalle erityisellä mallinuskielellä (esimerkkejä alla), jonka avulla voidaan käyttää päätösmuuttujille ja vakioille kuvaavia nimiä, vakioita voidaan laskea toisista vakioista ja lausekkeita voidaan kirjoittaa havainnollisesti käyttäen esimerkiksi jos-niin tai summa -rakenteita.

Dynaaminen optimointi on tapa ratkaista monimutkainen ongelma jakamalla se samanlaisiin osiin. Jos ongelma muodostuu yksinkertaisemmista, keskenään samanlaisista ongelmista, jotka on ratkaistava peräkkäin (ajassa tai paikassa), niin silloin ongelman tehokkaasti ratkaiseva algoritmi voi olla mahdollista kehittää dynaamisen optimoinnin periaatteella.

Dynaamisen optimoinnin periaate on yksinkertaisesti se, että riippumatta siitä tilasta missä kulloinkin ollaan, optimaalinen päätös on se, jonka seurauksena päädytään tilaan, jossa ollaan optimaalisella po-

lulla. Kaikkiin mahdollisiin tiloihin kaikissa mahdollisissa päätösvaiheissa liittyy siis optimaalinen päätös (ja optimikustannus). Yksittäinen ongelma on siis valita mahdollisista päätöksistä se, johon liittyyvä välitön kustannus plus tilaan, johon päädyttäisiin, liittyvä kustannus (mahdollisesti diskontattuna) on pienin.

Graafeihin tai verkkoihin liittyy monia optimointiongelmiä, kuten lyhyimmän polun ongelma, kauppatkustajan ongelma ja verkoston välityskykyyn liittyvät ongelmat. Lyhyimmän polun ongelmassa tavoitteena on löytää halvin tapa kulkea graafin solmusta A solmuun B, kun kaariin liittyy tietty kustannus. Kauppatkustajan ongelmassa tavoitteena on löytää halvin reitti, joka käy kaikissa graafin solmuissa yhden kerran. Verkoston välityskyky -ongelmissa tavoitteena on esim. löytää halvin tapa kuljettaa tietty määrä tavaraa solmusta A solmuun B, kun kaariin liittyy tietty kustannus ja välityskyky. Monien graafeihin liittyvien ongelmien ratkaisua voidaan lähteä hakemaan tarkastelemalla graafin yksittäistä solmua ja kirjoittamalla siihen liittyvät rajoitusehdot matemaattisesti. Dijkstran kuuluisa algoritmi vuodelta 1959 ratkaisee tietyn lyhyimmän polun ongelman<sup>4</sup>, kun taas kauppatkustajan ongelman ratkaisemiseksi ei ole onnistuttu löytämään tehokasta ratkaisualgoritmia.

Dynaamisten graafien (graafien joihin syntyy uusia ja joista kuolee solmuja ja kaaria) tarkastelu johti 1990-luvulla dynaamisten verkottuneiden järjestelmien tutkimukseen, jolla on sovelluksia esimerkiksi energian tuotanto ja jakelu -järjestelmien tutkimuksessa ja hallinnassa.

### 2.3.5. Laskennallisten mallien toteuttamisesta tietokoneessa

Ohjelmointikieli on väline, jolla laskennallinen malli toteutetaan tietokoneessa. Ohjelmointikieliä on lukematon määrä erilaisia ja niitä voidaan ryhmitellä eri tavoin. Laskennallisten mallien näkökulmasta tarkastellen voidaan ehkä erottaa yleiskäyttöiset ohjelmointikieliset erityisesti laskentaan tai mallintamiseen suunnitelluista kielistä.

Usein käytettyjä yleiskäyttöisiä ohjelmointikieliä ovat esimerkiksi Fortran ja Python. Fortran on standardoitu (standardeja on tosin vuosien varrella kirjoitettu useita), alunperin jo 1950-luvulla määritelty ohjelmointikieli, joka on kuitenkin edelleen suosittu tehokkuutensa takia tieteellisessä tietojenkäsittelyssä ja numeerisessa laskennassa. Python on kohtalaisen uusi ja ns. korkean tason kieli. Pythonia ei ole varsinaisesti standardoitu, vaan siitä on olemassa avoimen ja vapaan lähdekoodin periaatteella kehitetty referenssi-implementaatio. Fortranin ja Pythonin yksi perusero on niiden perusrakenteosien abstraktiotasossa. Voidaan ajatella, että Python lähtee ihmisen, ohjelmoijan ajattelusta ja Fortran lähtee enemmän tietokonelaitteiston ominaisuuksista. Toinen perusero on Pythonin helppous käytön kannalta. Python-ohjelman ja muiden vastaavien ns. skriptauskielten kääntäminen konekielille ja ajaminen tapahtuu yhdellä käskyllä<sup>5</sup>, kun Fortran-kielinen ohjelma täytyy erikseen kääntää ja linkittää. Molemmilla kielillä ohjelmoitaessa tarjolla on suuri joukko kirjastoja, joiden avulla yleisesti tarvittuja toiminnallisuksia voidaan ottaa käyttöön omissa ohjelmissa. On myös mahdollista hyödyntää esim. Fortranilla kirjoitettua ohjelmakirjastoa Python-ohjelmassa erityisen k.o. ohjelmakirjastoa varten kirjoitetun Python-spesifin liityntäkomponentin avulla.

Suosittuja erityisesti laskentaan tarkoitettuja ohjelmointikieliä ovat esimerkiksi MATLAB ja R. Molemmat ovat erityistarkoituksestaan huolimatta myös yleiskäyttöisiä korkean tason ohjelmointikieliä. Vastaavasti myös esim. Pythonille on olemassa kirjastoja, joiden avulla sillä voidaan ohjelmoida hyvin samaan tapaan kuin esimerkiksi MATLABilla. MATLAB on kaupallinen tuote, jolla on muutamia vapaita vastineita, joista tunnetuin on Octave. R taas on vapaa ohjelmistoprojekti, jonka kaupallinen vastine on S-PLUS. R ja S-PLUS ovat tilastollisen ohjelmointikielen S toteutuksia. S on määritelty kirjasarjassa<sup>6</sup>. MATLAB ja Octave ovat samankaltaisia, mutta niiden toteuttamalla ohjelmointikielillä ei ole erillistä auktoriteettia. Laskennallisten mallien toteuttamista tukevissa ohjelmointikielissä peruslähtökohtana

<sup>4</sup> Voidaan osoittaa, että Dijkstran algoritmi on itse asiassa dynaamisen optimoinnin sovellus.

<sup>5</sup> Käytännössä Python-kielinen ohjelma käännetään ns. tavukoodiksi, joka suoritetaan eräänlaisessa virtuaalikoneessa.

<sup>6</sup> Chambers, J. 1998. Programming with Data: A Guide to the S Language. Springer.

on yleensä vektorien ja matriisien käytön tukeminen sekä erilaisten yleisesti käytettyjen funktioiden tukeminen.

Korkean tason ohjelmointikielille on tyypillistä, että niitä voidaan käyttää vuorovaikutteisesti, eli käyttäjän antama ohjelmakoodi tai käsky voidaan kääntää ja ajaa ikäänkuin se olisi osa samaa ohjelmaa, joka myös luo vuorovaikutteisuuden.

Taulukkolaskentaohjelmat ovat suosittuja visuaalista ohjelmointia tukevia tietokoneohjelmia. Taulukkolaskentaohjelma toimii yleensä vuorovaikutteisesti ja muuttujien (solujen) väliset yhteydet huomioiden. Taulukkolaskentamallia muutettaessa muutoksen vaikutukset välittyvät soluista toisiin soluihin, jotka viittaavat niihin. Taulukkolaskentaohjelmien ohjelmointikielet tukevat yleensä aika samantyyppisiä käsitteitä kuin korkean tason ohjelmointikieletkin eli esimerkiksi assosiaatiotauluja ja operaatioita taulukoilla/matriiseilla. Taulukkolaskentaohjelmissa voi olla mahdollisuus käyttää taulukko-ohjelmointikielen lisäksi erillistä ohjelmointikieltä. Tällä erillisellä ohjelmointikielillä voidaan tehdä laajennuksia taulukkolaskennan ohjelmointikielen tai niillä voidaan ohjelmoida ns. makroja, jotka automatisoivat esimerkiksi taulukkojen rakentamisen tai muita toimintoja, jotka muuten pitäisi tehdä manuaalisesti. Taulukkolaskentaohjelmissa on myös yleensä datan visualisointityökaluja - vastaavasti kuin joissain korkean tason ohjelmointikielten vuorovaikutteiselle käytölle rakennetuissa ohjelmointiympäristöissä - ja erilaisia laajennuspaketteja esimerkiksi optimointia varten. Taulukkolaskentaohjelmia on kritisoitu siitä, että ne kannustavat hyvin yksilölliseen laskentamallien toteutukseen, jota on vaikea muiden tulkita tai jonka toimintaa on muiden vaikea ymmärtää. Taulukkolaskentamallit voivat myös helposti mennä rikki, koska niitä on helppo muokata. Sama ongelma koskee luonnollisesti myös muita ohjelmia, joita käyttäjän on helppo muokata (Python, MATLAB, R, jne).



### Assosiaatiotaulu

Assosiaatiotaulu on tietotyyppi, joka koostuu avain, arvo -pareista. Kutakin avainta, joka on yleensä merkkijono, vastaa yksi arvo, joka voi olla mikä vain kyseisen ohjelmointikielen perustietotyyppi, jopa viittaus toiseen assosiaatiotauluun jos ohjelmointikieli tukee viittauksia.

Mallinnuskielet (omana erityisryhmänään) ovat simulointimallien, optimointimallien tai muuntotyypisten mallien rakentamiseen ja ajamiseen tai muuhun tarkoitukseen määritellyjä erikoistuneita tietokonekieliä. Ne eivät yleensä ole yleiskäyttöisiä ohjelmointikieliä. Nämä kielet voivat olla myös visuaalisia tai toteutettu osaksi graafista käyttöympäristöä. Kaikkiin edellämainittuihin (yleiskäyttöisiin ohjelmointikieliin, laskentakieliin, taulukkolaskentaohjelmiin) on myös saatavissa erilaisia simulointimallintamista ja muuta mallintamista tukevia lisäpaketteja. Monet mallinnuskielistä ovat kaupallisia tuotteita. Esimerkkeinä voidaan mainita Simulink (<http://www.mathworks.com>) ja STELLA (<http://www.iseesystems.com>), jotka on tarkoitettu dynaamisten järjestelmien simulointimallien tekemiseen, ja AMPL (<http://www.ampl.com>) ja LINGO (<http://www.lindo.com>), jotka ovat optimointimallien kirjoittamiseen ja ratkaisemiseen tarkoitettuja kieliä sekä HUGIN (<http://www.hugin.com/>) ja Netica (<https://www.norsys.com/>), jotka ovat Bayes-verkkojen ja vaikutuskaavioiden määrittelyyn ja käyttöön tarkoitettuja ohjelmistoja.

## 2.4. Tietokannoista

Tietokannat ovat järjestettyä tietoa, koottuna teknisen järjestelmän hallinnoitavaksi ja säilytettäväksi pidemmäksi aikaa. Tietokantajärjestelmällä on useita tehtäviä.

- Mahdollistaa uusien tietokantojen luominen ja niiden teknisen tietomallin määrittely *tiedonmäärittelykielillä*.

- Mahdollistaa tehokas tiedonhaku tietokannasta *tiedohakukielellä* sekä tietokannassa olevan tiedon muokkaaminen *tiedonhallintakielellä*. Tietoa lisättäessä tai muokattaessa tietokantajärjestelmän on varmistettava, että lisätty tieto on tietomallin mukaista.
- Mahdollistaa moniosaisten (esim. yhdeltä tililtä nostaminen ja toiselle tilille paneminen) toimenpiteiden (ns. transaktioiden) suorittaminen joko kokonaan tai ei ollenkaan ja siten ylläpitää tiedon kokonaisuuden luotettavuutta.
- Tallentaa tieto luotettavasti.
- Mahdollistaa ja hallita useiden käyttäjien samanaikainen pääsy eri tietokantoihin tai samaan tietokantaan. Järjestelmän tehtävä on estää useiden käyttäjien yhtäaikaisten toimenpiteiden johtaminen tiedon korruptoitumiseen.

Hallitseva tietokantajärjestelmätekniikka on relaatiotietokanta, mutta myös ns. oliopohjaisia tietokantajärjestelmiä käytetään jonkin verran. Relaatiotietokannan keskeinen käsite on relatio eli taulu. Yleisesti ottaen entiteetti kuvataan tauluksi tietokannassa ja entiteetin yksinkertaiset ominaisuudet taulun sarakkeiksi. Relaatiotietokannassa tietomalliin kuuluvat entiteettien väliset yhteydet varmistetaan ns. avainten avulla. Pääavain on taulun kenttä tai joukko kenttiä, joka yksilöi taulun rivit. Viiteavain on taulun kenttä tai joukko kenttiä, joka viittaa toisen taulun pääavaimen. Relaatiotietokantaan ei voi tehdä muutosta, joka lisäisi johonkin tauluun rivin, jolla ei ole uutta arvoa pääavaimelle. Relaatiotietokantaan ei voi myöskään tehdä muutosta, joka on viiteavaimen vastainen eli poistaisi pääavaimen arvon jota käytetään tai lisäisi viiteavain-kenttään arvon, jolle ei ole vastaavaa pääavainta.

Relaatiotietokantajärjestelmässä kaikki tieto, myös pääosa itse tietokannan ylläpitoon liittyvistä tiedoista (mitä tauluja on, keitä käyttäjiä on jne), on tauluissa. Relaatiotietokantoja hallitaan yleisesti SQL:ksi (Structured Query Language) kutsutulla kielellä, jossa on välineet sekä tietokantojen luomiseksi että tiedon hakemiseksi ja muokkaamiseksi.

Tietokoneohjelmissa tietoa hallitaan usein ns. oliomalliin perustuen. Oliomallinnukseen keskeisesti liittyvä luokkien määrittely, erityisesti ns. periytyvyys luokkien välisenä yhteytenä on hankalaa toteuttaa relaatiotekniikan avulla. Tästä johtuen on kehitetty kokonaan oliopohjaisia tietokantatekniikoita mutta niiden käyttö on jäänyt vähäiseksi. Joissain relaatiotietokantajärjestelmissä voi määritellä taulun perimään jonkin toisen taulun ominaisuudet ja siten tuoda hieman oliopohjaisuutta tietokantaan.



## SQL

SQL on deklarativinen tietokonekieli, eli sillä ilmaistaan se mitä halutaan, eikä (kuten imperatiivisilla kielillä, joita useat ohjelmointikieliset ovat) se miten joku asia halutaan tehtävän. Deklaratiivisuudesta seuraa se, että joissain tapauksissa voi olla vaikeaa ennustaa kuinka kauan jokin toimenpide kestää. SQL on osittain standardoitu kieli, mutta käytännön SQL-toteutukset poikkeavat toisistaan monissa suhteissa ja monissa toteutuksissa on paljon varsinaisen standardin ulkopuolisia laajennuksia kieleen ja toisaalta sama asia voidaan ilmaista usealla eri tavalla samassakin toteutuksessa.

Relaatiotietokantatekniikkaa käytetään sekä pienissä järjestelmissä, joissa varsinainen tietokanta voi olla yhdessä tiedostossa että isoissa järjestelmissä, joissa tietokanta voi olla sitä tarkoitusta varten pyhitetyssä tietokoneessa tätä tarkoitusta varten alustetussa tallennuslaitteessa tai monistettuna useissa koneissa ja joita hallinnoiviin järjestelmäohjelmiin otetaan yhteyttä tietoverkon kautta.



### Huomaa

Tietokantatekniikasta voi lukea lisää esimerkiksi sivulle <http://www.cs.helsinki.fi/u/laine/tika-pe/k10/matsku.shtml> linkitetyistä materiaaleista.

### 2.4.1. XML

XML (Extensible Markup Language) on yleinen etenkin viestinnässä ja monimutkaisten dokumenttien tallentamisessa käytetty tekniikka. XML:ää ja siihen liittyviä tekniikoita kuvataan tässä kappaleessa vaikka ne eivät ole varsinaisesti tietokantatekniikkaa.

XML (Extensible Markup Language) on tiedontallennustekniikka, jossa tieto tallennetaan tekstitiedostona (joka voidaan sitten kyllä tallentaa pakattuna binääritiedostona tilan säästämiseksi tai tiedonsiirron nopeuttamiseksi). XML-dokumentilla on puumainen rakenne, jossa solmu voi aina sisältää yhden tai useamman solmun. XML-dokumentti itse on rakenteen juurisolmu. Solmulla voi olla ominaisuuksia, jotka ovat avain, arvo -pareja. Nämä molemmat ovat yksinkertaisia merkkijonoja. Yksinkertaisin solmu sisältää pelkkää tekstiä. Solmulla on nimi (tag), joka tallennetaan aina solmun mukana.

**<solmu ominaisuus="arvo">solmun sisältö</solmu>**

XML:ää käytetään yleensä sovelluksena (tässä sovellus ei tarkoita tietokoneohjelmaa). XML-sovellus on määrittely minkänimisiä solmuja voi olla ja mitä ominaisuuksia niillä voi olla ja mitkä solmut voivat olla minkäkin solmujen sisällä jne. Mikäli käyttäjä tietää mistä sovelluksesta (yksi XML-dokumentti voi käyttää useampaa sovellusta) on kyse ja lisäksi on varma että dokumentti on määrittelyjen mukaista (eli on ns. validi, jonka tarkistamiseen on työkaluja) helpottuu dokumentin sisältämien tietojen haku ja käyttö huomattavasti (voi olla että käyttäjä esim. tarvitsee vain tiettyä yhtä tietoa dokumentista).



### DocBook

DocBook on XML-sovellus kirjojen, artikkeleiden, dokumentaation tms. tekemiseen. Tämä oppikirja on kirjoitettu käyttäen DocBook-skeemaa, jonka ansiosta siitä on ollut helppo tuottaa sekä PDF, että HTML -versiot.

Paikkatietotekniikassa yksi keskeinen XML-sovellus on GML (Geographic Markup Language), joka on vektoridataformaatti.

#### XML-skeema

XML-skeema on dokumentti, joka kuvaa XML-dokumentin (sovelluksen) rakenteen ja sisällön. XML-skeeman määrittelyyn on muutamia standardoituja kieliä, esimerkiksi XML Schema.

#### XQuery, XPath

XQuery ja XPath ovat kyselykieliä tiedon hakemiseen XML-dokumentista. XPath on XQueryn osajoukko ja XQuery sisältää välineitä myös tiedon muokkaamiseen.

#### XSLT

XSLT (Extensible Stylesheet Language Transformations) on kieli, jolla XML dokumentista voi luoda uuden XML-dokumentin tai muun dokumentin.



## 2.4.2. Teknisen tietomallin määrittely

Tekninen tietomalli tarkoittaa relaatiotietokannan yhteydessä kuvausta relaatioista eli tauluista ja niiden välisistä yhteyksistä.

Tietomallin päätarkoitus on määrittellä ja lopulta taata tietokannassa olevan tiedon rakenne. Tietomalliin kirjoitettavat säännöt estävät sääntöjen vastaisen tiedon lisäämisen tietokantaan. Tämä on tärkeää tietokannan käyttäjän näkökulmasta, koska näin hän voi luottaa siihen, että tieto on ilmoitetun rakenteen mukaista, joka taas mahdollistaa esimerkiksi halutun tiedon hakemisen.

### CREATE TABLE

Taulu luodaan komennolla, joka on tyypillisesti muotoa **CREATE TABLE [taulun nimi] ( [kentän määrittely, ...] );** Kentän määrittely on muotoa **[nimi] [tietotyyppi] [rajoitukset]** (rajoitukset-osa voi puuttua). Erilaisia tietotyyppiä on suuri määrä ja tuetut tyypit vaihtelevat tietokantajärjestelmätyypeittäin. Yleisiä tyyppiä ovat kokonaisluvut, reaalityypit, merkkijonot, ajat ja päiväykset. Rajoituksista tärkeimmät ovat kentän asettaminen taulun pääavaimeksi (**PRIMARY KEY**) ja kentän asettaminen viittaamaan jonkin toisen taulun pääavaimeen (**REFERENCES [taulun nimi] ([kentän nimi])**). Muita rajoituksia ovat mm. kentän oletusarvo tai yleinen tarkistusehto **CHECK ([looginen testi])**.

### DROP TABLE

Taulu (ja moni muukin alkio tietokantajärjestelmässä) poistetaan komennolla drop: **DROP TABLE [taulun nimi];** Jos poistettavaan tauluun liittyy toisia tauluja viittauksilla niin taulun poistaminen ei välttämättä ole niin yksinkertaista, mahdollista tai toivottavaa.

Tietokannan tietomalli on mahdollista luoda SQL-kielisellä koodilla yhdellä kerralla (ajamalla k.o. koodi tietokantaan). Sellaisen koodin alussa on usein alussa komennot taulujen poistamiseksi koska olemassaolevan nimistä taulua ei voi luoda. On myös mahdollista luoda tietomalli tietokantatyökaluilla vuorovaikutteisesti. Mikäli tietokannassa on jo tietoa ja tietokannan rakennetta halutaan hieman muuttaa voidaan uusia tauluja luonnollisesti luoda tarpeen mukaan ja jo olemassaolevia tauluja voidaan muuttaa komennolla **ALTER TABLE [taulun nimi] [muutos];** Muutos voi olla **ADD COLUMN [kentän nimi]** tms.

## 2.4.3. Tiedon muokkaus

Tietokantatauluihin voidaan lisätä tietoa ja olemassaolevaa tietoa voidaan muuttaa ja poistaa. Tiedon lisääminen, muuttaminen tai poistaminen ei onnistu, jos uusi tieto rikkoisi tietomalliin kirjattuja rajoitusehtoja.

```
INSERT INTO [taulun nimi] VALUES ([arvo],...);
```

```
INSERT INTO [taulun nimi] ([kentän nimi],...) VALUES ([arvo],...);
```

Mikäli kentälle ei anneta arvoa, se saa oletusarvonsa tai mikäli sellaista ei ole se saa arvon NULL. NULL tarkoittaa puuttuvaa tietoa SQL:ssä.



### Huomaa

NULL:in olemassaolo SQL:ssä tarkoittaa, että SQL:n logiikka on itse asiassa kolmiarvoista eli lausekkeella voi olla kolme arvoa: tosi, epätosi ja NULL. Tämä on otettava huomioon kun SQL:llä rakennetaan kyselyjä. Huomaa erityisesti, että kentän arvon olemista NULL ei voi testata näin "kenttä" = 'NULL' vaan näin "kenttä" IS NULL.

Merkkijono on arvona rajattu yksinkertaisilla lainausmerkeillä: 'tämä on merkkijono'. Joissain SQL-murteissa (kuten harjoituksissa käytettävässä PostgreSQL:ssä) taulun tai kentän nimi voi sisältää erikoismerkkejä mutta silloin siinä käytetään kaksinkertaisia lainausmerkkejä: **INSERT INTO "Oma tauluni" ("kenttä") VALUES ('merkkijono');**

**UPDATE [taulun nimi] SET [kentän nimi] = [arvo] WHERE [ehto];**

Yllä **WHERE [ehto]** ei ole pakollinen mutta jos se poistetaan, niin taulun kaikkien tietueiden kyseiset kentät saavat annetun arvon. Jos taulussa ei ole pääavainta ja siinä on kaksi samanlaista tietuetta on käytännössä vaikeaa ellei mahdotonta muuttaa vain toista niistä. Ehto on kenttiä sisältävä lauseke, joka saa totuusarvon. Koska SQL:ssä on NULL, niin SQL:n logiikka on kolmiarvoista (siinä on tosi, epätosi ja tuntematon eli NULL).

**DELETE FROM [taulun nimi] WHERE [ehto];**

## 2.4.4. Tiedon haku

Tiedon haku SQL:ssä tapahtuu SELECT-komennon avulla.

**SELECT [kenttälista] FROM [taululauseke] WHERE [ehto];**

Ylläolevassa **WHERE [ehto]** ei ole pakollinen mutta usein käytännöllinen (**select \* from valtava\_taulu**; ei ole erityisen järkevää isojen taulujen tapauksessa). Kenttälustassa voi viitata taululausekkeen taulujen kenttiin ja siinä voi kenttien yksinkertaisen luettelemisen lisäksi rakentaa niistä lausekkeitä, joissa voi käyttää erilaisia funktioita. Taululauseke on yksinkertaisimmillaan yhden taulun nimi, mutta se voi myös olla yhdistelmä (JOIN) useasta taulusta. Yksinkertaisin yhdistelmä on kaksi taulua pilkulla erotettuna, joka muodostaa (virtuaalisesti) taulun, jossa on kaikki mahdolliset kombinaatiot taulujen tietueista (jos taulussa a on 3 tietuetta ja taulussa b on 2 tietuetta, niin taulussa a,b on 6 tietuetta).



### Predikaatti

Matemaattisessa logiikassa predikaatti on lauseke, joka on joko tosi tai epätosi kullekin joukon alkijolle. SQL:ssä tiedon päivitys, poisto ja haku kohdistuvat oletusarvoisesti taulun (joka voi olla virtuaalinen) kaikkiin tietueisiin, mutta tätä joukkoa voidaan rajata määrittelemällä predikaatti, joka saa arvo tosi vain halutuilla tietueilla.

## 2.5. Paikkatiedosta ja paikkatietomenetelmistä

Aiemmin on kerrottu lyhyesti mitä paikkatieto on, minkälaista paikkatietoa on ja minkälaisissa teknisissä tallennusmuodoissa paikkatietoaineistoja on sekä paikkatietoinfrastruktuurista. Tässä kappaleessa keskitytään pääasiassa paikkatietoon tietokannassa ja siihen minkälaisia asioita se mahdollistaa. Li-

säksi käydään lyhyesti läpi menetelmiä, jotka soveltuvat rasterimuotoiselle paikkatiedolle. Lopuksi kuvataan lyhyesti kolmioverkko ja voronoidiagrammi.

Relaatiotietokannoissa ei yleensä ole valmiina tukea paikkatiedolle, vaan käytännössä paikkatiedon tallentaminen vaatii paikkatietolaajennuksen tietokantajärjestelmään. Paikkatietolaajennus lisää tietokantajärjestelmään ainakin geometria-tietotyyppin (paikkatietokohteiden geometrioille) ja funktioita ja operaattoreita joilla geometria-tyyppisiä kenttiä voi käyttää ehtolausekkeissa ja muualla.

OGC on tuottanut standardin relaatiotietokantajärjestelmän ja SQL:n paikkatietolaajennukselle (<http://www.opengeospatial.org/standards/sfs>), joka perustuu saman organisaation yksinkertaisten paikkatietokohteiden arkkitehtuuriin (<http://www.opengeospatial.org/standards/sfa>).



### Huomaa

Tässä kappaleessa on kuvattu lyhyesti paikkatietoanalyyseja ja esitetty ne ikäänkuin tietokantatekniikan laajennuksena. Tietokantatekniikka on erinomainen perusta paikkatietojen hyödyntämiselle mutta vastaavasti ohjelmointikieliä voidaan hyödyntää lisäämällä niihin paikkatieto- ja paikkatietoanalyyseja (käytännössä niissä voidaan käyttää samoja ohjelmakirjastoja geometrisille operaatioille kuin tietokantajärjestelmissä). Analyyttisiä mahdollisuuksia voidaan tuoda edelleen käyttäjäystävällisempään muotoon rakentamalla graafisia sovelluksia (laajentamalla yleiskäyttöisen paikkatieto-ohjelman graafisen käyttöliittymän toiminnallisuutta) tietokantatekniikan tai ohjelmointikielten mahdollisuuksien varaan.



### Huomaa

Paikkatietoanalyyseista on verkossa hyvä kirja osoitteessa <http://www.spatialanalysisonline.com>

## 2.5.1. Yksinkertaiset paikkatietokohteet

Paikkatietokohteiden geometriaominaisuuksien kategoriat muodostavat luokkahierarkian OGC:n yksinkertaisten kohteiden mallissa.<sup>7</sup> Yleisimpänä kategoriana on Geometria, joka on abstrakti kategoria (todelliset geometriat kuuluvat aina johonkin konkreettiseen kategoriaan). Geometrialla on yhteys koordinaatistojärjestelmään, joten geometrian maantieteellinen sijainti on mahdollista määrittellä.

Geometrian välittömät alikategoriat ovat Piste, Käyrä, Pinta ja GeometriaJoukko. Näistä vain Piste on konkreettinen kategoria. Pisteellä on koordinaatit x ja y. Lisäksi pisteellä voi olla koordinaatit z ja m. z-koordinaatti on pisteen korkeusasema ja m-koordinaatti on mitta-arvo. m-koordinaatti on yleensä vain pisteillä, jotka ovat osa käyrää tai pintaa. m-koordinaatti on mielivaltaisen mitta, esimerkiksi sitä voidaan käyttää kertomaan pisteen todellinen (eikä datan edustamaa murtoviivaa pitkin laskettu) etäisyys käyrän alkupisteestä.

Paikkatietoaineistoissa (toisin kuin CAD-aineistoissa, joissa käytetään yleisesti splinejä) konkreettiset käyrät ovat murtoviivoja eli järjestetyllä pistejoukolla esitettyjä käyriä. Vastaavasti alueet ovat sulkeutuvien murtoviivojen rajaamia alueita. Alueella on ulkoreuna ja mahdollisesti yksi tai useampia "reikiä" eli sulkeutuvien murtoviivojen rajaamia alueita, jotka eivät ole k.o. aluetta.

<sup>7</sup> Graafinen esitys luokkahierarkiasta on esim. osoitteessa [http://ibis.colostate.edu/WebContent/IBIS/BlueSpray/UsersGuide/Script\\_SimpleFeatures.html](http://ibis.colostate.edu/WebContent/IBIS/BlueSpray/UsersGuide/Script_SimpleFeatures.html)

### 2.5.2. Spatiaaliset yhteydet

Kahden geometrian välisiä yhteyksiä tutkivat funktiot ovat yleensä käyttökelpoisia hakuehdossa. Spatiaaliset yhteydet on määritelty tietyn 3 x 3 matriisin (ns. DE-9IM matriisi ks. esim. [http://edndoc.esri.com/arcscde/9.1/general\\_topics/understand\\_spatial\\_relations.htm](http://edndoc.esri.com/arcscde/9.1/general_topics/understand_spatial_relations.htm)) avulla, joka saadaan tutkimalla geometrioiden sisäpuolen (interior), reunan (boundary) ja ulkopuolen (exterior) leikkausten dimensioita. Tyhjän joukon dimensio on -1, pisteen dimensio on 0, viivan 1 ja alueen 2.

Yksinkertaisten kohteiden standardin mukaisia spatiaalisia predikaatteja ovat equals, disjoint, intersects, touches, crosses, within, contains, overlaps (on yhtenevä, on erillinen, leikkaa, koskettaa, menee yli, on sisäpuolella, sisältää, on osittain päällekkäinen - nämä eivät välttämättä ole virallisia suomennoksia), Näistä on lyhyet sanalliset kuvaukset mm. PostGIS paikkatietolajennuksen dokumentaatioissa ovat osoitteessa [http://postgis.refrains.net/documentation/manual-1.5/reference.html#Spatial\\_Relationships\\_Measurements](http://postgis.refrains.net/documentation/manual-1.5/reference.html#Spatial_Relationships_Measurements).

### 2.5.3. Spatiaaliset mitat

Spatiaalisilla mitoilla voidaan tehdä yksinkertaisia analyyseja. Mitta-funktioita ovat Distance, Length, Area (etäisyys, pituus, pinta-ala - nämä eivät välttämättä ole virallisia suomennoksia).

### 2.5.4. Funktiot joiden tulos on geometria

Funktiot, joiden tulos on geometria, mahdollistavat monipuolisimpia analyyseja. Tällaisia funktioita ovat Envelope, Boundary, LocateAlong, LocateBetween, Buffer, ConvexHull, Intersection, Union, Difference, SymDifference (verhokäyrä, reuna, paikanna pitkin, paikanna välissä, puskurialue, konvekssi kuori tai vaippa, leikkaus, unioni, erotus, symmetrinen erotus - nämä eivät välttämättä ole virallisia suomennoksia). Envelope palauttaa pienimmän suorakaiteen, jonka sisälle argumenttigeometria mahtuu. Pisteiden boundary on tyhjä joukko, viivan boundary on sen päätepisteet ja alueen boundary on sen reunaviivat. Buffer palauttaa alueen, jonka pisteet ovat enintään annetulla etäisyydellä argumenttigeometriasta. Konvekssi vaippa on alue, jonka rajaa argumenttigeometrian ympärille viritetty "kuminauha".

Kartografian kannalta tärkeä funktio on sellainen, jonka avulla alunperin usean pisteen avulla esitetty murtoviiva voidaan yksinkertaistaa pienimittakaavaisia karttoja varten. Ns. Douglas-Peucker-algoritmi on yleisesti käytetty algoritmi tällaisen funktion toteuttamiseen.

Spatiaalinen interpolointi tuottaa pisteistä koostuvasta aineistosta kentän, jonka arvo eri paikoissa määräytyy (yleensä k.o. paikkaa lähellä olevien) pisteiden arvojen avulla. Interpolointimenetelmiä on useita.

Lyhyimmän polun graafissa laskeva funktio perustuu optimointiin. Funktiolle annetaan lähtötietoina graafi, jossa on suuntaamattomat linkit ja niillä ei-negatiiviset kustannukset, ja lähtösolmu. Algoritmi, joka perustuu Dijkstran vuonna 1959 julkaisemaan algoritmiin, laskee pienimmät matkakustannukset lähtösolmusta graafin jokaiseen solmuun.

### 2.5.5. Rasteritekniikat

Rasterimuotoinen paikkatieto on hyvin yleistä, koska esimerkiksi satelliittikaukokartoitus tuottaa rasterimuotoista tietoa ja koska rastereiden käsittely on tietoteknisesti helppoa. Tyypillisiä rasteripaikkatietoaineistoja ovat maankäyttö- tai maanpeitetietoaineisto ja maasto- eli korkeusmalli.

Paikkatietorastereiden muokkaamiseen ja analysoimiseen on joukko yleisiä menetelmiä. Näiden lisäksi ja koska paikkatietorasteri ja rasterikuva ovat (ainakin teknisesti) samankaltaisia, monia kuvankäsittelytekniikoita voidaan soveltaa paikkatietorastereihin.

Paikkatietorasterimenetelmät voidaan jakaa menetelmiin, jotka tarkastelevat aina kerrallaan yhden solun aluetta (local), solun ja sen naapuruston aluetta (focal), yhtä osa-aluetta (zonal) tai koko tarkastel-

tavaa aluetta (global). Kussakin menetelmässä tai operaatiossa lähtötietoina on aina yksi tai useampi rasteri. Jos operaatio käsittelee useampaa rasteria kerrallaan, täytyy siis käytännössä niiden koordinaattien olla samat. Käytännössä niiden täytyy myös yleensä kattaa sama alue ja niiden solujen täytyy olla samankokoisia ja -muotoisia.

Solun naapurusto voidaan määritellä eri tavoin. Yksinkertaisia naapurustoja ovat ns. 4-naapurusto, johon kuuluu solun kanssa reunaviivan jakavat solut, ja 8-naapurusto, johon kuuluvat solun ympärillä olevat solut. Osa-alueen taas määrittää käytännössä erillinen kokonaislukurasteri, jossa kunkin alueen soluilla on sama lukuarvo. Koko alue taas ei välttämättä ole koko rasterin kattama suorakulmainen alue, vaan esimerkiksi maskirasterin (joka on 0/1 -rasteri) rajaama alue.

Yhden solun aluetta kerrallaan tarkastelevia operaatioita ovat mm. matemaattiset operaatiot (yhteen- ja vähennyslasku jne), loogiset operaatiot (vertailut, loogiset operaatiot olettaen että operandit ovat 0/1-rastereita, ja jos-niin operaatiot), uudelleenluokittelut (solun uusi arvo on entisen arvon funktio - yleensä yksinkertainen kuvaus).

Naapurustoa kerrallaan tarkastelevia operaatioita ovat mm. naapuruston suurin arvo, kuinka monta kertaa alueella esiintyy tietty arvo jne. Monet maastomalleja (korkeusmalleja, DEM digital elevation model) analysoivat menetelmät ovat naapurustoa-analyysija: maaston kaltevuus, kaltevuuden suunta, veden virtaussuunta jne.

Osa-alueita ja koko aluetta käsittelevät operaatiot eivät oikeastaan tuota tuloksena rasteria vaan assosiaatiotaulukon, joka on indeksoitu osa-alueiden arvoilla (koodeilla).

Korkeusmallien hydrologinen analyysi on tyypillinen ja paljon tutkittu paikkatietorasteritekniikka. Kullekin solulle voidaan laskea (menetelmiä on useita ja ne antavat hieman erilaisia tuloksia) veden virtaussuunta, joka on yleensä yksi (voi olla useampikin) 8:sta naapurista tai määrittelemätön (kun solu on paikallinen minimi tai alue on tasainen). Kun paikalliset minimi- ja tasaiset alueet on käsitelty niin, että kaikilla soluilla on selkeä virtaussuunta, voidaan virtaussuuntarasterin avulla päätellä valuma-alueet ja niiden jakautuminen osa-alueiksi sekä uomien sijainnit ja uomaverkosto (oikeastaan uomat muodostavat yleensä puumaisen, ei verkostomaisen rakenteen). Virtaussuuntarasterista voidaan laskea esimerkiksi virtauspolkujen (rinteiden) pituuksia.

Kahden tai useamman rasterin yhdistäminen laskennallisesti on (laskennallista) päällekkäistämistä (overlay). Graafista päällekkäistämistä käsitellään alla paikkatietojärjestelmien yhteydessä.

Teknisesti rasterimenetelmien toteuttaminen on helppoa ja käyttö nopeaa, kun rasterien koko on kohtuullinen. Mikäli käsiteltävät rasterit ovat niin suuria, että tietokoneen työmuisti ei riitä, rastereita täytyy käsitellä osina ja osien välisten yhteyksien käsittely on hankalaa (analyysissa, joissa naapurustolla on merkitystä).

Usein paikkatietoaineistoja täytyy muuntaa vektorimuodosta rasterimuotoon tai rasterimuodosta vektorimuotoon. Ensimmäinen on periaatteessa helppoa koska se on periaatteessa tietokonegrafiikkaa, johon on laaja joukko tehokkaita menetelmiä. Jälkimmäinen on hankalampaa, mutta esimerkiksi yhtenäisten solualueiden muuntaminen polygoneiksi on kohtalaisen helppoa.

## 2.5.6. Kolmioverkko ja Voronoi-diagrammi

Kolmioverkko on tietorakenne, jolla voidaan kuvata kenttä. Kolmioverkko koostuu pisteistä (x, y) ja niiden muodostamista kolmioista.

Kolmioverkko on tessellaatio, eli pinnan jako alueisiin ilman päällekkäisyyttä ja rakoja.

Delaunay-kolmiointi on sellainen pistejoukosta muodostettu kolmiointi, jossa minkään kolmion kulmapisteiden kautta piirretyn ympyrän sisällä ei ole pisteitä. Delaunay-kolmiointi maksimoi kolmioinnin jokaisen pienimmän kulman. Delaunay-kolmioinnin duaali on ns. Voronoi-diagrammi.



### Huomaa

Geofysiikassa ja meteorologiassa Voronoi-diagrammia kutsutaan Thiessen-polygoneiksi.

Voronoi-diagrammi jakaa pinnan pisteiden avulla monitahokkaiseen (polygoneihin) niin, että kuhunkin pisteeseen liittyy monitahokas, jonka muodostavat ne pisteet, jotka ovat k.o. pistettä lähinnä. Voronoi-diagrammin virittävät pisteet ovat sen Delaunay-kolmioinnin, jonka duaali se on, kolmioiden kulmapisteiden kautta piirrettyjen ympyröiden keskipisteet.

### 2.5.7. Kuvien georeferointi

Kuvan (oikaistu ilmakeku tai satelliittikuva, skannattu paperikartta tms.) georeferointi tarkoittaa se kuvakoordinaattien ja (tiettyjen) karttakoordinaattien välisen yhteyden määrittämistä. Yksinkertaisimmillaan tämä tapahtuu hakemalla kuvasta kohtia, joiden sijainti karttakoordinaatistossa tunnetaan. Kun tällaisia pisteitä on riittävästi, voidaan koordinaattien välille estimoida matemaattinen malli. Tämän jälkeen käydään läpi haluttu alue karttakoordinaatistossa halutulla hilalla ja lasketaan mallin avulla hilapisteen sijainti kuvassa ja poimitaan siitä arvo karttakuvaan.

## 2.6. Suunnittelun tietojärjestelmistä

Tässä kappaleessa tarkastellaan tietojärjestelmä-käsitettä ja muutamia tietojärjestelmiin liittyviä asioita. Lisäksi esitellään muutamia ympäristösuunnittelun kannalta keskeisiä tietojärjestelmätyyppejä. Usein yksittäisen tietokoneohjelmiston ja tietojärjestelmän välinen ero on epämääräinen.

CAD (Computer-aided desing) on esimerkki aiemmin manuaalisesta toiminnasta (piirtämisestä/suunnittelusta), joka on siirretty tietokoneeseen.

### 2.6.1. Tietojärjestelmä

Tietojärjestelmä on joukko tietokoneohjelmia (sovelluksia) ja (määritelmästä riippuen) niiden (organisointia) käyttöä. Tietojärjestelmä ja sen käyttö voi olla hyvin organisointia (esim. ydinvoimalan, lennonohjauksen tai teho-osaston potilaan hallintaan käytetty tietojärjestelmä) tai vapaamuotoista. Tietojärjestelmään kuuluu tietokonelaitteistoja ja -ohjelmistoja, tietoliikenneverkko, tietoa, ihmisiä ja toimintatapoja. Tietokone tai tietoliikenneverkko ei ole välttämätön osa tietojärjestelmää.

Tietojärjestelmiä voi tarkastella niiden arkkitehtuurin näkökulmasta. Arkkitehtuuri koostuu osista ja niiden välisistä kytkennöistä. Tietojärjestelmässä on useita osia, jotka kukin tuottavat oman tietyn toiminnallisuuden kokonaisuuteen. Erilaisia toiminnallisuuksia ovat esimerkiksi tiedon tallennus, piirustusten tekeminen ja (visualisointien) tuottaminen, viestintä, erilaisten dokumenttien tuottaminen, tiedon haku, tiedon muokkaaminen, sovellusten tekeminen, tilastollinen analyysi, simulointien ajaminen, optimointien tekeminen, karttojen tekeminen jne. Tietojärjestelmän osien väliset kytkennät ovat tiedon siirtoa tai palveluiden käyttöä varten.

Tietojärjestelmiä voi tarkastella niiden tyypin tai tarkoituksen näkökulmasta. Erilaisia tietojärjestelmätyyppejä ovat esimerkiksi tapahtumankäsittelyjärjestelmä (tapahtuma voi olla esim. lähetys saapunut, opintosuoritus tehty tms.), toimistojärjestelmä, päätöksenteon tukijärjestelmä ja tietämyksenhallintajärjestelmä. Tietojärjestelmätyyppi on oikeastaan pelkkä idea tai nimike, jonka taustalla on näkemys hyödyllisestä yleisen tason toiminnallisuuskokonaisuudesta, kuin konkreettinen ohje rakentaa tietojärjestelmä. Tietojärjestelmät ovat keskeisiä organisaatioiden toiminnalle ja tieto- ja kommunikaatioteknologian (laitteiden ja ohjelmistojen) merkitys niissä on koko ajan kasvanut ja fyysisen työn vähentynyt.

Tietojärjestelmiä voi tarkastella niiden hallitseman tiedon ja sen virtojen näkökulmasta. Tietojärjestelmään tulee jostain tietoa, esimerkiksi mittaustoiminnasta, tavaroiden tms. ostamisesta ja myymisestä, ihmisten palkkaamisesta ja palkanmaksusta, jätehuollosta, tienpidosta jne. Järjestelmässä tätä tietoa prosessoidaan ja tallennetaan. Tietojärjestelmä tuottaa jotain tietoa, esimerkiksi karttoja, toimintaohjeita, yhteenvetoja jne.



## Dashboard

Dashboard (suomeksi auton kojetaulu tms.) on yleistyvä käsite, jolla kuvataan tapaa esittää tietojärjestelmän tuottamaa tai hallitsemaa tietoa käyttäjälle. Dashboard liittyy yleensä johtamista tai päätöksentekoa tukeviin tietojärjestelmiin. Dashboardissa on yleensä joukko erilaisia graafisia esityksiä (visualisointeja) tiedosta.

Tietojärjestelmät ovat toisaalta välttämättä sidoksissa todellisuuteen mutta toisaalta niiden kehitystä on leimannut siirtyminen entistä kokonaisvaltaisemmin tietoteknologian avulla tapahtuvaan tietojenkäsittelyyn. Esimerkiksi jos aikaisemmin tietty asia hoidettiin tietyntyyppisillä lomakkeilla, niin ensimmäisessä vaiheessa lomakkeet siirrettiin sellaisinaan tietokoneisiin ja ne saatettiin tulostaa työprosessin aikana ja myöhemmin jälleen viedä tietokoneeseen. Tässä kehitysvaiheessa tietokoneella käsitellyt dokumentit lähinnä toistavat lomakevaiheen toimintatapoja ja muotoja. Seuraavassa kehitysvaiheessa on mietitty pidemmälle sitä, mitä tietoa asian hoitaminen oikeastaan vaatii ja prosessi on ehkä muokattu tietokannassa tapahtuviksi askeliksi. Käyttäjää varten saatetaan edelleen tuottaa paperilomakkeen näköinen tai jopa tulostettava dokumentti, mutta itse tieto hallitaan varsinaisesti tietokoneessa tietomallin avulla.

Tietojärjestelmien eräs keskeinen tekninen piirre on sovellusten välinen kommunikaatio. Kommunikaatio voi tapahtua teknisesti hyvin eri tavoin ja sovellukset voivat olla samassa tietokoneessa tai fyysisesti hyvin etäällä toisistaan. Sovellusten väliselle kommunikaatiolle on monia ja monenlaisia standardeja, jotka mahdollistavat eri tahojen tekemien sovellusten toimimisen yhdessä. Kommunikaatio voi perustua tiedostoihin tai se voi perustua sovellusten keskustelemiseen suoraan keskenään. Kommunikaatio edellyttää että eri sovellukset käyttävät (ainakin osin) samanlaisia tietomalleja (samaa sanastoa tai ontologiaa) tai että tietoa voidaan siirtää tietomallista toiseen.

### 2.6.2. Tietoturvallisuudesta

Tietoturvallisuus on tiedon, tietojenkäsittelyn ja tietojärjestelmien suojaamista vaaraa, vauriota, häviämistä, rikollista toimintaa ja muuta haittaa mahdollisesti aiheuttavaa vastaan.

Suojattavia asioita ovat tiedon luottamuksellisuus, tiedon eheys, tiedon aitous ja tietojärjestelmän käytettävissä olo.

### 2.6.3. Tietojärjestelmien laadusta

Tietojärjestelmien laatua voidaan tarkastella usean mittarin kuten esimerkiksi käytettävyyden, yhteentoimivuuden, ylläpidettävyyden jne kannalta.

Käytettävyys tarkoittaa helppokäyttöisyyttä, johon liittyy mm. opittavuus ja intuitiivisuus.

Yhteentoimivuus tarkoittaa sitä kuinka hyvin tietojärjestelmän komponentteja voidaan kytkeä toisiinsa ja kuinka hyvin tiedon eheys säilyy, kun sitä siirretään komponentista toiseen.

Ylläpidettävyys tarkoittaa mm. sitä kuinka hyvin järjestelmä voidaan muuttaa vastaamaan uusiin vaatimuksiin ja kuinka hyvin viat voidaan eristää ja korjata.

### 2.6.4. Tietomallin ja prosessikuvausten yhteydestä

Tekninen tietomalli on tietokannan toteuttamisen perusta. Prosessikuvaus (tämä ei tarkoita tietojärjestelmän vaan ei-tietoteknisen järjestelmän prosessikuvausta) on simulointi- ja optimointimallien toteuttamisen perusta. Tietokannan ja simulointimallin yhdistäminen on tietojärjestelmätason ongelma. Perinteisesti nämä kaksi teknologiaa tai tieteenhaaraa ovat kehittyneet paljolti toisistaan erillään. Prosessikuvaus kuitenkin edellyttää jonkinlaista järjestelmäkuvausta, jossa taas tietokantatekniikka voisi olla hyödyksi. Erilaisten parametrikombinaatioiden tai lähtötietojen hallinta on tietokannassa helpompaa. Myös käytännön tehtävät voivat edellyttää simulointimallin operatiivista käyttöä, jolloin tietokannan käyttö voi helpottaa tulosten hallintaa.

### 2.6.5. Tietojärjestelmätekniikoita

*Visualisointi* on laeva termi, mutta tässä yhteydessä sillä tarkoitetaan numeerisen tai ei-numeerisen tiedon graafista esittämistä. Kartta on paikkatiedon visualisointi tiettyyn tarpeeseen. Visualisointiin liittyy paljon visualisoinnin käyttäjän tarpeiden analysointia jne.

*Virtuaalitodellisuus* on simuloinnin avulla tuotettu todellisuutta tai fantasiaa jäljittelevä kokemus.

*Lisätty todellisuus (augmented reality)* on tekniikka, jolla esim. kameran tuottamaan kuvaan lisätään jokin visualisointi, joka lisää esityksen hyödyllisyyttä.

### 2.6.6. Paikkatietojärjestelmä

Paikkatietojärjestelmä (GIS, Geographic Information System) on tietojärjestelmä, jonka tarkoitus on hallita paikkatietoa, mahdollistaa erilaisten analyysien tekeminen k.o. paikkatietoon nojaten ja mahdollistaa erilaisten visualisointien kuten karttojen tehokas tuottaminen. Sanoja paikkatietojärjestelmä ja paikkatieto-ohjelmisto käytetään aika usein löyhästi samasta asiasta.

Aiemmin yksi paikkatietojärjestelmien keskeisistä toiminnallisuuksista oli paperikarttojen tai paperisten ilmakuvien siirtäminen ohjelmistoon eli digitointi. Digitointiin on kaksi tekniikkaa: 1) skannaus ja kuvaruudulla digitointi ja 2) digitointipöydän käyttö. Skannaus tuottaa rasterin ja kuvaruudulla siitä merkitään pisteitä, joiden koordinaatit tunnetaan. Digitointipöydän avulla tuotetaan suoraan vektoridataa.

Digitoinnille sukua oleva tekniikka on *geokoodaus*, jossa tieto muutetaan paikkatiedoksi muuntamalla epäsuorat sijainnin määrittelyt suoraksi sijainniksi. Sijainti voidaan epäsuorasti määritellä esim. osoitteen avulla.

Paikkatietojärjestelmissä aineisto esiintyy yleensä tasona. Järjestelmän esittämä karttakuva on tasojen visuaalinen yhdistelmä, jossa ylempänä oleva taso voi osittain tai kokonaan peittää alemman tason. Taso voi olla osittain läpinäkyvä.

Paikkatietojärjestelmän graafista käyttöliittymää voidaan käyttää vektoriaineiston sisältämien kohteiden hakuun, tarkastelemiseen ja muokkaamiseen.

Paikkatietojärjestelmissä on yleensä menetelmiä paikkatietoaineistojen muuntamiseksi eri tavoin. Muuntaminen voi olla liittyä aineiston tallennusmuotoon, koordinaattijärjestelmään, tai esimerkiksi aineiston tekniseen esitysmuotoon (vektorista rasteriksi tai päinvastoin).

Toistaiseksi melko kehittymättömiä toiminnallisuuksia paikkatietojärjestelmissä ovat mm. tehokas suurimittainen paikkatietolaskenta (geocomputation) ja spatiaalinen simulointi (geosimulation).

Kartta on perinteinen paikkatietovisualisointi. Kartan automaattinen tuotanto paikkatiedosta ei ole yksinkertainen asia ja visuaalisesti hyvän ja käyttökelpoisen kartan tekeminen vaatii yleensä ihmisen työ-



panosta. Verkossa olevat kartat<sup>8</sup> ovat yleensä automaattisesti paikkatietoaineistoista tehtyjä. Kartalla on mittakaava, joka kuvaa missä suhteessa luonnossa esiintyvä etäisyys on kuvattu karttaan. On myös karttoja, joissa ei ole mittakaavaa vaan asioiden esittäminen perustuu johonkin muuhun seikkaan. Kartta on myös tasainen kaksikulotteinen kuvaus (paitsi pallokartta ja kohokuvakartta) kaartuvasta ja maastoltaan vaihtelevasta todellisuudesta. Mittakaavan ja kartan tarkoituksen perusteella karttaan on valittu esitettävät kohteet ja niiden esitystapa. Kartoissa käytetään yleisesti symboleja ja niille eri kokoja ja värejä yms. kuvaamaan erilaisia asioita.

### 2.6.7. Päätöksenteon tukijärjestelmä

Päätöksenteon tukijärjestelmä on tietojärjestelmä, jonka tarkoitus on tukea päätöksentekoa.



#### Päätöksenteko

Päätöksenteko on prosessi, jossa tunnetuista vaihtoehdoista valitaan se, jonka mukaan toimitaan. Valinta perustuu vaihtoehtojen ja niiden vaikutusten ja seurausten vertailuun. Päätöksentekoon liittyy usein epävarmuutta, koska ei voida täsmälleen tietää, mitä seuraukset ovat eikä voida aina olla varmoja siitäkään, onko paras vaihtoehto tunnettujen vaihtoehtojen joukossa. Päätöksentekoon liittyy aina subjektiivisia asioita.

Päätöksentekoa voidaan tukea informaatiolla eri vaihtoehdoista ja niiden vaikutuksista ja seurauksista. Suosittuja menetelmiä ovat visualisointi ja esim. simulointimalleilla tehdyt ennusteet sekä aiempiin tapauksiin vertaaminen. Myös päätöksenteon prosessia, johon liittyy tavoitteiden ja vaihtoehtojen analyysia ja vertailua, voidaan tukea.

Johtamisjärjestelmien kirjossa päätöksenteon tukijärjestelmä käsitetään yleensä ylempien toimihenkilöiden työkaluna, kun alimpana on tapahtumien hallintajärjestelmät, niiden yläpuolella tapahtumien johtamisen tietojärjestelmät, niiden yläpuolella päätöksenteon tukijärjestelmät ja kaikkein ylimpänä strategisen johtamisen tukijärjestelmät (termit eivät liene vakiintuneita).

### 2.6.8. Tietokoneavusteinen suunnittelu

Tietokoneavusteinen suunnittelu (CAD, Computer-aided design) on tietotekniikan hyödyntämistä laitteen, teknisen järjestelmän, tuotantolaitoksen tms. suunnittelussa. Suunnitteluun kuuluu suunnittelun kohteen olemuksen, muodon, toiminnan jne. luominen ja määrittelemine ja suunnittelun kohteen esittäminen niin että se voidaan tuottaa tai rakentaa.

Perinteisesti suunnitelma on paperilla esitetty piirustus ja siihen liittyvät laskelmat ja komponenttien mitoitus. Tietokoneavusteinen suunnittelu on pitänyt näitä kahta suunnittelun puolta pitkään erillään mutta tietomallitekniikan kautta ne voidaan pitkälti yhdistää. Tietomallitekniikan perusajatus on kohteen suunnitelman luominen tietomallien ja niiden pohjalta luotujen entiteettien avulla. Suunnitelma luodaan käytännössä tietokantaan, jota erilaiset visualisointiohjelmat ja laskentamallit käyttävät.

## 2.7. Tiedonhankinnasta

Tässä kappaleessa esitellään lyhyesti pääasiassa geomatiikan alaan kuuluvat tiedonhankintamenetelmät. Geomatiikalla tarkoitetaan yleensä paikkatiedon hankinnan, käsittelyn, mallintamisen, analyysin

<sup>8</sup> Karttapalveluita ovat mm. <http://maps.google.com>, <http://www.bing.com/maps>, <http://www.openstreetmap.org> ja <http://www.paikkatietoikkuna.fi/web/fi/kartta>.

ja hallinnan tekniikoita. Geomatiikka kattaa maanmittauksen, joka on vanhempi termi, ja paikkatietotekniikan monin osin.

### 2.7.1. Satelliittipaikannus

Maailmanlaajuisen satelliittipaikannusjärjestelmän (GNSS Global Navigation Satellite System) avulla vastaanottimet voivat paikantaa itsensä (määrittää pituus- ja leveyspiirinsä ja korkeusasemansa) ja määrittää tarkan ajan. Tällaiset vastaanottimet ovat viime vuosina halventuneet, pienentyneet ja niiden tehonkulutus on laskenut niin, että sellainen voidaan sisällyttää esimerkiksi matkapuhelimeen. Yhdysvaltalainen GPS (Global Positioning System) on keskeinen mittaus ja kartoitusjärjestelmä. Yhdysvaltain armeija käynnisti GPS-ohjelman vuonna 1973. 1980-luvulla GPS-järjestelmä avattiin yleiseen käyttöön. Vuonna 2000 avoimesta järjestelmästä poistettiin tarkoituksellinen paikannuksen tarkkuuden heikennys, jonka jälkeen sen tarkkuus on noin  $\pm 3\text{...}15$  metriä. GPS:ää vastaavia järjestelmiä ovat venäläinen GLONASS, EU:n Galileo ja Kiinan Compass. Ainoastaan GPS on tällä hetkellä maailmanlaajuisessa operatiivisessa käytössä.

GPS sisältää 24 satelliittia, jotka kiertävät maata noin 20 000 kilometrin korkeudessa. Satelliitit lähettävät radiosignaalia, jota vastaanottimet voivat ottaa vastaan riippumatta säätilasta, kellonajasta ja paikasta, mikäli niillä on vapaa näkymä taivaalle. Yleensä noin kolmen satelliitin yhtäaikainen näkyminen riittää luotettavaan paikannukseen ja neljännen avulla voidaan myös korkeusasema määrittää luotetavasti. Paikannuksen tarkkuus riippuu useista seikoista, kuten näkyvissä olevien satelliittien määrästä ja ajasta joka paikannukseen on käytettävissä. Merkittävä parannus paikannuksen tarkkuuteen saadaan, mikäli käytettävissä on referenssitieto tunnetulta tukiasemalta (ns. differentiaali-GPS eli DGPS). Referenssisignaali voidaan yhdistää satelliittisignaaleihin joko reaaliaikaisesti (langattomasti) tai jälkikäsitellyssä. DGPS:n tarkkuus voi olla alle senttimetrin luokkaa.

GPS-paikannustiedosta voidaan helposti laskea esim. kulkuneuvon nopeus ja suunta. GPS:ään ja paikkatietoon perustuvat paikannus- ja navigointilaitteet ovat muodostunut suosituiksi mm. autoissa sekä matkapuhelimeissa. Laitteita on tarjolla kulutustavaroina sekä ammattikäyttöön.



#### Inertiaalipaikannus

Inertiaalipaikannus (INS Inertial Navigation System) perustuu gyroskooppeihin ja kiihtyvyyssantureihin ja sillä voidaan seurata kulkuneuvon (auto, helikopteri, lentokone) liikettä kolmiulotteisessa koordinaatistossa.

Yhdistämällä GPS ja INS saadaan aikaan tarkka kolmiulotteinen paikannus, jota voidaan hyödyntää esimerkiksi kaukokartoituksessa.

### 2.7.2. Kaukokartoitus

Kaukokartoitus on tiedonhankintamenetelmä, jossa kohteita mitataan ilman fyysistä kontaktia niihin. Kaukokartoitusinstrumentteja ovat mm. kamera, tutka ja laserkeilain eli lidar (light detection and ranging). Kameran käyttö kaukokartoitukseen kuumailmapalloista ja lentokoneista muodostui rutiiniksi jo toisen maailmansodan aikana. Ilmakuvat antavat hyvin tarkkaa laadullista (maapeite, kasvillisuus) ja määrällistä (etäisyydet, kohteiden koko) informaatiota kohteista. Stereokuvien (kaksi kuvaa, jotka on otettu samasta alueesta tai kohteesta vierekkäisillä kameroilla) avulla voidaan kohteesta saada kolmiulotteista tietoa esim. maastomallia varten. Nykyisin kaukokartoituskamerat tuottavat suoraan digitaalisia kuvia ja kuvaukseen käytetyn lentokoneen tms. paikannustieto yms. voidaan ottaa huomioon kuvien käsittelyssä. Ilmakuvat *oikaistaan* kuvapisteen ja vastaavien maastopisteiden (x, y ja z -koordinaatit) niin, että lopputulos vastaa karttaprojektion tuottamaa kuvaa.

Satelliittikaukokartoitus siviilikäyttöön alkoi 1960-luvulla sääsatelliiteista. 1970-luvulla otettiin käyttöön useita maapeitettä kuvaavia satelliitteja. Joidenkin satelliittien tarkoituksena on ollut kuvata laajoja alueita (esim. yksi tieto aina 1 km x 1 km alueelta) usein (esim. kerran päivässä) kun taas joidenkin tarkoituksena on ollut kuvata tarkemmin (esim. 20 m x 20 m rasterisoluissa, nykyisin jopa 1 m x 1 m) mutta harvemmin (esim. kerran kuussa). Satelliittien instrumentit kartoittavat maata yleensä pyyhkäisemällä näkymää lentorataansa nähden kohtisuoraan. Pyyhkäistyn alueen leveys voi vaihdella ja se voi olla esim. 10 km. Nämä instrumentit ovat yleensä monikanavaisia, tallentaen kohteista eri taajuuksilla (näkyvä valo, infrapuna jne) tulevaa säteilyä. Miehittämättömien satelliittien automaattisesti ottamat kuvat siirretään radion avulla maa-asemille, joissa ne koostetaan rasteripaikkatietoaineistoiksi. Satelliiteilla on oma toiminta-aikansa ja uusia satelliitteja lähetetään koko ajan käytöstä poistuvien tilalle.

Kaukokartoitusinstrumentti voi olla passiivinen tai aktiivinen. Passiivinen instrumentti mittaa maan tai kohteen pinnasta tulevaa säteilyä (esim. heijastunutta auringonvaloa). Pilvet häiritsevät suuresti passiivista satelliittikaukokartoitusta. Aktiiviset kaukokartoitusinstrumentit (tutkat) lähettävät kohteeseen säteilyä ja mittaavat takaisin heijastuneen säteilyn. Tutkatekniikalla mittauksen spatiaalinen resoluutio ei ole yleensä yhtä hyvä kuin passiivisessa kaukokartoituksessa, mutta toisaalta pilvet eivät ole niin iso ongelma eikä vuorokaudenajalla (yö/päivä) ole merkitystä. Laserkeilaimia (laseria hyödyntäviä aktiivisia kaukokartoituslaitteita) käytetään usein lentokoneista, helikopterista, autosta tai maanpinnalta. Laitte tuottaa hyvin tiheän joukon pisteitä, joiden sijainti (x,y,z) voidaan laskea ja lisäksi tiedetään kuinka mones kaiku piste on (säde läpäisee kasvuston osittain ja tuottaa kaikuja eri viiveillä).

Satelliittikaukokartoituksella on tuotettu ja tuotetaan merkittäviä globaaleja tai lähes globaaleja aineistoja maanpeitteestä, lumen esiintymisestä, lehtialasta, maan pinnanmuodoista, painovoiman jakaantumisesta jne.

Monikanavaista kaukokartoitustietoa käytetään hyödyksi tunnistettaessa kohteita. Tunnistus perustuu siihen, että kullakin kohdetyypillä on sille ominainen tapa heijastaa eritaajuisia säteilyä.

### 2.7.3. Maastomittaus

Maastossa voidaan mitata kohteiden välisiä etäisyyksiä, suuntia ja kulmia (geomatiikka) ja kohteiden ominaisuuksia (ympäristömittaus).

Perinteinen maanmittaustekniikka perustuu maaston kohtien ja kiintopisteiden tarkkaan paikantamiseen. Nykyisin tällainen paikantaminen tehdään pääasiassa yhdellä laitteella (takymetrillä, engl. total station), jolla mitataan pisteiden sijainteja kolmiulotteisessa koordinaatistossa. Takymetrit kykenevät käsittelemään ja laskemaan mittaustiedon sisältämällään mikroprosessorilla ja ohjelmistolla. Maanmittauksessa tarkoituksena on alueiden (kiinteistöjen) sijainnin paikantaminen maastossa sekä esim. rakennettavien kohteiden sijainnin osoittaminen. Maastomittaus on ilmakuvausten, GPS:n ja muiden tekniikoiden ohella edelleen tärkeä ja tarkin maanmittaustekniikka.

Maastomittauksella saadaan kohteista parhaiten ja tarkinta laadullista tietoa ja tietoa laadullisten tekijöiden muuttumisesta. Tiedon keräämiseksi on kannettavaan laitteisiin (PDA eli personal digital assistant -laitteet, maastokelpoiset kannettavat tietokoneet, älypuhelimet) kehitelty paikka- yms. tiedon käsittelyyn kykeneviä ohjelmistoja. Yhdessä GPS:n ja mobiiliviestinnän kanssa tietoa voidaan kerätä ja siirtää tietokantoihin tehokkaasti.

Ympäristömittaus kattaa mm. automaattiset reaaliaikaiset vedenlaatuanturit, jotka voivat välittää mittaustietonsa mobiiliviestinnän avulla keskustietokantoihin, sensoriverkkoperiaatteella toimivat monianturijärjestelmät ja doppler-ilmiötä hyödyntävät manuaaliset 2- ja 3-ulotteiset virtausmittauslaitteet. (Langaton) Sensoriverkko on joukko autonomisia antureita, jotka mittaavat jonkin suureen alueellista jakautumista tietyssä kohteessa. Autonomisuus tarkoittaa, sitä, että anturit kykenevät lähettämään ja välittämään muiden antureiden tuottamaa dataa, niin, että käyttäjän kannalta anturiverkko toimii kuin yksi iso anturi.



### Telematiikka

Telematiikka tarkoittaa viestintäteknologian (joka voi olla langallista tai langatonta) ja tietotekniikan yhdenmukaista käyttöä. Telematiikka (telematics) ja ICT (Information and Communications Technology) tarkoittavat siten käytännössä lähes samaa asiaa mutta telematiikkaan luetaan yleensä paikannustekniikan (esim. GPS) hyväksikäyttö. Liikennetelematiikka lienee melko vakiintunut termi kuvaamaan telematiikan hyödyntämistä liikennejärjestelmissä. Liikennejärjestelmä koostuu toisaalta kaduista, teistä, tunneleista, silloista, losseista jne. ja toisaalta tuota järjestelmää käyttävistä ajoneuvoista ja muista. Liikennetelematiikkaan kuuluu siten telemaattiset järjestelmät näissä kahdessa osassa ja näiden järjestelmien välinen kommunikaatio ja yhteistoiminta.

#### 2.7.4. Muita menetelmiä ja tekniikoita

Ympäristötietoa ja paikkatietoa voidaan kerätä ja koota muillakin tavoin kuin GPS:llä, kaukokartoituksella ja maastomittauksella. Monilla ympäristötekniikan (sen siviili-insinööritekniikan osa-alueella) alueilla on omia tekniikoita: liikennettä mitataan ja seurataan mm. teihin upotetuilla antureilla, videokameroilla, tutkilla, akustisilla tai infrapunasensoreilla. Näillä voidaan mitata liikennemäärien ja liikenteen ruuhkautumisen lisäksi mm. liikenteen tyyppiä (henkilöautot vs kuorma-autot).

Vapaaehtoinen (paikka)tieto on tietoa, jota ihmiset antavat vapaaehtoisesti johonkin tietojärjestelmään. Vapaaehtoinen paikkatieto on siten yksi muoto ns. joukkojen hyödyntämistä (crowdsourcing). Vapaaehtoinen paikkatieto voi olla kohteiden sijainteja, kohteiden ominaisuuksia tai esim. valokuvia.

Digitointi skannaamalla (esim. kuvia) taso- tai rumpuskannereilla on jo hieman poistuva menetelmä (koska tieto nykyisin on jo syntyessään yleensä digitaalista) siirtää paperilla tai kuvina olevaa aineistoa digitaalseksi.

---

# Aineistoharjoitus

Tavoitteet:

- Tutustua suomalaisiin aineistotuottajiin ja niiden tarjoamiin aineistoihin
- Osata ladata aineisto omalle koneelle ja purkaa pakattu tiedosto
- Osata tunnistaa aineisto
  - Tunnistaa paikkatietovektoriaineisto
  - Tunnistaa paikkatietorasteriaineisto
- Tutustua valittuun taulukkolaskentaohjelmaan (Excel)
  - Datat ja kaavojen syöttö
  - Kuvaajan teko
  - Haetun aineiston tuominen taulukkoon
  - Tunnuksien laskeminen aineistosta
- Tutustua valittuun paikkatieto-ohjelmaan (Quantum GIS)
  - Rasteriaineiston avaaminen ohjelmaan: ominaisuuksien etsiminen: karttaprojektio, georeferointi, solun koko, väritys, no data, datatyyppi, kaista
  - Vektoriaineiston avaaminen: karttaprojektio, kohteiden geometria, attribuuttitieto
  - Aineistopalvelun määrittäminen ja aineiston avaaminen sellaisesta

## 3.1. Aineistotuottajia

PaITuli

PaITuli on geoinformatiikan yliopistoverkoston FIUGINETin ja CSC:n yhdessä tuottama paikkatietojen välityskanava tutkijoiden ja opiskelijoiden käyttöön. PaITulin esittelysivu on osoitteessa <http://www.csc.fi/tutkimus/alat/geotieteet/paikkatieto/paituli>. Avaa k.o. sivulta aineistolista. Aineistojen koordinaattijärjestelmä on joko YKJ, KKJ tai EUREF-FIN. Näistä Suomessa käytössä olevista koordinaattijärjestelmistä on tietoa Julkisen hallinnon tietohallinnon neuvottelukunnan julkaisemasta dokumentista JHS 154, joka löytyy mm. osoitteesta <http://docs.jhs-suositukset.fi/jhs-suositukset/JHS154/JHS154.pdf>. Aineistojen formaatteja ovat ECW, ESRI Coverage, ESRI Grid, TIFF, Shape, DAT, LST, Mif, e00 jne. Aineistoformaateista löytyy tietoa mm. paikkatieto-ohjelmistojen sivuilta, esim. <http://www.envitia.com/products/productdetails.aspx?id=11,40,0>, [http://www.gdal.org/formats\\_list.html](http://www.gdal.org/formats_list.html) ja [http://www.gdal.org/ogr/ogr\\_formats.html](http://www.gdal.org/ogr/ogr_formats.html).

Avaa PaITulipalvelu. Sinun täytyy tunnistautua Aalto-salasanallasi koska palvelu vaatii Shibboleth-tunnistautumisen. Lue palvelun käyttöehdot. Avaa latauspalvelu. Hae peruskarttalehti, jolla näkyy Jyväskylän keskusta.

PaITulissa on myös Liikenneviraston tuottama Digiroad-aineisto. Ei kuitenkaan ladata siitä osaa PaITulista vaan käytetään sen sijaan tätä kurssia varten tehtyä WFS aineistopalvelua, jonka osoite on <http://geoinformatics.aalto.fi/edu/wfs.pl> (käyttöohjeet alla).

### OIVA

OIVA on ympäristöhallinnon tietopalvelu. OIVA on osoitteessa <http://www.wp2.ymparisto.fi/sc-ripts/oiva.asp>. Rekisteröidy OIVAn käyttäjäksi. OIVAn paikkatietoaineistot ovat myös PaITulissa, joten ei katsota niitä. Avaa ympäristönsuojelun tietojärjestelmä Vahti. Hae Jyväskylän Nenäinniemen puhdistamon päästötiedot tekstitiedostona.

Avaa ympäristötiedon hallintajärjestelmä Hertta. Siirry hydrologisten havaintotietojen hakuun valitsemalla Vesivarat → Hydrologiset havainnot → Tietojen haku. Haetaan Päijänteen pohjoisen asteikon vedenkorkeushavainnot vuodelta 2010. K.o. tiedot löytyvät esim. valitsemalla hakuehdoksi Alueen kautta kunnat, joiden listasta valitaan Jyväskylä. Listaamalla Jyväskylän havaintopaikat löydetään haluttu paikka. Laita k.o. paikan viereiseen ruutuun rasti ja valitse ylhäältä "Tulokset taulukkona". Uusi selainikkuna avautuu jossa sinun pitää muokata haluttua ajanjaksoa. Paina Tiedostoon-nappia ja vielä uudelleen avautuvassa uudessa ikkunassa. Klikkaa sen jälkeen ikkunaan tulevaa linkkiä ja tallenna tiedosto.

### Paikkatietoikkuna

Paikkatietoikkuna <http://www.paikkatietoikkuna.fi> on Maanmittauslaitoksen kehittämä paikkatietosivusto ja -portaali. Hae portaalin kautta Maanmittauslaitoksen avoimesta aineistosta "10 m Korkeusmalli" tiili, joka kattaa Nenäinniemen.

### HELCOM

HELCOM on Itämeren suojelukomissio, joka ylläpitää tietokantaa Itämereen kohdistuvista ympäristöpaineista. HELCOMin sivut ovat osoitteessa <http://www.helcom.fi/>. Tarkastellaan HELCOMin aineistoja aineistopalvelun kautta, jonka osoite on <http://maps.helcom.fi/ArcGIS/services/DataDelivery/MapServer/WMSServer>.

### Euroopan ympäristövirasto

Euroopan ympäristövirastolla (EEA) on paljon euroopanlaajuisia tietokantoja. EEA:n sivut ovat osoitteessa <http://www.eea.europa.eu/>.

### Globaalit aineistot

Globaaleja aineistoja löytyy esimerkiksi seuraavien sivujen kautta: <http://www.landcover.org/index.shtml>, <http://www.igrac.net/>.

## 3.2. Aineistojen tarkastelu

Alla kuvataan aineistojen tarkastelua. Pura ja tallenna aineistot esimerkiksi omalle verkkolevyillesi tai esim. siirrettävälle USB-tikulle.

### Peruskarttalehti PaITulista

Pura PaITulista saamasi zip-pakkaus. Hakemistoon mml/peruskartta/2010/RK1\_1/N4 tulee neljä tiedostoa, joiden päätteet ovat tab, tfw, tif ja xml. TIF-tiedoston olemassaolo kertoo, että kyse on TIFF-formaatissa olevasta rasteriaineistosta. TIFF sinänsä ei ole geospaatialinen formaatti vaan yleinen kuvaformaatti. TIFF-formaatti tukee metatietojen liittämistä kuvaan. GeoTIFF-formaatti hyödyntää tätä, mutta tässä tapauksessa geospaatialinen metatieto on ulkoisissa tiedostoissa. TFW-tiedosto on ns. ESRI world file ja se on tekstitiedosto, jossa on allekkain kuusi numeroa: solun leveys x-akselin suuntaan, kierto y-akselin suhteen, kierto x-akselin suhteen, solun korkeus y-akselin suuntaan negatiivisena, ylhäällä vasemmalla olevan solun keskipisteen x-koordinaatti, ylhäällä vasemmalla olevan solun keskipisteen y-koordinaatti. TAB-tiedosto on MapInfo-ohjelmaspesifinen tekstitiedosto. XML-tiedostosta voi lukea, että siinä on käytetty RASTA-nimistä XML-kielitä. Nimiavaruuden määrittely viittaa maanmittauslaitokseen (nls.fi). XML-tiedostosta voi lukea mm. että käytetty koordinaattijärjestelmä on ETRS-TM35FIN.

Peruskarttalehden ja digiroad aineiston tarkastelu yhdessä Quantum GIS -ohjelmassa

Avaa Quantum GIS -ohjelma. Avaa peruskarttalehti ohjelmaan esim. Layer | Add raster layer -konnella: navigoi dialogilla data-hakemistoosi tif-tiedostoon ja avaa se. Huomaa, että x-koordinaattien arvot ovat yli 400 000 ja y-koordinaattien arvot ovat yli 6 000 000. Hae kartalta Nenäinniemi.



### Huomaa

Quantum GIS -ohjelmiston käyttöoppaita eri versioille ja kielillä on ladattavana sivulla <http://www.qgis.org/en/documentation/manuals.html>.

Haetaan pala Digiroad-aineistoa palvelusta. Avaa Quantum GISsin WFS-taso -dialogiboksi ja paina "New". Määrittele avautuvaan dialogiin nimi yhteydelle ja yhteyden URL (yllä). WFS-taso -dialogiboksissa valitse uusi yhteys ja paina "Connect". Luetteloon tulee muutamia tasoja, joista digiroad-aineistoa on liikenne-elementti(geom). Kun valitset kyseisen tason, huomaat että se on tarjolla oletusarvoisesti EPSG:4258 koordinaattijärjestelmässä (ETRS89). Koordinaatistojärjestelmä voidaan vaihtaa "Change..." nappulan avulla mutta tässä tapauksessa se ei onnistu, koska kyseinen aineisto ei ole tarjolla muissa koordinaattijärjestelmissä tässä palvelussa.



### Huomaa

Koska peruskarttalehti on järjestelmässä ETRS-TM35FIN, kannattaisi Digiroad-aineisto hakea samassa järjestelmässä mutta nyt se ei ole mahdollista. Kun haet vektoriaineistoja WFS-palvelusta kannattaa taustakartta ensin kohdistaa haluttuun kohtaan ja hakea aineisto sitten laittaen rasti "Only request features..." kohtaan. Palvelussa voi nimittäin olla iso aineisto. Jos se olisi koko Suomen tiestö (tässä ei ole), niin ohjelman pyytämän aineiston koko olisi satoja megatavuja. Lisäksi palvelimen tekemä muunnos tietokannasta GML-muotoon ja mahdollinen koordinaattijärjestelmän muutos vaativat prosessointia.

Quantum GIS kykenee tekemään muunnoksia lennossa, joten nyt on käytettävä sitä mahdollisuutta jos ja kun tieaineistoa halutaan katsoa peruskarttalehden päällä. Aseta järjestelmä ETRS-TM35FIN karttanäytölle oletusarvoksi. Valitse File | Project properties... ja valitse ETRS-TM35FIN. Laita myös rasti kohtaan "Enable on the fly CRS transformation".

Lisää vastaavasti Maanmittauslaitoksen palvelusta saamasi korkeusmalli paikkatieto-ohjelmaan. Pura saamasi zip-paketti ja avaa xyz-tiedosto. MML:n tässä käyttämä xyz-formaatti on hyvin yksinkertainen tekstitiedosto, jossa ei ole mukana tietoa koordinaattijärjestelmästä, joten QGIS kysyy sitä. Kyseinen tieto on MML:n latauspalvelussa.

Lisätään karttaan vielä uusi aineistokerros jäteveden puhdistamoille. Valitse Layer | New | New shapefile layer. Valitse geometriatyyppiä piste, CRS:ksi YKJ ja lisää attribuutti Nimi tyyppiä teksti. Paina ok ja valitse sitten aineiston menusta editointi päälle (Toggle editing). Lisää kerrokseen piste painamalla Capture point nappula alas ja klikkaamalla Nenäinniemen puhdistamon kohtaa kartassa. Anna kohteelle nimi Nenäinniemi. Tallenna tekemäsi aineisto ja laita editointi pois päältä. Muokkaa sitten aineiston ominaisuudet dialogin kautta aineiston esitystapaa niin, että pisteen lisäksi kartassa näkyy kohteen nimi. (alue Labels, Display labels, Field containing label kohtaan nimi).

## Luku 3. Aineistoharjoitus

---

### Itämeren fosforikuormituskartta HELCOMilta

Hae yllämainitusta HELCOMin WMS-palvelusta Itämeren taustakartta ja jokien fosforikuormitustiedot. Hae kartan avulla Siikajoen fosforikuormitustiedot.

### Puhdistamon päästötiedot OIVasta

Siirrä lataamasi tekstitiedosto data-hakemistoon, jos se ei ole jo siellä, ja uudelleennimeä se kuvaavalla nimellä. Avaa taulukkolaskentaohjelma ja tuo tiedot siihen taulukoksi (Insert | Sheet from file). Tiedostossa on käytetty tabulaattoria kenttien erottimena. Saatat joutua muuttamaan pilkut pisteiksi esim. taulukkolaskentaohjelman find & replace -komennolla. Datatiedostossa on Puhdistusprosentti-riveillä liian vähän tabulaattoreita mutta sillä ei ole lopputuloksen kannalta merkitystä.

Tehdään aineistosta kuvaaja vesistöön päätyneen kokonaisfosforin määrästä vuosina 1981-2010. Aseta taululle Autofiltteri päälle (Data | Filter | AutoFilter). Valitse sitten sarakkeen Tyyppi kohdalta niin, että ainoastaan rivit joissa on "Vesistöön" näkyvät. Valitse sitten sarakkeet Vuosi ja Kokonaisfosfori ja valitse komento Insert | Chart. Valitse kuvaajan tyyppi XY-data.

### Vedenkorkeusaikasarja OIVasta

Siirrä lataamasi aikasarjatiedosto data-hakemistoon, jos se ei ole jo siellä, ja uudelleennimeä se kuvaavalla nimellä. Tuo vedenkorkeusaikasarja taulukkolaskentaohjelmaan. Määrittele tuodessaasi, että vasemmanpuoleinen aikasarja on muotoa päivä.kuukausi.vuosi. Tee aikasarjasta kuvaaja.

Luo uusi aikasarja ja anna sille arvot (Pvm, W) = (1.1.2010, 150; 15.3.2010, 140; 15.6.2010, 220; 31.10.2010, 160; 31.12.2010, 160). Lisää tämä aikasarja vedenkorkeushavaintoaikasarjakuvaajaan.

## 3.3. Raportointi

Tee ja palauta harjoituksesta raportti, jossa on karttakuva Nenäinniemenestä sekä aikasarjakuvaajat Nenäinniemen puhdistamon kokonaisfosforikuormituksesta ja Päijänteen vedenkorkeudesta vuonna 2010 sekä HELCOMin fosforikuormituskartta. Karttakuvassa tulee olla peruskartan lisäksi digiroad-aineisto ja Nenäinniemen puhdistamo. Päijänteen vedenkorkeusaikasarjan kuvaajassa tulee olla mukana "tavoitevedenkorkeuden" kuvaaja. Raportoi myös Siikajoen fosforikuormitustiedot.



---

# Ohjelmointiharjoitus

Tässä harjoituksessa tutustutaan teknis-tieteelliseen laskentaan MATLAB-ohjelmointikielellä. Harjoituksessa tehdään kaksi regressioanalyysia, joiden tuloksia käytetään seuraavassa harjoituksessa altaan simuloinnissa.

## 4.1. Varastoyhtälö

Altaan toimintaa simuloidaan varastoyhtälön

$$S_{t+1} = S_t + I_t - O_t \quad (4.1)$$

avulla. Varastoyhtälö kertoo, että altaassa oleva vesimäärä aika-askelen  $t$  lopussa saadaan lisäämällä aika-askelen alussa olevaan vesimäärään aika-askelen aikana tuleva vesimäärä ja vähentämällä lähtevä vesimäärä. Altaassa olevaa vesimäärää mitataan epäsuorasti altaan vedenkorkeuden kautta ja altaasta lähtevä virtaama on säännöstelemättömässä altaassa vedenkorkeuden funktio. Tiettyihin vedenkorkeuksiin liittyvät menovirtaamat ja vesipinta-alat on esitetty allaolevassa taulukossa.

Taulu 4.1. Altaan tiedot

Vedenkorkeus (W) [m]	Menovirtaama (O) [m <sup>3</sup> /s]	Vesipinta-ala (A) [km <sup>2</sup> ]
0,50	0,002	10,0
1,00	0,096	11,0
1,20	0,302	12,0
1,45	0,966	13,0
1,60	1,594	14,0
1,72	2,501	15,0
1,80	3,200	16,0
1,92	4,740	17,0
2,00	6,255	18,0

Vedenkorkeuksien  $W_i$  ja  $W_{i+1}$  välisen varaston tilavuus voidaan laskea katkaistun kartion kaavalla  $S = h(A_i + A_{i+1} + \sqrt{A_i A_{i+1}})/3$ . Osavarastojen kumulatiivisen summan avulla saadaan kutakin vedenkorkeutta vastaava varaston tilavuus valitsemalla varastotilavuudelle mielivaltainen nollakohta.

## 4.2. Regressioanalyysi

Allaolevassa kuvassa on esitetty regressioanalyysin peruskaavoja.

$$S_{xx} = \sum_{i=1}^n (x_i - \bar{x})^2 = \sum_{i=1}^n x_i^2 - \frac{(\sum_{i=1}^n x_i)^2}{n}$$

$$S_{yy} = \sum_{i=1}^n (y_i - \bar{y})^2 = \sum_{i=1}^n y_i^2 - \frac{(\sum_{i=1}^n y_i)^2}{n}$$

$$S_{xy} = \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) = \sum_{i=1}^n x_i y_i - \frac{(\sum_{i=1}^n x_i)(\sum_{i=1}^n y_i)}{n}$$

$$b = \frac{S_{xy}}{S_{xx}}$$

$$a = \bar{y} - b\bar{x}$$

$$SSR = bS_{xy}$$

$$SST = S_{yy}$$

$$R^2 = \frac{SSR}{SST}$$

Kuva 4.1. Regressioanalyysin kaavat.  $S_{xx}$ ,  $S_{yy}$  ja  $S_{xy}$  ovat apumuuttujia. SSR on regressiomallin selittämä osa kokonaisneliösummasta SST.  $R^2$  on mallin selitysaste.

Sovitetaan purkautumiskäyrään yhtälö  $O = a_p + b_p W^6$  ja varastokäyrään yhtälö  $W = a_v + b_v S$  oleva kaava.

### 4.3. Laskenta MATLAB:illa

Tehdään regressioanalyysi MATLAB-ohjelmalla (tai Octavella) vuorovaikutteisesti. Lähdetään liikkeelle määrittelemällä kolme vektoria W, O ja A ja tallentamalla niihin mittausdata:

```
W = [0.5 1 1.2 1.45 1.6 1.72 1.8 1.92 2]
O = [0.002 0.096 0.302 0.966 1.594 2.501 3.2 4.74 6.255]
A = [10 11 12 13 14 15 16 17 18]
```

Luodaan sitten x ja y vektorit purkautumiskäyrän sovitus varten:

```
x = W.^6
y = O
```

.^-operaattori tarkoittaa vektorin alkioittain potenssiin korottamista. Vastaavasti esimerkiksi .\* tarkoittaa elementteittäin tapahtuvaa kertomista. Vektorin alkioiden summa lasketaan sum-funktiolla ja alkioiden määrä numel-funktiolla.

Vedenkorkeuksia vastaavan varastotilavuus-vektorin laskeminen vaatii for-silmukan käyttöä:

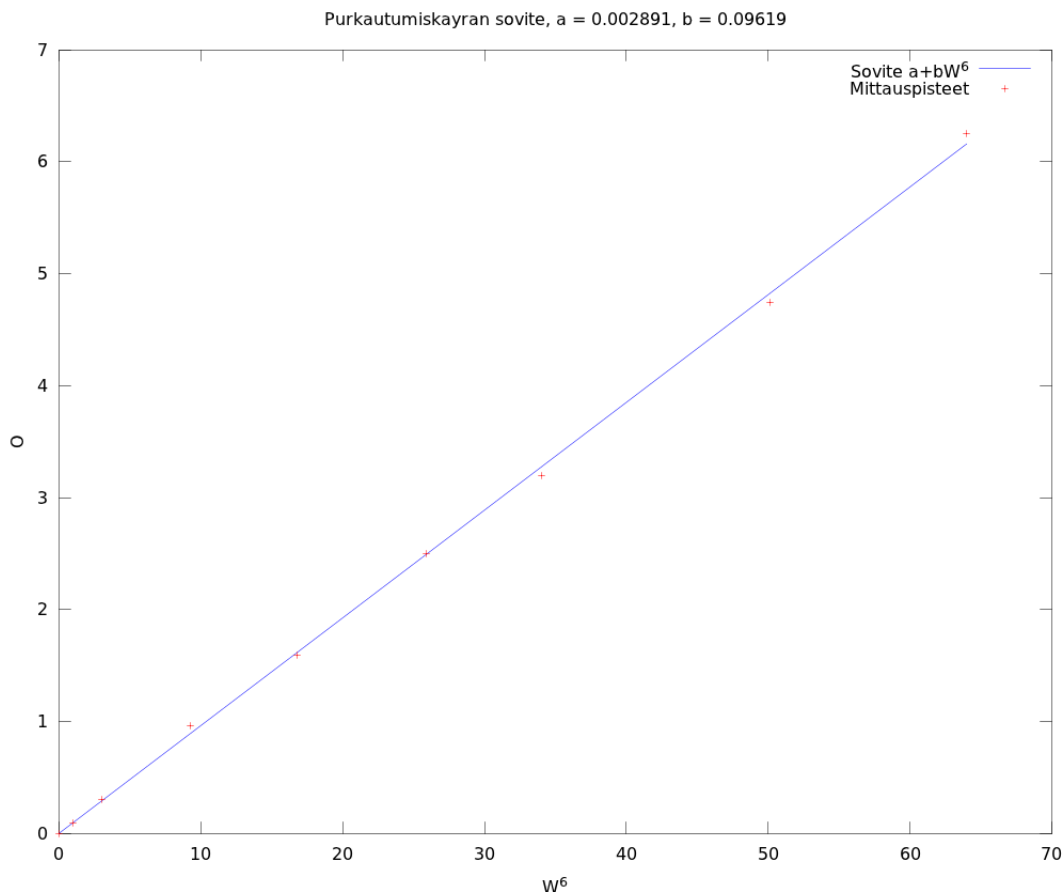
```
S = 0
for i=2:numel(W),
```

```
S(i)=(W(i)-W(i-1))*(A(i-1)+A(i)+sqrt(A(i-1)*A(i)))/3+S(i-1), end
```

Laske tämän jälkeen purkautumiskäyrän ja varastokäyrän parametrit ( $a_p$ ,  $b_p$ ,  $a_v$ ,  $b_v$ ) ja regressiomallien selitysasteet toteuttamalla ylläolevat kaavat MATLABissa.

## 4.4. Raportointi

Raportoi tehtävä esittämällä koodi, jolla olet tehnyt analyysit, analyysin tulokset (käyrien parametrit ja sovituksien hyvydet) ja sovite kuvana. Alla on esimerkki kuvasta ja koodista, jolla se on luotu (kuva on tehty Octavella, jossa skandinaaviset merkit eivät toimineet).



```
plot(x, a+b*x)
hold on
plot(x, y, '+','color','red')
ylabel('O')
xlabel('W^6')
legend('Sovite a+bW^6','Mittauspisteet')
title('Purkautumiskayran sovite, a = 0.002891, b = 0.09619')
print -dpng foo.png
```

---

---

# Taulukkolaskentaharjoitus 1: simulointi

Tässä harjoituksessa tutustutaan taulukkolaskentaohjelman mahdollisuuksiin. Harjoituksessa tehdään altaan toimintaa simuloiva laskentamalli.

## 5.1. Varastoyhtälö

Altaan toimintaa simuloidaan varastoyhtälön

$$S_{t+1} = S_t + I_t - O_t \quad (5.1)$$

avulla. Varastoyhtälö kertoo, että altaassa oleva vesimäärä aika-askeleen  $t$  lopussa saadaan lisäämällä aika-askeleen alussa olevaan vesimäärään aika-askeleen aikana tuleva vesimäärä ja vähentämällä lähtevä vesimäärä. Altaassa olevaa vesimäärää mitataan epäsuorasti altaan vedenkorkeuden kautta ja altaasta lähtevä virtaama on myös vedenkorkeuden funktio ns. purkautumiskäyrän kautta. Altaasta olevat vedenkorkeuteen liittyvät tiedot on esitetty taulukossa alla.

## 5.2. Altaan toiminnan simulointi

Oletetaan, että simuloinnin alussa allas on tyhjä, ja siihen aletaan johtaa vettä  $3 \text{ m}^3/\text{s}$ . Toteutetaan taulukkolaskentaohjelmaan simulointi, joka laskee altaan vedenkorkeuden ja siitä purkautuvan menovirtaaman ajan funktiona.

Kysymyksessä on ns. alkuarvotehtävä, joka ratkaistaan numeerisesti käyttäen valittua aika-askelta ja valittua numeerista menetelmää. Valitaan numeeriseksi menetelmäksi neljännen asteen Runge-Kutta, joka estimoii tilamuuttujan (tässä varaston koko) arvoa aika-askeleen lopussa yhtälöllä

$$S_{t+dt} = S_t + (k_1 + 2k_2 + 2k_3 + k_4) dt / 6 \quad (5.2)$$

jossa

$$k_1 = f(t, S_t) \quad (5.3)$$

$$k_2 = f(t+dt/2, S_t + k_1 dt/2) \quad (5.4)$$

$$k_3 = f(t+dt/2, S_t + k_2 dt/2) \quad (5.5)$$

$$k_4 = f(t+dt, S_t + k_3 dt) \quad (5.6)$$

$f(t, S)$  on tilanmuutosyhtälö, eli tässä tapauksessa varastoyhtälö on differenssimuodossa

$$f(t, S) = dS/dt = I(t) - O(t) \quad (5.7)$$

Tässä tapauksessa  $I(t)$  on vakio ja  $O(t)$  on purkautumiskäyrän mukainen menovirtaama (varaston funktio).

Tee simulointimallin sarakkeiden otsikkorivit. Ne ovat  $t, W, S, O, k_1, S_2, O_2, k_2, S_3, O_3, k_3, S_4, O_4, k_4$ . Simuloinnin parametrit ovat  $dt, I$  ja  $W_0$ . Laita  $dt$  ja  $I$  mallin yläpuolelle. Koska  $S$ :n yksikkö on  $\text{Mm}^3$ , niin  $dt$ :n yksikkö on  $\text{Ms}$ . Tehdään malliin aikasarake, niin, että sen yksikkö on päiviä. Sen ensimmäisellä rivillä on arvo 0 ja alemmilla edellisen rivin arvo  $+ dt * 1000000 / (60 * 60 * 24)$ . Vedenkorkeussarakkeen ensimmäisen rivin arvoksi tulee  $W_0 = 0,5 \text{ m}$  ja seuraavien rivien arvot lasketaan varaston arvosta sen oikealta puolelta. Varastosarakkeen ensimmäinen arvo lasketaan  $W_0$ :sta ja seuraavat arvot lasketaan Runge-Kutta -kaavalla. Jokainen  $O$ -sarake lasketaan sen viereisen  $S$ -sarakkeen sekä varasto- ja purkautumiskäyrien avulla. Jokainen  $k$ -sarake lasketaan  $I$ :n ja viereisen  $O$ -sarakkeen arvon avulla niiden erotuksena.  $S_2, S_3$  ja  $S_4$  sarakkeet lasketaan ylläolevien  $k_2$ :n,  $k_3$ :n ja  $k_4$ :n yhtälöiden oikeanpuoleisten lausekkeiden mukaisesti.

### 5.3. Simulointimallin raportointi

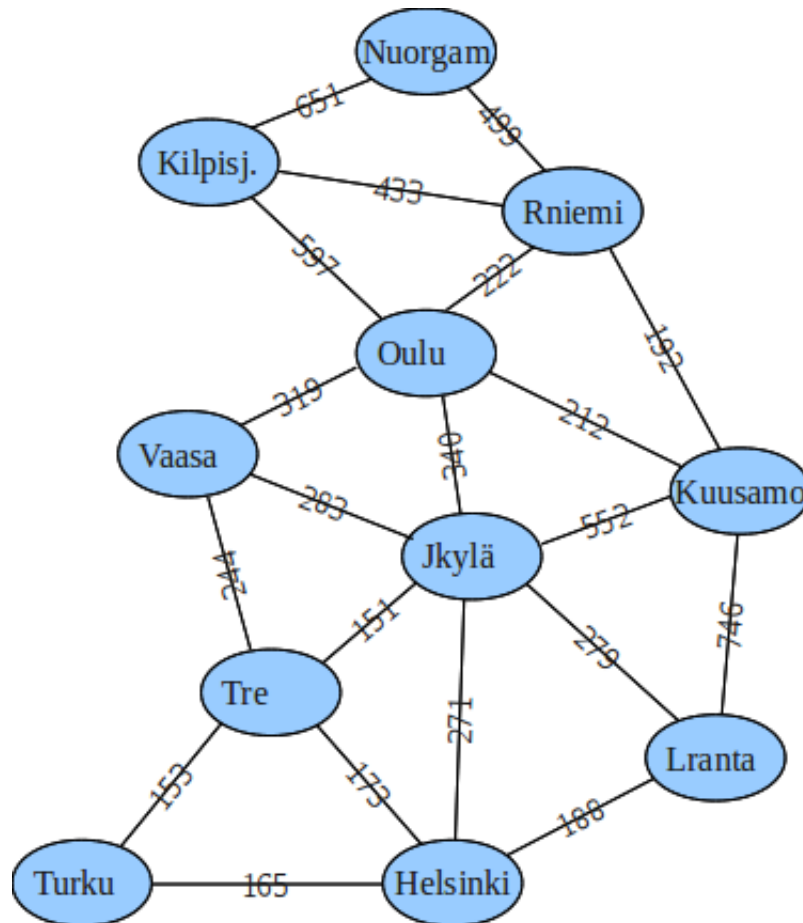
Raportoi simulointimalli yhdellä A4:llä esittämällä purkautumiskäyrän ja varastokäyrän yhtälöt, simuloinnin lähtötiedot sekä altaan vedenkorkeuden kehittyminen ajan funktiona.

# Taulukkolaskentaharjoitus 2: optimointi

Tässä harjoituksessa tutustutaan taulukkolaskentaohjelman mahdollisuuksiin. Harjoituksessa tehdään verkostoon liittyviä optimointeja.

## 6.1. Verkoston kuvaaminen taulukkolaskentaohjelmassa

Tee oheisen kuvan mukainen verkosto taulukkona, jonka sarakkeet ovat alkupää, loppupää ja välipää. Anna sarakkeiden soluille niiden otsikkoa vastaavat nimet.



Helsinki;Jyväskylä;271  
Helsinki;Lappeenranta;188  
Helsinki;Tampere;173  
Helsinki;Turku;165  
Jyväskylä;Helsinki;271  
Jyväskylä;Kuusamo;552  
Jyväskylä;Lappeenranta;279  
Jyväskylä;Oulu;340  
Jyväskylä;Tampere;151  
Jyväskylä;Vaasa;283  
Kilpisjärvi;Nuorgam;651  
Kilpisjärvi;Oulu;597  
Kilpisjärvi;Rovaniemi;433  
Kuusamo;Jyväskylä;552  
Kuusamo;Lappeenranta;746  
Kuusamo;Oulu;212  
Kuusamo;Rovaniemi;192  
Lappeenranta;Helsinki;188

Lappeenranta;Jyväskylä;279  
Lappeenranta;Kuusamo;746  
Nuorgam;Kilpisjärvi;651  
Nuorgam;Rovaniemi;499  
Oulu;Jyväskylä;340  
Oulu;Kilpisjärvi;597  
Oulu;Kuusamo;212  
Oulu;Rovaniemi;222  
Oulu;Vaasa;319  
Rovaniemi;Kilpisjärvi;433  
Rovaniemi;Kuusamo;192  
Rovaniemi;Nuorgam;499  
Rovaniemi;Oulu;222  
Tampere;Helsinki;173  
Tampere;Jyväskylä;151  
Tampere;Turku;153  
Tampere;Vaasa;244  
Turku;Helsinki;165  
Turku;Tampere;153  
Vaasa;Jyväskylä;283  
Vaasa;Oulu;319  
Vaasa;Tampere;244

### 6.2. Lyhin reitti verkostossa

Tehdään laskentamalli lyhyimmän reitin optimoivaa laskentaa varten.

Lisää välimatkataulukon sarake "reitissä". Anna k.o. sarakkeen soluille vastaava nimi. Tähän sarakkeeseen tulee arvo 0, jos yhteys ei kuulu reittiin ja 1 jos se kuuluu reittiin. Laita koko sarakkeeseen alustavasti arvoksi 0. Tee koko reitin pituuden laskeva solu ja kirjoita siihen kaava `SUMPRODUCT(<etäisyys-sarakkeen nimi>, <reitissä-sarakkeen nimi>)`.

Tee uusi taulukko johon tulee sarakkeet kaupunki, tullaan, lähdetään ja erotus. Taulukon tulee kullekin verkon kaupungille yksi rivi. Sarakkeeseen "tullaan" tulee kaava `SUMIF(loppu, CONCATENATE("=", <viittaus viereiseen soluun, jossa on kaupungin nimi>), reitissa)` ja sarakkeeseen "lähdetään" tulee muuten samanlainen kaava paitsi että viittauksen loppu sijasta on viittaus alku. Sarakkeeseen "erotus" tulee sarakkeen "tullaan" ja sarakkeen "lähdetään" erotus. Tutki nyt miten tämän taulun arvot muuttuvat, kun muutat välimatkataulukon sarakkeen "reitissä" arvoja 0:sta 1:een ja takaisin.

Käytetään Solver-työkalua lyhyimmän reitin Helsingistä Nuorgamiin etsimiseen. Valitse reitin pituuden laskeva solu ja avaa Solver. Valitse "minimoi". Kirjoita kohtaan "by changing cells" "reitissa". Lisää seuraavat ehdot: lähtöpaikan rivillä erotuksen pitää olla -1, ja matkakohteen rivillä sen tulee olla 1, kaikilla muilla riveillä erotuksen tulee olla nolla. Näiden ehtojen lisäksi lisää ehto, jonka mukaan jokaisen "reitissa" solun tulee olla 0 tai 1 (eli binary).

### 6.3. Verkoston välityskyky

Tehdään malli, jolla voidaan laskea edellisen tehtävän tieverkoston kyky välittää tavaraa kaupungista toiseen.

Lisää välimatkataulukon kaksi saraketta "kapasiteetti" ja "käytössä". Anna sarakkeiden soluille vastaavat nimet. Aseta kapasiteetin arvoksi 10000/etäisyys pyöristettynä kokonaisluvuksi ja käytössä-ominaisuuden arvoksi 0.

Tee taulukko kuljetuksille samaan tapaan kuin edellisessäkin tehtävässä mutta käytä "reitissa" viittauksen sijasta viittausta "kaytossa".

Muuta kaupungit-aulussa sarakkeet tullaan ja lähdetään laskemaan tulevaa ja lähtevää tavaramäärää.



Maksimoi nyt Helsingistä Rovaniemelle vietävää tavaramäärää. Rajoitusehtoja ovat: käytetyn kapasiteetin määrän pitää olla suurempi kuin nolla ja pienempi kuin k.o. yhteyden kapasiteetti. Kuhunkin kaupunkiin (paitsi Helsinkiin ja Rovaniemelle) tulevan ja lähtevän tavaramäärän pitää olla yhtä suuri (eli erotuksen pitää olla nolla). Rovaniemelle tulevan tavaramäärän pitää olla sama kuin Helsingistä lähtevä ja Helsinkiin tulevan tavaramäärän sekä Rovaniemeltä lähtevän tavaramäärän pitää olla nolla. (Huomautus: maksimivälitys voidaan saada aikaan useammalla kuin yhdellä tavalla koska mallissa ei ole taloudellisuusrajoitusta)

## 6.4. Raportointi

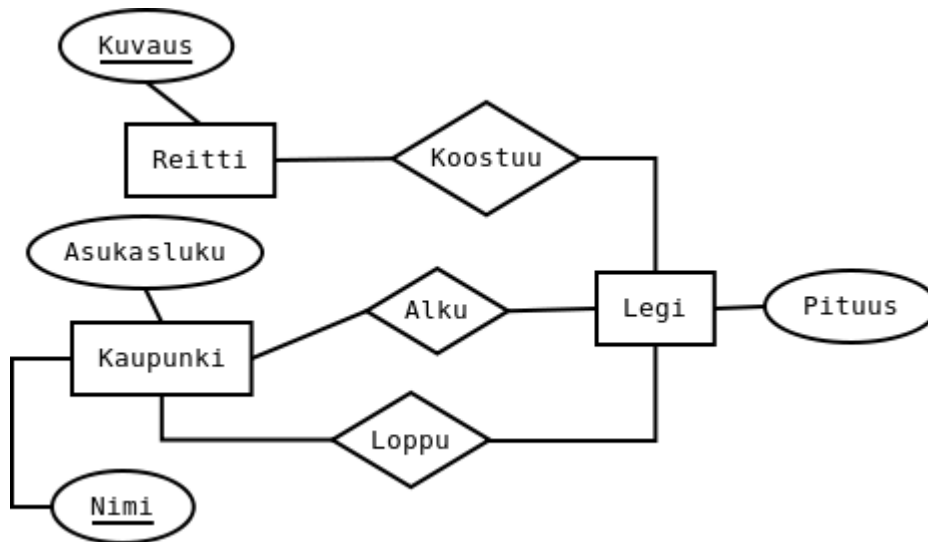
Raportoi verkostotehtävät esittämällä 1) lyhin reitti ja sen pituus ja 2) tapa, jolla Helsingistä voidaan kuljettaa maksimimäärä tavaraa Rovaniemelle. Käytä raportissa myös kuvia havainnollistamaan suunnittelemissi reittejä. Kiitettävän arvosanan voit saada tutkimalla tavarankuljetustehtävän ratkaisua taloudellisuuden kannalta.

---

# Tietokantaharjoitus

Tässä harjoituksessa tutustutaan tietokantatekniikkaan. Tehtävänä on toteuttaa yksinkertainen tietokanta suunnitelman pohjalta. Kun tietokanta on valmis, siihen lisätään tietoa ja siitä haetaan tietoa kyselyillä.

Tietokannan ER-malli on esitetty oheisessa kuvassa. Toteutettava tietokanta koostuu kolmesta entiteetti-luokasta: kaupungeista, legeistä (osamatkoista, matkoista kaupungista kaupunkiin) ja reiteistä. Kukin kaupunki määrittyy nimensä perusteella. Kukin legi koostuu kahdesta kaupungista (legin alku- ja loppupää). Legillä on pituus. Kahden kaupungin välillä voi periaatteessa mallin mukaan olla useita legejä, mutta tarkoitus on, että niitä on vain yksi. Reittejä saadaan yhdistämällä legejä toisiinsa. Kullakin reitillä on oma kuvauksensa.



## 7.1. Tietokantayhteys

Tehtävässä käytetään PostgreSQL tietokantaa, joka on palvelimessa water.hut.fi. Tietokantaan otetaan yhteys pgAdmin III ohjelmalla. Samoja ohjelmia käytetään myöhemmin paikkatietoharjoituksessa, jolloin keskeisessä osassa on PostgreSQL:n paikkatietolaajennus PostGIS. Molemmat ovat ilmaisia, avoimen lähdekoodin ohjelmia ja ne on helppo asentaa omaankin koneeseen, mutta water.hut.fi:n käyttö mahdollistaa valmiin ympäristön.

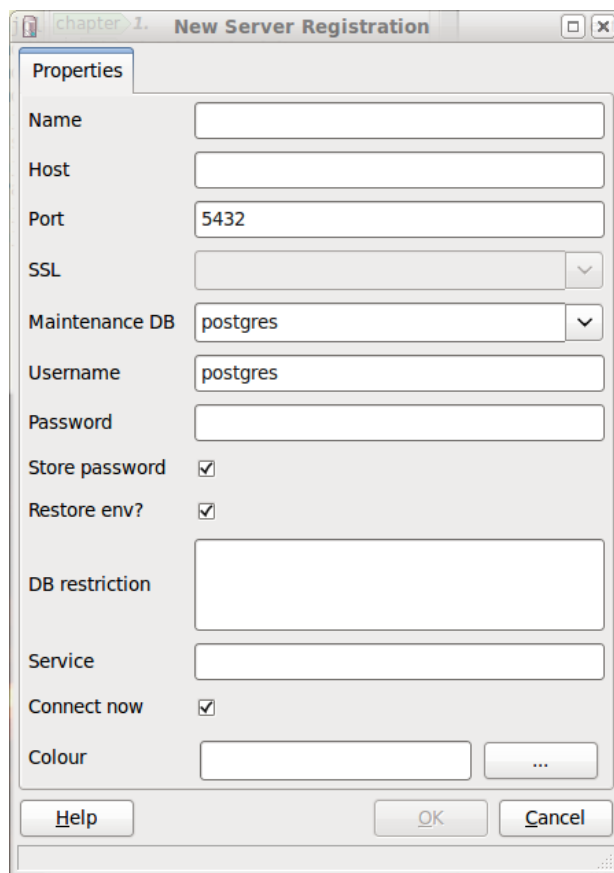
Käynnistä pgAdmin III -ohjelma ja valitse File|Add server. Dialogiboksissa (kuva alla) anna tiedot: Name: kuvaava nimi, esimerkiksi "water", Host: water.hut.fi, Username: käyttäjätunnuksesi, Password: käyttäjätunnuksesi. Käyttäjätunnuksesi on luotu nimestäsi ottamalla etunimestä ensimmäinen kirjain ja sukunimestä 7 kirjainta. Nimen mahdollisista skandinaavisista kirjaimista on poistettu pisteet ja mahdollinen väliviiva on poistettu kokonaan ensin. Kun yhteys toimii saat pgAdmin-ohjelman Object browser -paneeliin uuden ikonin, jonka valitsemalla ja/tai jonka vasemmalla puolella olevasta ruudusta painamalla saat näkyviin yhteyden komponenttipuun. Sinulle on luotu oma tietokanta, joka on komponentin Databases alla. Avaamalla oman tietokantasi komponentin Schemas ja sen alta komponentin public löydät tietokantasi taulualueen komponentin. Valitsemalla tietokantasi taulualueen komponentin ja valitsemalla Tools | Query tool saat ikkunan (kuva alla kolmantena), jonka kautta voit antaa harjoituksessa vaaditut komennot tietokannallesi.



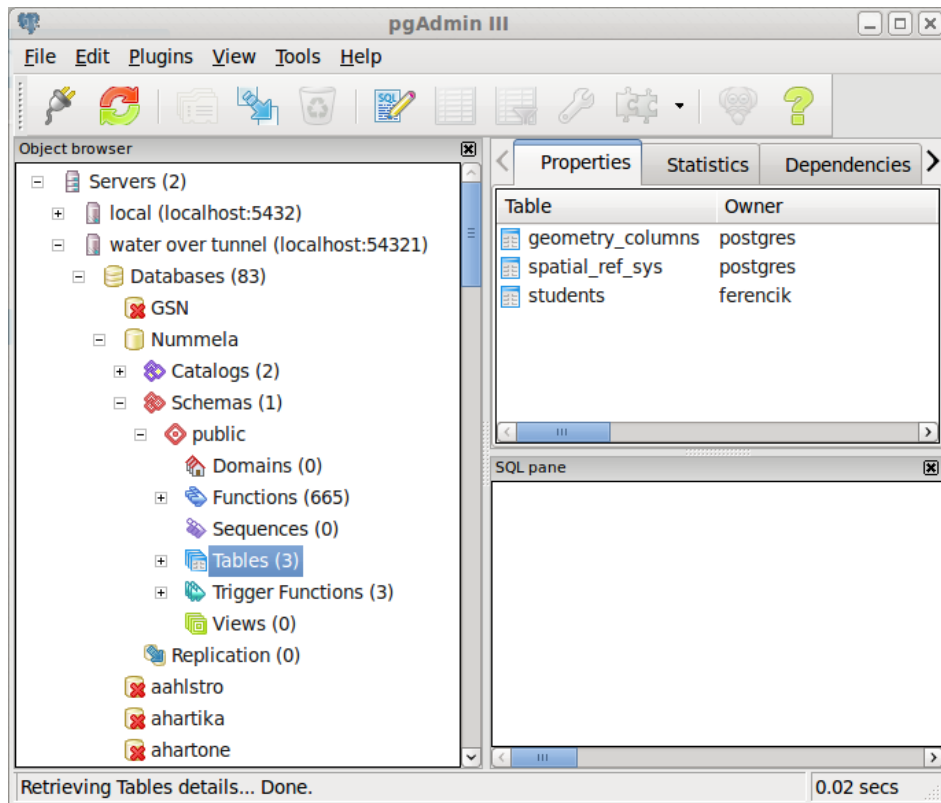
### Huomaa

Käytät parhaillaan hajautettua, asynkronista tietojärjestelmää. Tämä tarkoittaa mm. sitä, että pgAdmin III ei jatkuvasti kysele tai PostgreSQL-palvelin jatkuvasti kerro mikä tietokannan tila kulloinkin on. Tietokanta saattaa siis muuttua ilman että huomaat sitä heti. Esimerkiksi joku toinen, jolla on oikeus käsittelemääsi tietokantaan saattaa luoda sinne uuden taulun. Jopa itse tietokantakomentoikkunassa tietokantaan tekemäsi muutokset eivät automaattisesti päivity pääikkunaan. Tätä varten pgAdmin III ohjelmassa on virkistys-toiminto (Refresh). Kun valitset View | Refresh, niin valitun komponentin tila päivittyy.

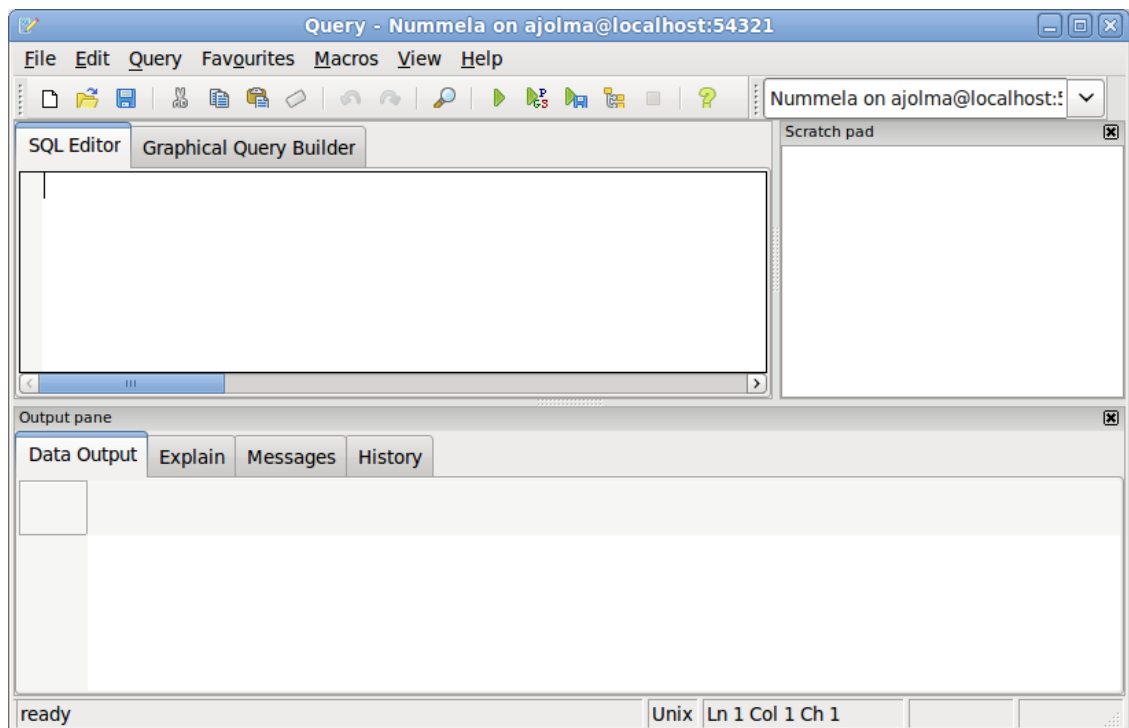
Tietokantakomentoikkunassa on kolme paneelia, joista kahdessa on valinnaisia sisältöjä. Oikealla ylempänä olevassa paneelissa olevan SQL Editorin kautta annetaan SQL-kieliset komennot, joilla luodaan ja manipuloidaan tietokantatauluja ja haetaan niistä tietoja. Komento ajetaan valitsemalla Query | Execute. Komennon tulokset tulevat oikealla alla olevaan paneeliin. Vasemmalla puolella oleva paneeli on ikäänkuin suttupaperi, jonne voi tallentaa väliaikaisesti komentoja, joita ei juuri sillä hetkellä tarvita.



Kuva 7.1. pgAdmin III -ohjelman uuden tietokantapalvelimen määrittämisdialogi



Kuva 7.2. pgAdmin III:n pääikkuna, jossa Object browser -paneeliin on avattu yhteys water.hut.fi:n tietokantapalvelimessa olevaan tietokantaan Nummela ja valittu sen taulualue.



Kuva 7.3. pgAdmin III:n tietokantakomentoikkuna

## 7.2. Tietokannan toteuttaminen

Luodaan tietokannan kolme entiteettitaulua ja asetetaan samalla niiden pääavaimet.

```
create table kaupungit (nimi text primary key,  
                        asluku integer);  
create table legit (id serial primary key,  
                  pituus integer,  
                  alku text,  
                  loppu text);  
create table reitit (id serial primary key,  
                   kuvaus text);
```

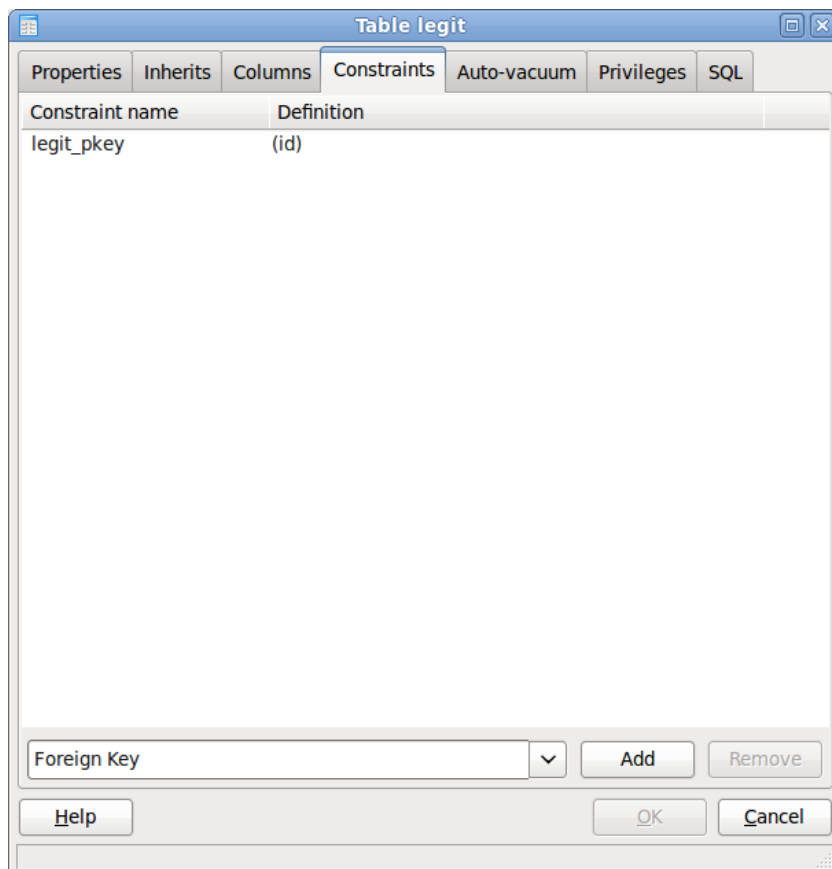


### SQL, isot ja pienet kirjaimet

SQL-komennot esitetään usein niin, että varatut sanat kuten insert ja select on kirjoitettu isolla. SQL-standardeissa tätä ei ole määritely pakolliseksi, mutta eräät toteutukset vaativat sen, PostgreSQL ei.

Kuten yllä olevasta huomataan, on malliin nähden tehty kaksi muutosta. Tauluihin legit ja reitit on lisätty kenttä id, joka on asetettu pääavaimeksi. id-kentän tyyppi on serial, joka on itseksensä kasvava, eli aina kun k.o. tauluun lisätään tietue, niin id saa automaattisesti uuden, aiempaa yhtä isomman arvon. id:lle ei siis tarvitse antaa erikseen arvoa, kun k.o. tauluun lisätään tietueita.

Hae nyt taulua legit vastaava komponentti pgAdminin pääikkunasta ja valitse se (joko Object browser-paneelissa tai Properties-paneelissa). Huomaa, että taulua vastaava SQL-koodi on lähes, muttei kuitenkaan täysin sama kuin antamasi komento. Se on kuitenkin funktionaalisesti aivan sama. Avaa taulun ominaisuudet dialogi komennolla Edit | Properties ja valitse välilehti Constraints (kuva).



Kuva 7.4. Taulun ominaisuudet -dialogi, rajoitteet-välilehti

Huomaa, että taulun pääavain on jo valmiiksi määritelty koska teimme sen SQL:llä. Lisätään nyt viiteavaimet. Valitse Foreign Key ja paina Add. Valitse avautuvasta ikkunasta välilehdellä Properties kohtaan References "kaupungit" ja välilehdellä Columns kohtaan Local column "alku" ja kohtaan Referencing "nimi". Paina Add ja OK. Tee nyt samoin viiteavain kentästä loppu taulun "kaupungit" kenttään "nimi". Paina OK taulun "legit" ominaisuudet-dialogissa. Huomaa muutokset taulun SQL-määrittelyssä.



### Huomaa

Olisimme voineet luoda koko taulun pgAdmin:in komennoilla (Taulut-komponentti valittuna komento Edit | New object | New table) ja dialogeilla tai olisimme voineet luoda myös viiteavaimet pelkällä SQL:llä.

Koska reitti koostuu monesta legistä meillä on kaksi mahdollisuutta toteuttaa yhteys näiden entiteettien välillä. Voisimme luoda tauluun reitit kentän, joka on tyyppiä kokonaislukutaulukko ja tallentaa k.o. taulukkaan reittiin kuuluvat legit. Tämä ei kuitenkaan pakottaisi siihen, että taulun alkiot todella ovat viiteauksia olemassaolevaan legiin. Vastaavasti voisimme luoda tauluun legit taulukon, johon tallentaisimme reitit, joihin se kuuluu. Paras tapa on luoda viittausta varten oma taulunsa, johon tulee kaksi kenttää reitti ja legi, ja niille viiteavaimet vastaaviin tauluihin.

Tee uusi taulu dialogeilla. New Table dialogissa anna taulun nimeksi reitit\_legit. Sulkematta dialogia lisää siihen kaksi kenttää reitti ja legi. Molempien tietotyyppi (Data type) on integer. Lisää tauluun vielä pääavaimeksi reitin ja legin yhdistelmä ja viiteavaimet. Sulje sitten dialogi ja päivitä Taulut-komponentin Properties-paneeli.

Tietokanta on nyt valmis. Siinä on neljä taulua ja pääavaimet ja viiteavaimet jotka varmistavat, että tietokantaan syötetty tieto on jossain määrin haluamamme tietomallin mukainen.

## 7.3. Tiedon vieminen tietokantaan

Tallennetaan nyt tietokantaan taulukkolaskentaharjoituksessa käyttämämme tieverkko. Tietokantatauluun voi lisätä tietoa erilaisilla SQL-komennoilla ja pgAdminissa myös dialogin kautta. Käytetään täydellisiä SQL:n insert-lausekkeita.

```
insert into kaupungit (nimi,asluuku) values ('Nuorgam','200');
insert into kaupungit (nimi,asluuku) values ('Kilpisjärvi','114');
insert into kaupungit (nimi,asluuku) values ('Rovaniemi','60038');
insert into kaupungit (nimi,asluuku) values ('Oulu','141284');
insert into kaupungit (nimi,asluuku) values ('Vaasa','59633');
insert into kaupungit (nimi,asluuku) values ('Jyväskylä','130857');
insert into kaupungit (nimi,asluuku) values ('Kuusamo','16494');
insert into kaupungit (nimi,asluuku) values ('Tampere','213143');
insert into kaupungit (nimi,asluuku) values ('Lappeenranta','72035');
insert into kaupungit (nimi,asluuku) values ('Turku','177504');
insert into kaupungit (nimi,asluuku) values ('Helsinki','588195');
```

Asukasluvut on poimittu Wikipediasta. Aja komennot SQL-ikkunan kautta ja tarkista tulos katsomalla taulun tietoja. Lisätään seuraavaksi legit.

```
insert into legit (alku,loppu,pituus) values ('Kilpisjärvi','Nuorgam',651);
insert into legit (alku,loppu,pituus) values ('Rovaniemi','Nuorgam',499);
insert into legit (alku,loppu,pituus) values ('Kilpisjärvi','Rovaniemi',433);
insert into legit (alku,loppu,pituus) values ('Kilpisjärvi','Oulu',597);
```

```
insert into legit (alku,loppu,pituus) values ('Rovaniemi','Oulu',222);
insert into legit (alku,loppu,pituus) values ('Rovaniemi','Kuusamo',192);
insert into legit (alku,loppu,pituus) values ('Oulu','Vaasa',319);
insert into legit (alku,loppu,pituus) values ('Oulu','Jyväskylä',340);
insert into legit (alku,loppu,pituus) values ('Oulu','Kuusamo',212);
insert into legit (alku,loppu,pituus) values ('Vaasa','Jyväskylä',283);
insert into legit (alku,loppu,pituus) values ('Vaasa','Tampere',244);
insert into legit (alku,loppu,pituus) values ('Jyväskylä','Kuusamo',552);
insert into legit (alku,loppu,pituus) values ('Jyväskylä','Tampere',151);
insert into legit (alku,loppu,pituus) values ('Jyväskylä','Lappeenranta',279);
insert into legit (alku,loppu,pituus) values ('Jyväskylä','Helsinki',271);
insert into legit (alku,loppu,pituus) values ('Kuusamo','Lappeenranta',746);
insert into legit (alku,loppu,pituus) values ('Tampere','Turku',153);
insert into legit (alku,loppu,pituus) values ('Tampere','Helsinki',173);
insert into legit (alku,loppu,pituus) values ('Lappeenranta','Helsinki',188);
insert into legit (alku,loppu,pituus) values ('Turku','Helsinki',165);
```

Ylläoleva lisää kunkin legin vasta yhteen suuntaan. Toinen suunta voidaan lisätä yksinkertaisella insert-komennolla.

```
insert into legit (alku,loppu,pituus) select loppu,alku,pituus from legit;
```

Lisätään lopuksi kaksi reittiä reitit-tauluun.

```
insert into reitit (kuvaus) values ('Kesäloma');
insert into reitit (kuvaus) values ('Hiihtoloma');
```

Suunnittele nyt kesäloma- ja hiihtolomamatka käyttäen Edit Data -dialogeja niin, että katsot reittejä ja legejä vastaavat id:t ja lisäät niitä tauluun reitit\_legit. Hiihtolomamatka on Helsinki-Lappeenranta-Kuusamo-Jyväskylä-Helsinki ja kesälomamatka on Helsinki-Tampere-Vaasa-Oulu-Kilpisjärvi-Nuorgam-Rovaniemi-Oulu-Jyväskylä-Helsinki. Kun lisäät Edit Data dialogin kautta tietoa, tieto ei päivity tietokantaan ennenkuin olet antanut Refresh-komennon.

### 7.4. Vastausten hakeminen tietokannasta

Tietokannasta haetaan tietoa select-lauseella, jossa määritellään joukko kenttiä joiden arvot halutaan, joukko tauluja joissa halutut kentät ovat tai jotka liittyvät muuten hakuun, sekä mahdollisesti joukko ehtoja, joilla hakua ohjataan.

Haetaan ensin kesälomamatkan legit.

```
select alku,loppu from reitit,legit,reitit_legit where
reitit.kuvaus='Kesäloma' and
reitit_legit.reitti=reitit.id and
reitit_legit.legi=legit.id
```

Huomaa, että viiteavaimia pitää eksplisiittisesti käyttää haussa, ne eivät ole mukana mitenkään automaattisesti. Huomaa, että kentän määrittelyssä voi olla mukana taulun nimi: <taulun nimi>.<kentän nimi>, jos tauluissa on samannimisiä kenttiä tai muuten vain selvyuden vuoksi. Huomaa, että tuloksessa legit ovat (todennäköisesti) ajojärjestyksessä koska syötit ne siinä järjestyksessä. Tulostaulun tietueiden järjestys on tässä täysin mielivaltainen ja toteutuksesta riippuva. Oletetaan seuraavaksi, että tämä tietokanta on auton tietojärjestelmässä johon tulee tieto, että auto on Oulussa. Tehdään kysely, jolla saadaan selville reitin seuraava etappi.

```
select alku,loppu from reitit,legit,reitit_legit where
```



```
reitit.kuvaus='Kesäloma' and  
reitit_legit.reitti=reitit.id and  
reitit_legit.legi=legit.id and  
legit.alku='Oulu'
```

Huomataan, että Oulussa käydään matkan aikana kaksi kertaa ja seuraava etappi riippuu siitä missä vaiheessa matkaa ollaan.

Seuraavaksi halutaan laskea hiihtolomamatkalla vierailtavien kaupunkien yhteenlaskettu asukasmäärä.

```
select sum(asluku)  
from reitit,legit,reitit_legit,kaupungit where  
reitit.kuvaus='Hiihtoloma' and  
reitit_legit.reitti=reitit.id and  
reitit_legit.legi=legit.id and  
kaupungit.nimi=legit.alku
```

## 7.5. Raportointi

Raportoi tietokannan toteutus kuvaamalla se lyhyesti sanallisesti ja liitä mukaan SQL-koodi, joka tuottaa tietokannan.

Kuvaa tietokannan käyttömahdollisuuksia (myös muita kuin mitä tässä on esitetty) ja kuvaa miten sen avulla vastataan näitä käyttötapauksia vastaaviin kysymyksiin. Liitä mukaan kysymysten muotoilu SQL-koodina. Mieti erityisesti ongelmaa, johon yllä päädyttiin, eli kuinka erotetaan seuraava etappi matkoilla, joilla käydään samassa paikassa useammin kuin kerran.



---

# Paikkatietoharjoitus

Tässä harjoituksessa syvennetään paikkatietoaineistojen ja -menetelmien tuntemusta. Paikkatietoaineistoihin on aiemmin tutustuttu aineistoharjoituksessa.

Harjoitus keskittyy ns. vektorimuotoiseen paikkatietoon, joka on tallennettu tietokantaan ja jota muokataan ja analysoidaan siellä. Geometrisiä predikaatteja hyödynnetään tiedonhaussa.

## 8.1. Paikkatietoa sisältävien tietokantataulujen luominen

Yksinkertainen paikkatieto, joka voidaan esittää kohteen geometria-ominaisuutena, voidaan tallentaa tietokantaan mikäli käytetty tietokantajärjestelmä mahdollistaa sen. Tässä harjoituksessa käytetään PostgreSQL-tietokantajärjestelmän PostGIS-laajennusta, joka määrittelee tietotyypin geometry ja sille joukon funktioita.

Luodaan ensin tietokantataulu, jossa on edellisen tehtävän kaupungit mutta uutena ominaisuutena kaupungin sijainti. Edellisessä tehtävässä luotua taulua on hankala käyttää koska Quantum GIS -ohjelmaa varten pääavaimen on paras olla tyyppiä integer.

```
create table kohteet (id serial primary key,  
nimi text);  
insert into kohteet (nimi)  
select nimi from kaupungit;
```

Uuteen tauluun täytyy vielä lisätä varsinainen paikkatieto.

Olet varmasti huomannut, että tietokannassasi oli jo alussa kaksi taulua geometry\_columns ja spatial\_ref\_sys. Ne kuuluvat standardiin tietokannan paikkatietolaajennukseen. Edellinen on tietokannassa olevien kohdeluokkien määrittelyä varten ja jälkimmäinen sisältää koordinaatistojärjestelmätiedot. Lisätään tauluun kohteet paikkatietokenttä ja uusi paikkatietokohdeluokka tauluun geometry\_columns.

```
select AddGeometryColumn('','kohteet','the_geom',4326,'POINT',2);
```



### Huomaa

Paikkatietokentän voi lisätä tauluun aivan samoin kuin minkä tahansa kentän mutta kun käyttää AddGeometryColumn-funktiota, niin se samalla lisää kentän geometry\_columns-tauluun ja lisää datatauluun tiettyjä rajoitteita.

Tämä tieto kertoo, että kohteet taulussa on kentässä the\_geom pistemäinen geometriaominaisuus. Kukin piste on kaksiulotteinen (x ja y koordinaatit on annettu). Pisteiden koordinaatit on annettu koordinaatistojärjestelmässä 4326, joka on kentän srid arvo taulussa spatial\_ref\_sys. K.o. taulusta voi lukea että se tarkoittaa järjestelmää, jonka EPSG arvo on sama 4326 eli WGS 84, joka on maantieteellisten koordinaattien (pituus- ja leveysasteiden) järjestelmä. Kaupunkien koordinaatit WGS84 järjestelmässä löytyvät helposti esim. Wikipedian avulla. Wikipediassa koordinaatit on annettu asteina, minuutteina ja sekunteina, jota klikkaamalla pääsee toolserver.org:in sivulle, jossa ne on muunnettu desimaalilukuina annetuiksi asteiksi. toolserver.org:in sivulla koordinaatit on järjestyksessä leveyspiiri ja pituuspiiri. ST\_GeomFromText-funktiolle ne annetaan WKT-muodossa (POINT (x y)), jonka x on tässä tapauksessa pituuspiiri ja y leveyspiiri eli päinvastoin kuin verkkosivulla.

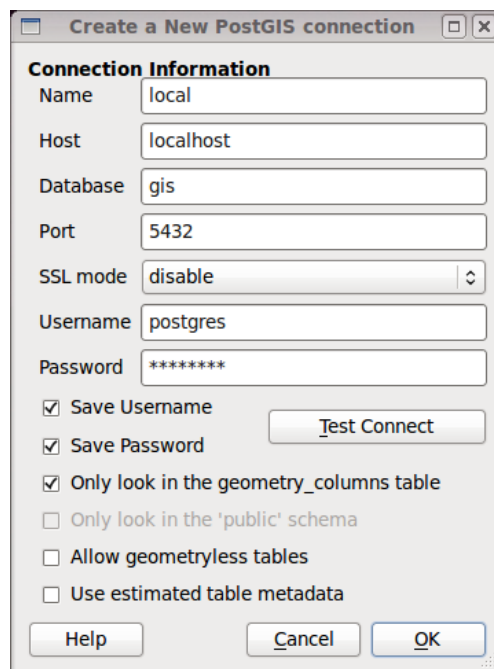
```
update kohteet set the_geom = st_geomfromtext('POINT(24.9375 60.170833)', 4326) where nimi = 'Helsinki';
update kohteet set the_geom = st_geomfromtext('POINT(22.266667 60.45)', 4326) where nimi = 'Turku';
update kohteet set the_geom = st_geomfromtext('POINT(28.183333 61.066667)', 4326) where nimi = 'Lappeenranta';
update kohteet set the_geom = st_geomfromtext('POINT(23.766667 61.5)', 4326) where nimi = 'Tampere';
update kohteet set the_geom = st_geomfromtext('POINT(29.183333 65.966667)', 4326) where nimi = 'Kuusamo';
update kohteet set the_geom = st_geomfromtext('POINT(25.741667 62.241667)', 4326) where nimi = 'Jyväskylä';
update kohteet set the_geom = st_geomfromtext('POINT(22 63.166667)', 4326) where nimi = 'Vaasa';
update kohteet set the_geom = st_geomfromtext('POINT(25.466667 65.016667)', 4326) where nimi = 'Oulu';
update kohteet set the_geom = st_geomfromtext('POINT(25.733333 66.5)', 4326) where nimi = 'Rovaniemi';
update kohteet set the_geom = st_geomfromtext('POINT(20.794444 69.049167)', 4326) where nimi = 'Kilpisjärvi';
update kohteet set the_geom = st_geomfromtext('POINT(27.875 70.080556)', 4326) where nimi = 'Nuorgam';
```

Aineisto on nyt valmis ja voimme katsoa sitä paikkatieto-ohjelmalla.

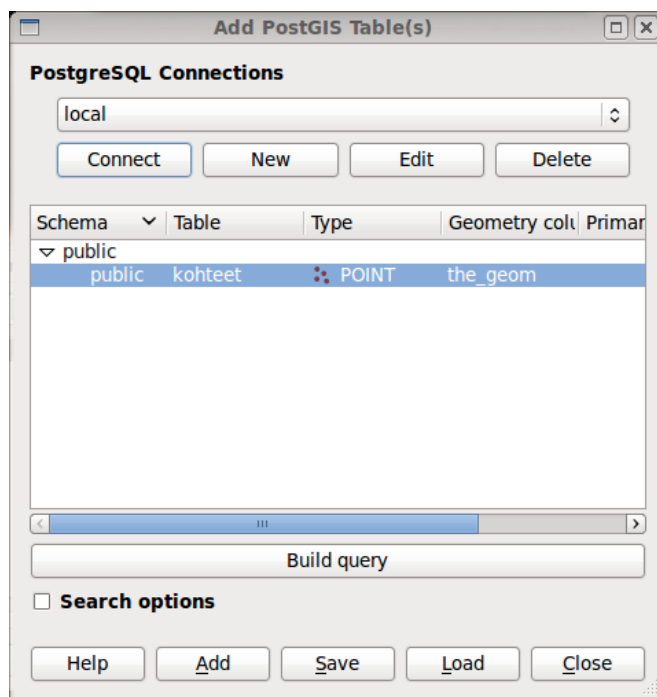
## 8.2. Paikkatiedon avaaminen tietokannasta paikkatieto-ohjelmaan

Käytetään aineistojen visualisointiin samaa paikkatieto-ohjelmaa kuin aineistoharjoituksessa eli Quantum GIS:iä. Quantum GIS:iin tuodaan paikkatietoa tietokannasta komennolla Layer | Add PostGIS Layer. Avautuvassa dialogissa täytyy ensin määrittää yhteys tietokantapalvelimeen ja sen tarjoamaan tiettyyn tietokantaan. Anna yhteydelle kuvaava nimi ja muille parametreille seuraavia arvoja Host=water.hut.fi, Database=<oma tietokantasi>, Port=5432, Username ja Password kuten pgAdmin III yhteydessä.

Kun yhteys toimii saat sillä Add PostGIS Table -dialogin paneeliin listan tietokannassa olevista kohde- luokista. Quantum GIS antaa tässä vaiheessa mahdollisuuden tehdä valittuun tauluun lähes mielival- taisen SQL-kyselyn Build query -työkalulla mutta tässä harjoituksessa riittää yleensä kohdejoukon li- sääminen tasona ohjelmaan sellaisenaan.

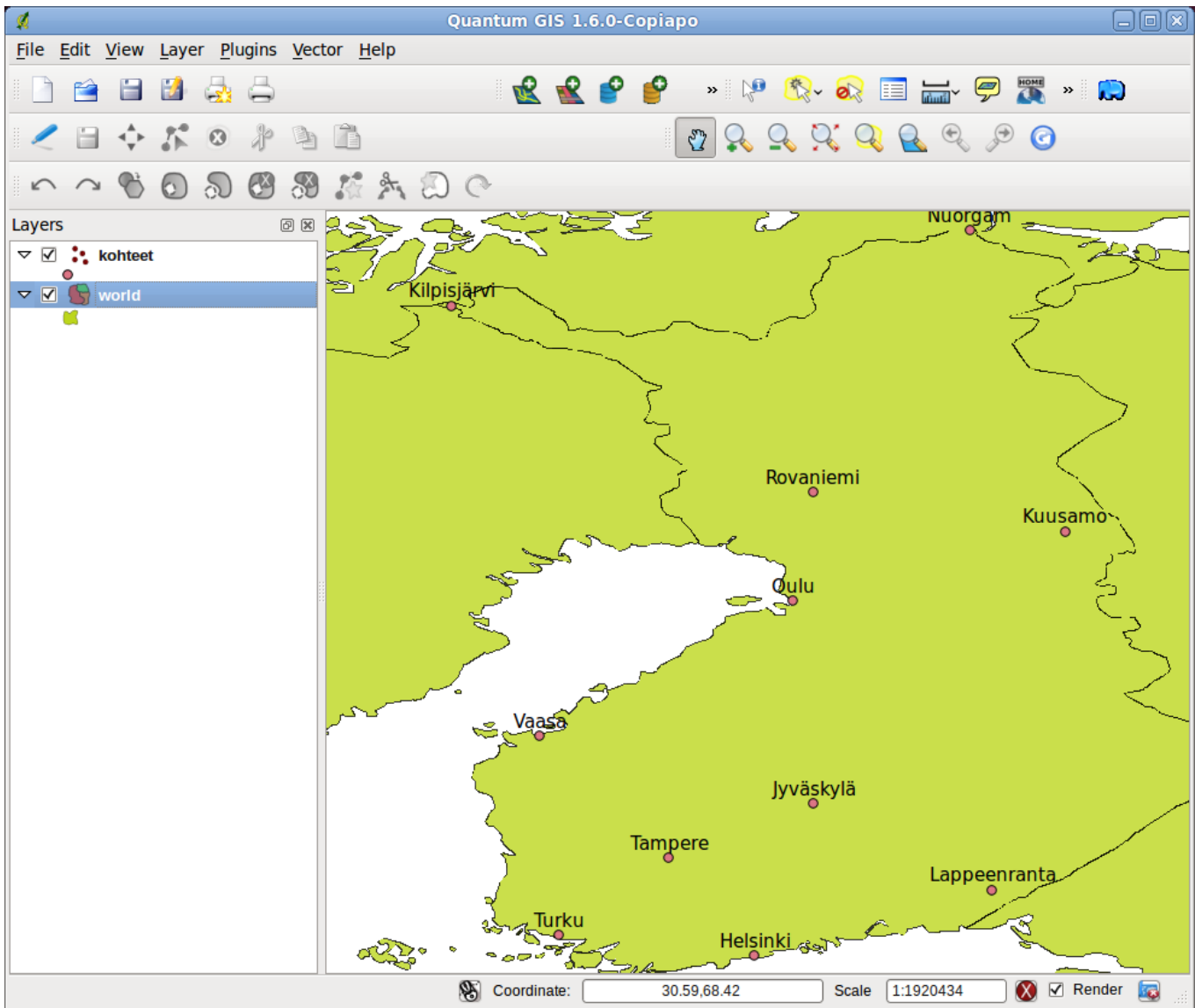


Kuva 8.1. Yhteyden luominen tietokantapalvelimeen ja tietokantaan



Kuva 8.2. Tietokannassa olevan kohdejoukon lisääminen paikkatietokerroksena

Lisää vielä maailmankartta tietokannasta geodata water.hut.fi:n tietokantapalvelimesta ja laita kaupunkien nimet karttaan niin saat oheisen mukaisen karttakuvan.



Kuva 8.3. Karttakuva jossa taustalla on maailmankartta ja edustalla kaupungit uudessa paikkatietoaineistossa

### 8.3. Paikkatietoanalyysit tietokannassa

Tietokantojen varsinainen hyöty tulee esiin, että niiden avulla voidaan vastata kysymyksiin tallennetun tiedon avulla. Paikkatietolaajennus määrittelee tietokantajärjestelmään ja sen SQL-kieleen ison joukon funktioita, joita voidaan käyttää hakuehdoissa, geometrioiden muokkaamisessa ja niin edelleen. PostGIS-laajennuksen (version 1.5) dokumentaatio on sivulla <http://postgis.refractory.net/documentation/manual-1.5/>.

Tässä paikkatietoanalyysissä on tarkoitus selvittää kuinka iso osa Otaniemen rakennuksista on enintään 300 metrin päässä bussipysäkistä. Lähtötietoina annetaan Espoon postinumeroalueet, Espoon bussipysäkit ja Otaniemen ja sen lähiympäristön rakennukset ESRI Shapefile -formaattissa. Kaikki lähtötiedot on annettu KKK 2 -koordinaattijärjestelmässä. Kun kaikki lähtötiedot on siirretty tietokantaan ne voidaan visualisoida.

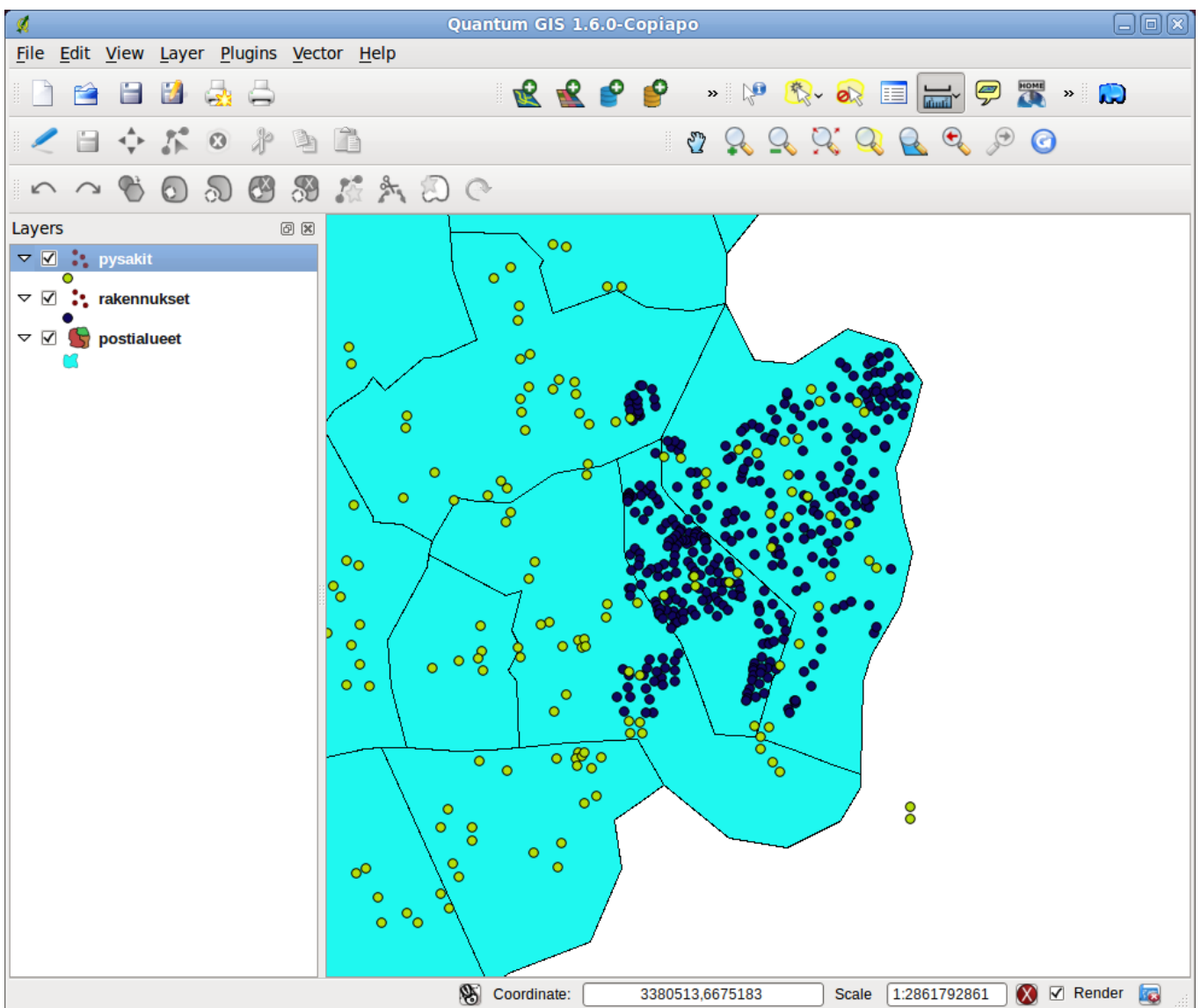


## Paikkatietoaineiston vieminen tietokantaan

PostGIS-ohjelmistoon kuuluu shp2pgsql-komentoriviohjelma, jolla ESRI Shapefile -aineisto voidaan muuttaa SQL-muotoon, jossa aineisto voidaan viedä tietokantaan. K.o. ohjelma tuottaa shapefilestä SQL-kielisen insert-komentosarjan, joka ajettaessa luo tietokantaan taulun. Optio -W mahdollistaa merkistömuunnokset UTF-8:aan (esimerkissä aineisto.dbf:n merkkijonot ovat UTF-8 muodossa). Optio -s määrittelee aineiston karttaprojektion. EPSG-koodi 2392 on KKJ 2.

```
shp2pgsql -W UTF-8 -s 2392 aineisto.shp taulu >taulu.sql
```

Quantum GIS:in uudemmissa versioissa on plugin, jolla voidaan tehdä sama. Avoimen lähdekoodin maailmassa on vielä mm. ogr2ogr-ohjelma, jolla voidaan siirtää vektoriaineistoja tallennusmuodosta toiseen. Kaikkein joustavin työkalu on mikä tahansa ohjelmointikieli, jolla voidaan kirjoittaa ohjelma, joka tuottaa SQL:n insert-komentoja halutusta datasta.



Kuva 8.4. Otaniemen alueen rakennukset ja pysäkit postinumeroaluekartan päällä

## Luku 8. Paikkatietoharjoitus

---

Poimitaan ensin rakennukset, jotka ovat Otaniemessä, erilleen.

```
create table r_otaniemi (gid int primary key);
select AddGeometryColumn("",
  'r_otaniemi','the_geom',2392,'POINT',2);
insert into r_otaniemi
select rakennukset.gid,rakennukset.the_geom from
rakennukset,postinumerot
where within(rakennukset.the_geom,
  postinumerot.the_geom)
and postinumerot.posti='02150';
```

Luodaan sitten alue, jonka pisteet ovat enintään 300 metrin päässä pysäkestä käyttämällä collect ja buffer-funktioita.

```
create table buf (id serial primary key);
select AddGeometryColumn("",
  'buf','the_geom',2392,'MULTIPOLYGON',2);
insert into buf (the_geom)
select buffer(collect(the_geom),300) from pysakit;
```

Tämän vaiheen voi tehdä monella tavalla mutta collect-funktion käyttäminen on hyvin kätevää. Tutkitaan nyt kuinka monta rakennusta on pysäkkien läheisyydessä ja kuinka monta niitä on yhteensä.

```
select count(r_otaniemi.gid) from r_otaniemi;
```

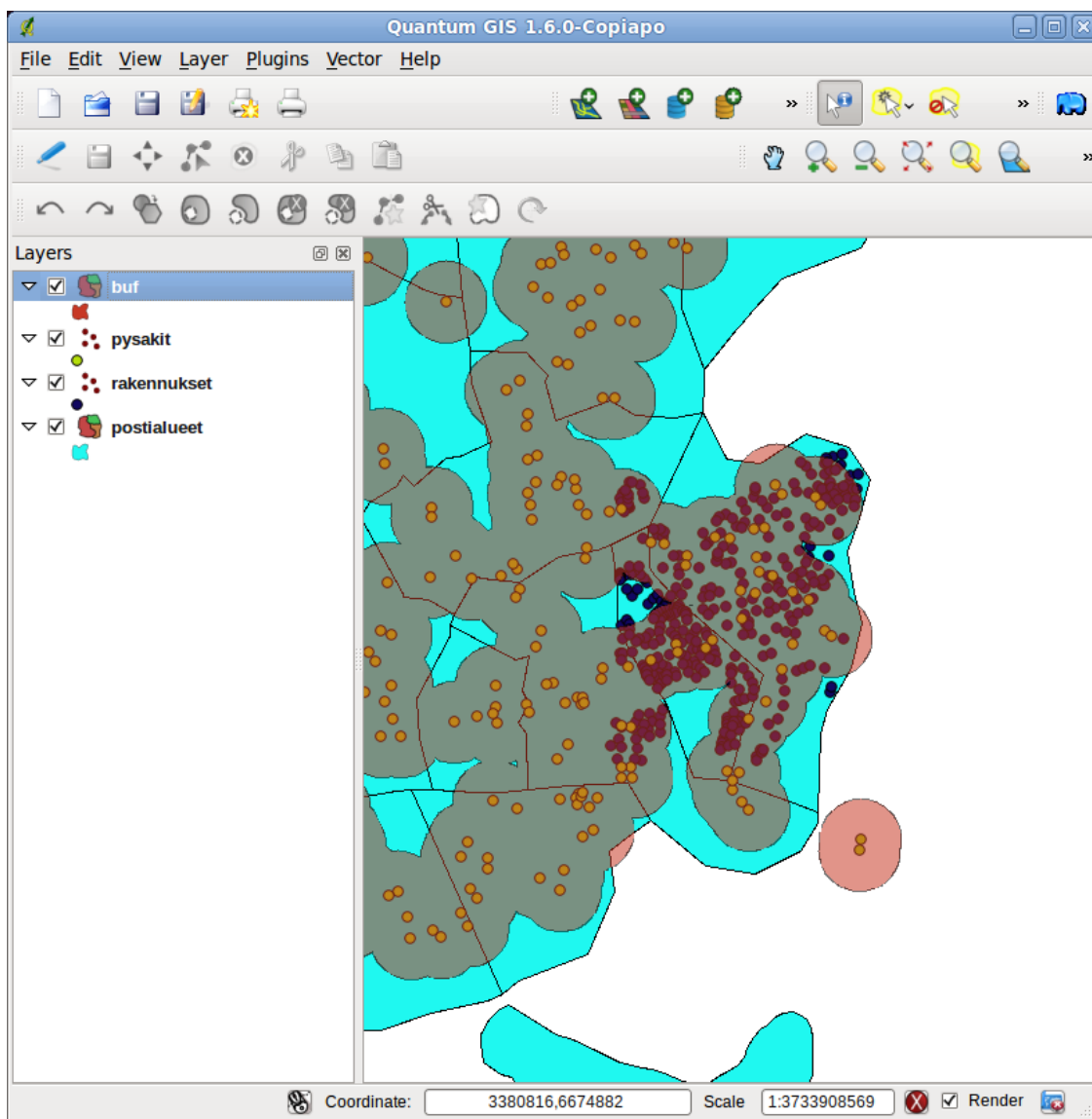
Antaa tulokseksi 172.

```
select count(r_otaniemi.gid) from r_otaniemi,buf
where within(r_otaniemi.the_geom,buf.the_geom);
```

Antaa tulokseksi 164. Taloista on siis  $(164/172)*100\% = 95\%$  enintään 300 metrin päässä pysäkestä.

Paikkatietoa olevien tulosten visualisointi on hieman hankalaa Quantum GIS -ohjelmalla. Quantum GIS vaatii, että taululla on oltava pääavain ja käytännössä myös geometry\_columns taulun käyttäminen näyttää olevan pakollista (vaikka yhteys-dialogissa se onkin optiona).





Kuva 8.5. Pysäkeistä enintään 300 metrin etäisyydellä olevat alueet

## 8.4. Raportointi

Raportoi tekemäsi analyysi ja kuvita se QGISillä tekemilläsi karttakuvilla.



---

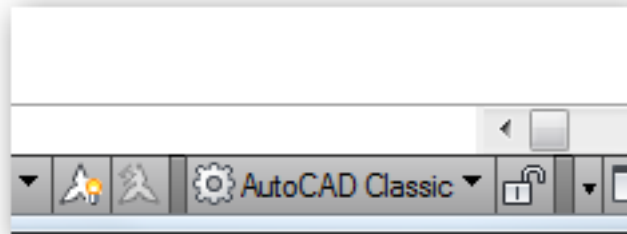
# AutoCAD käyttöharjoitus

Tämän käyttöharjoituksen tehtävänä on tutustuttaa opiskelija käyttöliittymän perusominaisuuksiin harjoitustehtävän kautta. Harjoitusta voi tehdä omaan tahtiin, mutta jos AutoCADin käyttö tuntuu vieraalta, kannattaa tehtävää tehdä assistentin perässä. Tässä harjoituksessa kerrotaan vain harjoituksen kannalta oleelliset käyttöliittymän ominaisuudet ja toiminnot. Lisää ohjelman käytöstä ja toiminnoista voit lukea oheisesta käyttöoppaasta.

Tämä käyttöharjoitusmateriaali on tehty AutoCADin versiolle 2010, mutta se soveltuu suurimmaksi osaksi myös muille versioille.

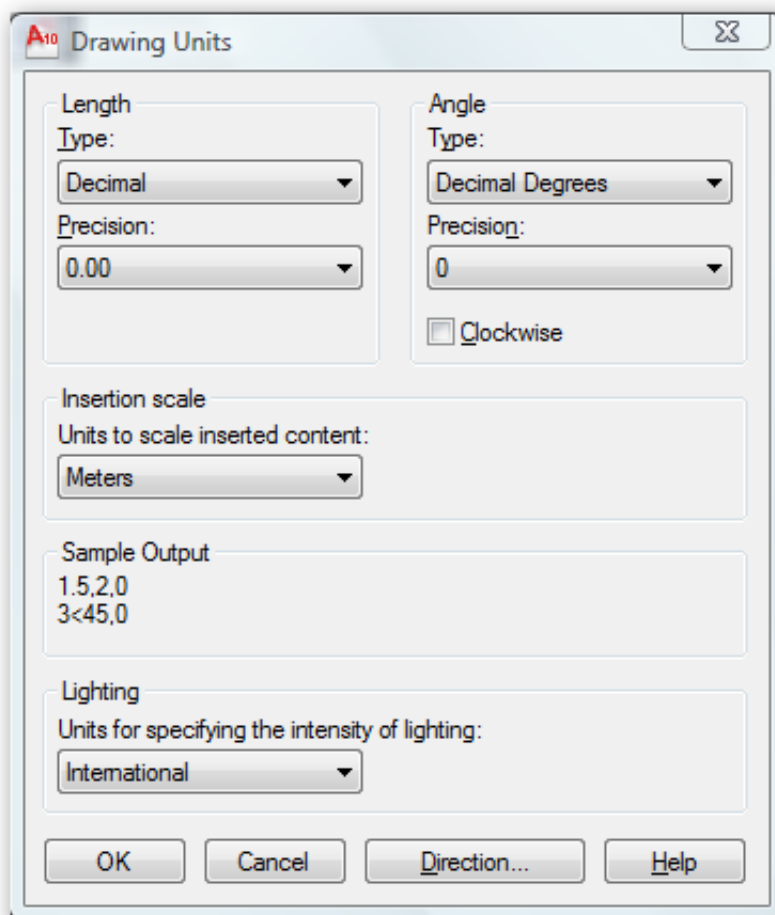
## 9.1. Käyttöliittymän perusominaisuudet

Käynnistä AutoCAD 2010 ja tarkista, että työtila on AutoCAD classic -niminen. Työtilaa voit vaihtaa ohjelman oikeassa alareunassa olevasta workspace switching -valikosta.



Kuva 9.1. Workspace switching -valikko

Käytettävät piirtoyksiköt määritellään komennon **units** kautta tai valitsemalla **Format** -alasuvalikosta **units**. Monista komennoista voidaan käyttää myös lyhenteitä eli aliaksia. Units komennon lyhenne on **un**. Komennolla avautuu seuraavan kuvan mukainen ikkuna.



Kuva 9.2. Piirtoyksiköiden määrittäminen

Length-kohdasta määritetään käytettävän piirtoyksikön tyyppi ja desimaalien lukumäärä. Angle-kohdasta asetetaan kulman tyyppi ja desimaalien lukumäärä. Käytettävä pituusyksikkö asetetaan kohdasta Insertion scale. Määritä asetukset kuvan mukaisiksi.



### Huomaa

Kulman oletusarvoinen positiivinen kiertosuunta on vastapäivään. Suunta voidaan vaihtaa myötäpäivään klikkaamalla Angle-kohdasta Clockwise. 0-kulman suuntaa voidaan määrittää Direction...-painikkeen kautta.

Seuraavaksi asetetaan työskentelyalueen rajat. Tämä tapahtuu **limits** komennon kautta tai valitsemalla **Format**-alavasetovalikosta **Drawing limits**. Ohjelma kysyy vasemman alakulman ja oikean yläkulman koordinaatteja. Pisteverkko piirretään vain syötettyjen rajojen sisään. Vasemmaksi alakulmaksi asetetaan origo, jota ohjelma ehdottaa valmiiksi. Koordinaatit voidaan valita painamalla **Enter**. Oikeaksi yläkulmaksi asetetaan 10 metriä x-suuntaan ja 4 metriä y-suuntaan. Huomaa, että ohjelma käyttää desimaalien erotukseen pistettä!

```
Command: limits
Reset Model space limits:
Specify lower left corner or [ON/OFF] <0.00,0.00>:
Specify upper right corner <420.00,297.00>: 10.00,4.00
```

Kuva 9.3. Työskentelyalueen rajojen määrittäminen

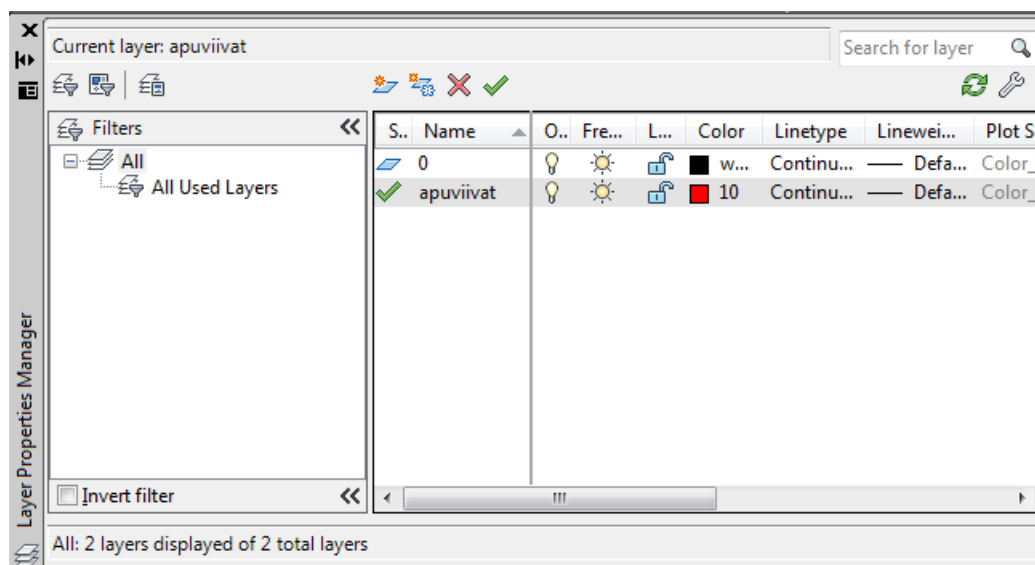


### Huomaa

Pisteverkko (Grid) voidaan kytkeä päälle joko funktionäppäimellä **F7** tai komentorivin tilarivin **GRID**-painikkeella. Pisteverkko koostuu joukosta pieniä pisteitä ja niiden tiheys x- ja y-suunnassa voidaan määrittää erikseen.

## 9.2. Ohjattu harjoitus

Nyt aloitamme varsinaisen kuvan piirtämisen. Aloitetaan piirtämällä apuviivat varsinaisen kuvan piirtämisen helpottamiseksi. Luodaan apuviivoille oma taso eli layer. Ikkuna tasojen määrittelyyn ja hallintaan saadaan avattua komennolla **layer** tai **la**. Luodaan uusi taso klikkaamalla kuvaketta, jossa on sinisävyinen laatta ja vasemmassa yläkulmassa tähti (New Layer). Tasoikkunaan avautuu uusi taso, jonka nimi on Layer1. Vaihdetaan tason nimeksi apuviivat ja valitaan väriksi punainen. Asetetaan taso tämänhetkiseksi piirtotasoksi klikkaamalla tasoikkunan yläpuolella olevaa vihreää merkkiä (Set Current). Annetaan muiden asetusten olla oletusarvoisina ja suljetaan ikkuna sen vasemmassa yläkulmassa olevasta rastista.



Kuva 9.4. Uuden tason määrittäminen

Ennen piirtämisen aloittamista zoomataan ikkunaan piirustuksen rajojen määrittämä alue. Tämä tapahtuu komennolla **zoom**, jonka jälkeen kirjoitetaan komento **all**.

```
Command: zoom  
Specify corner of window, enter a scale factor (nX or nXP), or  
[All/Center/Dynamic/Extents/Previous/Scale/Window/Object] <real time>: all  
generating model.
```

Kuva 9.5. Zoomaaminen

Piirtäminen tapahtuu komennolla **polyline** tai **pl**. Asetetaan aloituspisteeksi origo. Valitaan tämän jälkeen käyttöön **Ortho** pikanäppäimellä **F8**. Ortho pakottaa piirrettävän viivan suunnan x- tai y-akselin suuntaiseksi. Asetetaan suunta hiirellä y-akselin suuntaiseksi ylöspäin ja kirjoitetaan pituudeksi 2.00. Kuvaan ilmestyy kahden metrin mittainen punainen viiva ylöspäin origosta.

```
Command: pl  
PLINE  
Specify start point: 0,0  
Current line-width is 0.00  
Specify next point or [Arc/Halfwidth/Length/Undo/Width]: <Ortho on> 2.00
```

Kuva 9.6. Viivan piirtäminen

Piirtämistä voidaan jatkaa suoraan piirretyn viivan päätepisteestä. Lopetetaan piirtäminen kuitenkin tähän viivaan ja harjoitellaan offset-komennon sekä pakotusten käyttöä. Komennon saa lopetettua painamalla **Esc**-näppäintä.

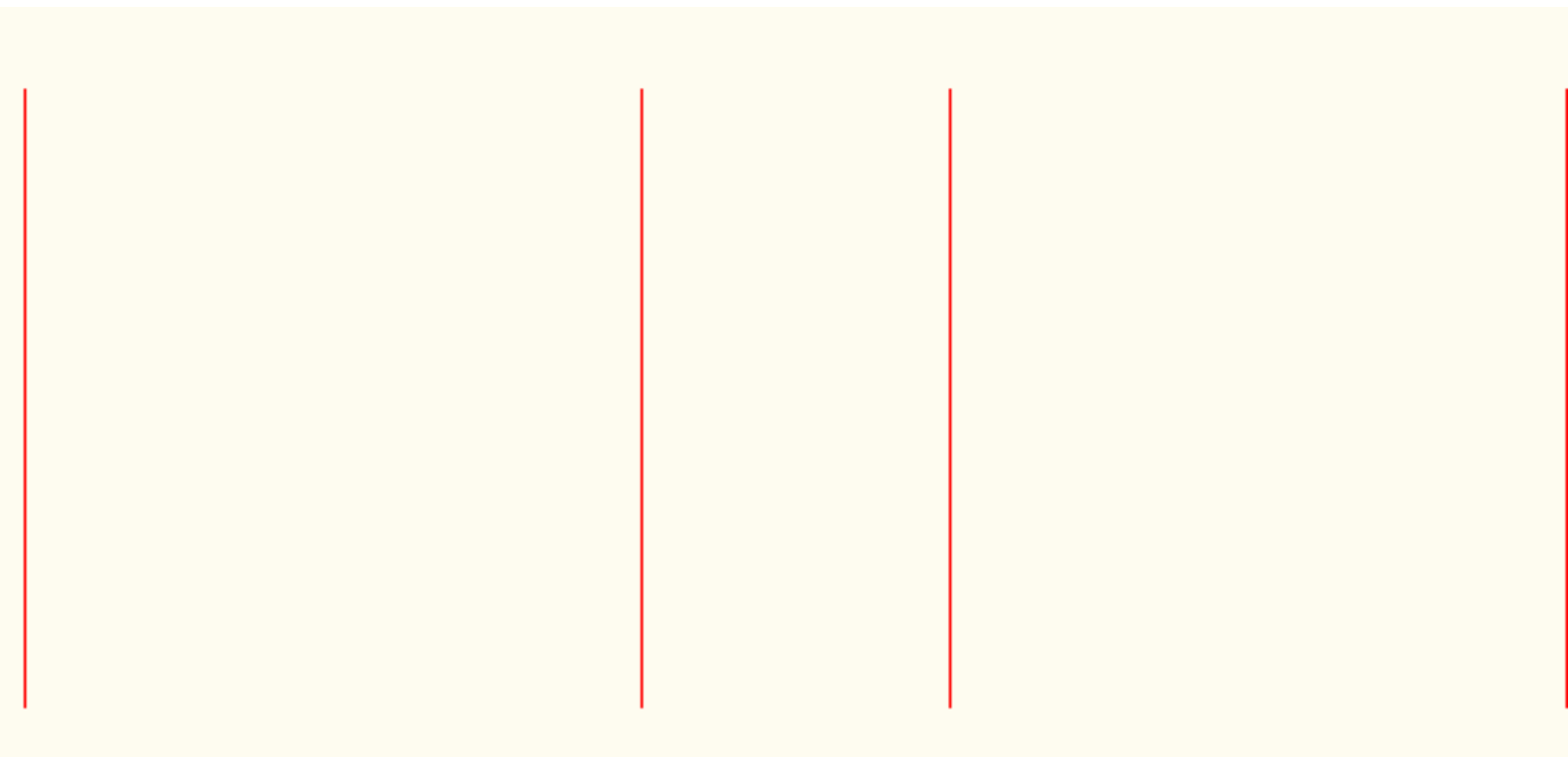


### Huomaa

Polyline-komento piirtää viivasta yhtenäisen. Komennolla peräkkäin piirrettyjä viivoja ei voi siis muokata erikseen. Mikäli halutaan yksittäin muokattavia viivoja, on käytettävä komentoa **line** tai **l**.

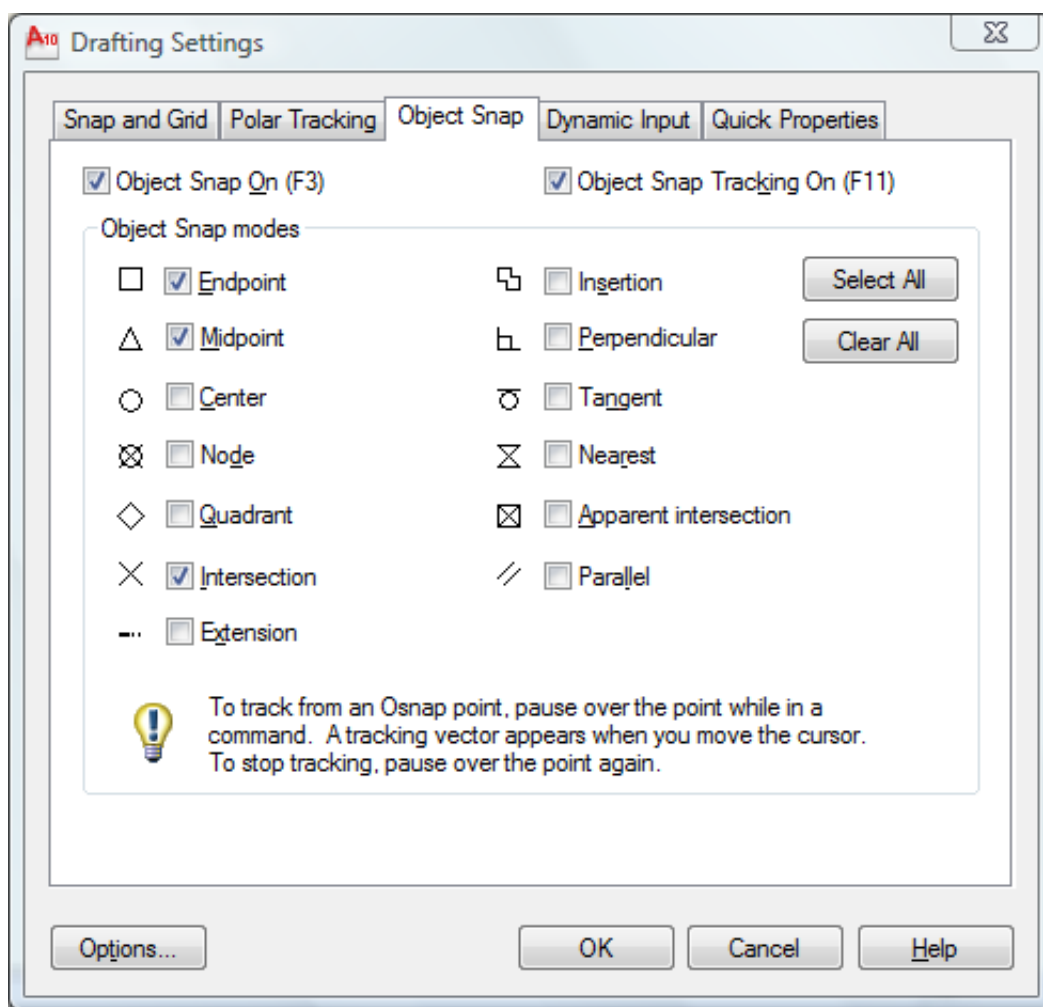
Komennolla **offset** tai **o** saadaan luotua yhdensuuntainen kopio viivasta tai muusta elementistä. Kirjoitetaan komennoksi **o**, annetaan etäisyydeksi 1, klikataan kopioitavaa objektia eli piirrettyä punaista viivaa ja klikataan hiirellä viivan oikealle puolelle. Kuvaan ilmestyy nyt kopio alkuperäisestä viivasta metrin päähän siitä sen oikealle puolelle. Uusi viiva halutaan nyt kopioida kahden metrin päähän oikealle. Koska etäisyys vaihtuu, lopetetaan komento **Esc**-näppäimellä ja tehdään sama komento uudestaan, eli kirjoitetaan komennoksi **o**. Kirjoitetaan etäisyydeksi 2, klikataan äsken piirrettyä viivaa ja klikataan aluetta sen oikealla puolella.

Toistetaan offset-komento niin monta kertaa, että viivoja on yhteensä 6, ja niiden etäisyydet toisistaan ovat vasemmalta oikealle lukien 1, 2, 1, 2, 1. Kuvan pitäisi tämän jälkeen näyttää seuraavalta:



Kuva 9.7. Kuva offset-komentojen jälkeen

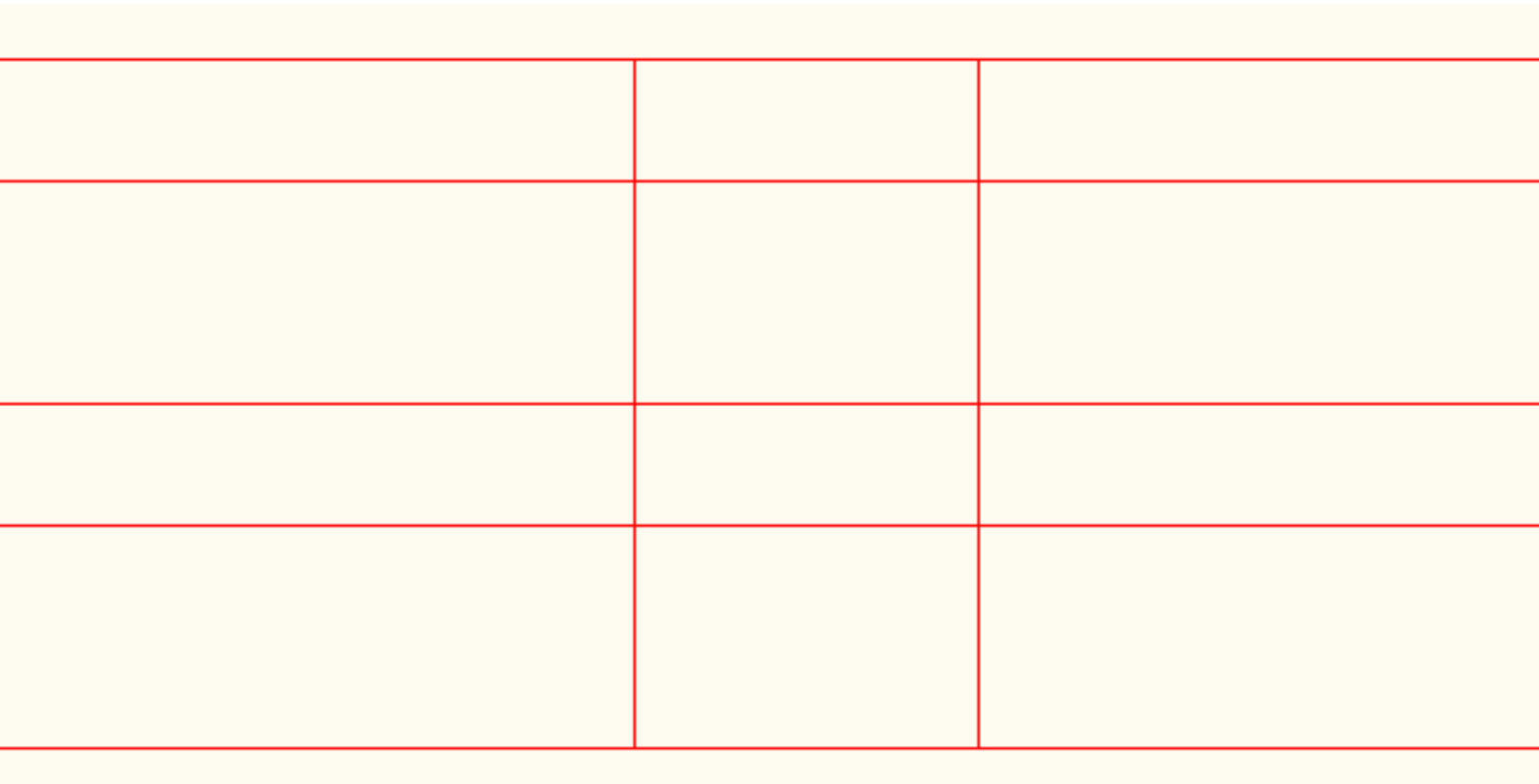
Pakotukset eli Snap-asetukset ovat hyödyllisiä silloin, kun uuden kuva-alkion piirtäminen aloitetaan olemassa olevasta kuva-alkiosta. Pakotuksia käyttämällä käyttäjä pystyy osoittamaan hiirellä kohdan, josta aloitetaan uuden alkion piirtäminen, esimerkiksi viivan pään. Pakotusasetuksia päästään muokkaamaan komennolla **osnap** tai **os**. Valitaan avautuvan ikkunan välilehdeltä Object snap pakotuspisteiksi **Endpoint**, **Midpoint** ja **Intersection** ja klikataan **OK**.



Kuva 9.8. Pakotusten asettaminen

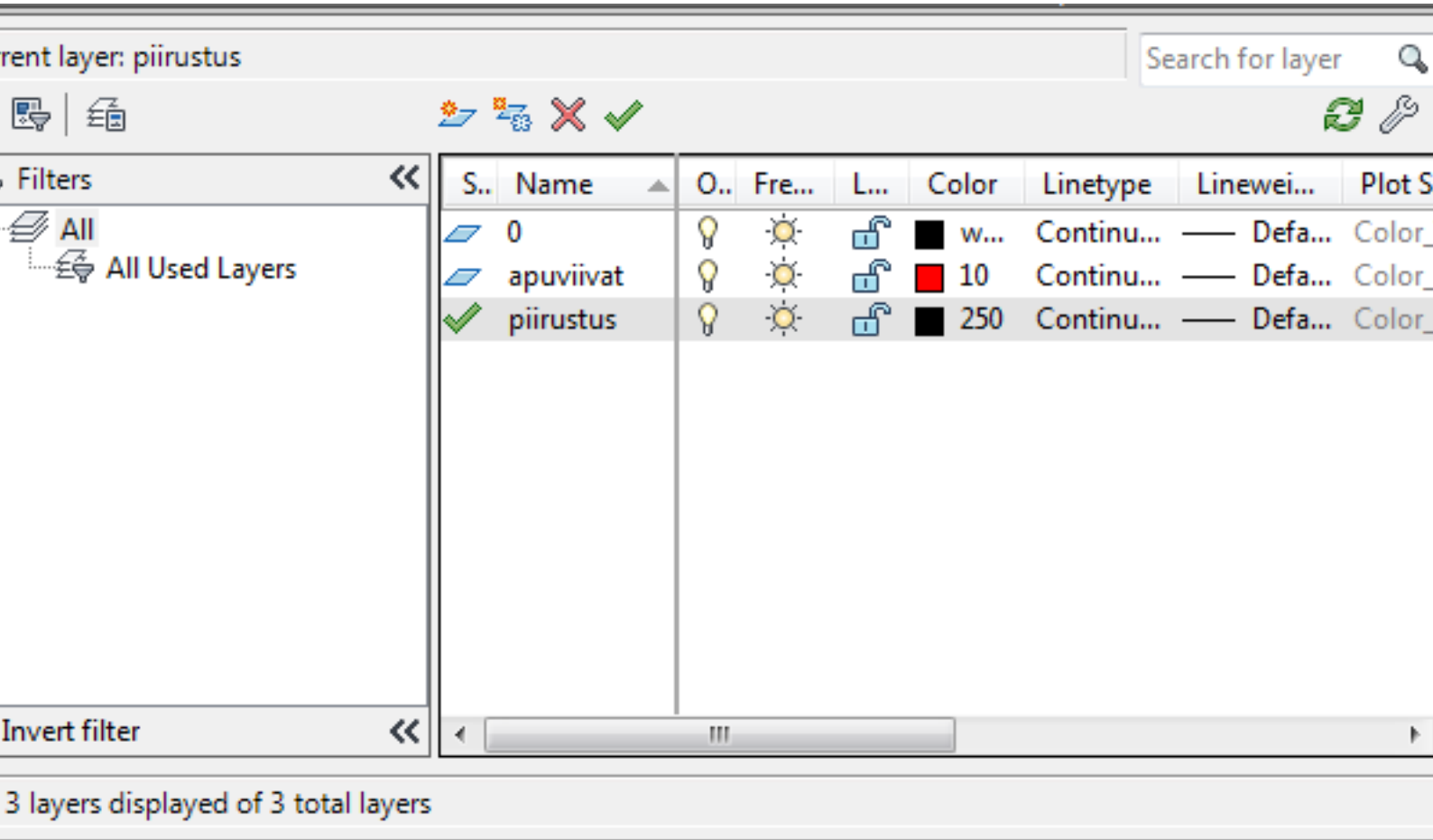
Jatketaan nyt apuviivojen piirtämistä. Kirjoitetaan komennoksi **pl**, asetetaan hiirellä lähtöpisteeksi kaikkein vasemmanpuoleisimman viivan yläpää ja loppuviivaksi kaikkein oikeanpuoleisimman viivan yläpää. Kopioidaan viiva äsken käytetyllä komennolla **o** neljä kertaa, jokainen viiva aina edellisen alapuolelle niin että etäisyydet ovat aina edellisestä viivasta lukien 0.35, 0.65, 0.35, 0.65. Kuvan tulisi tämän jälkeen näyttää seuraavalta:





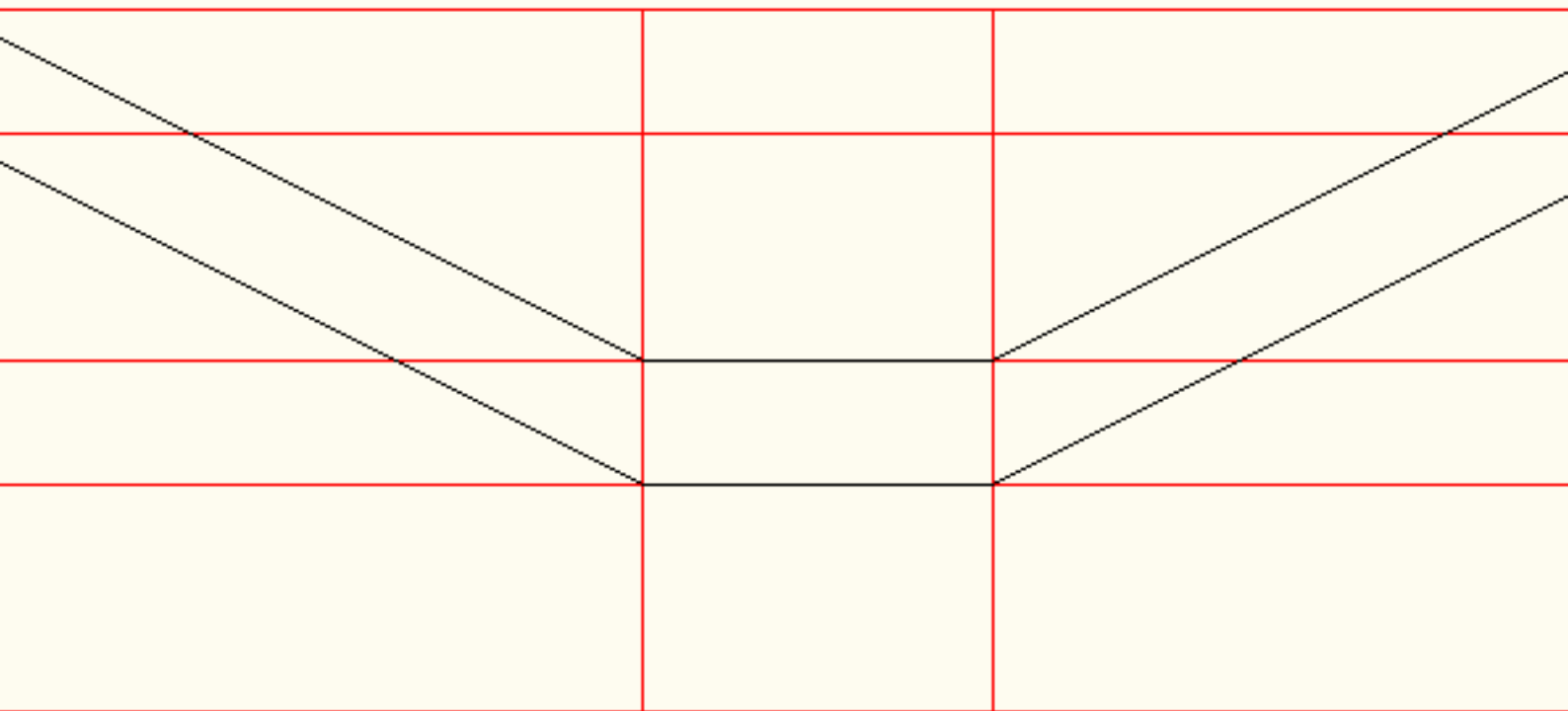
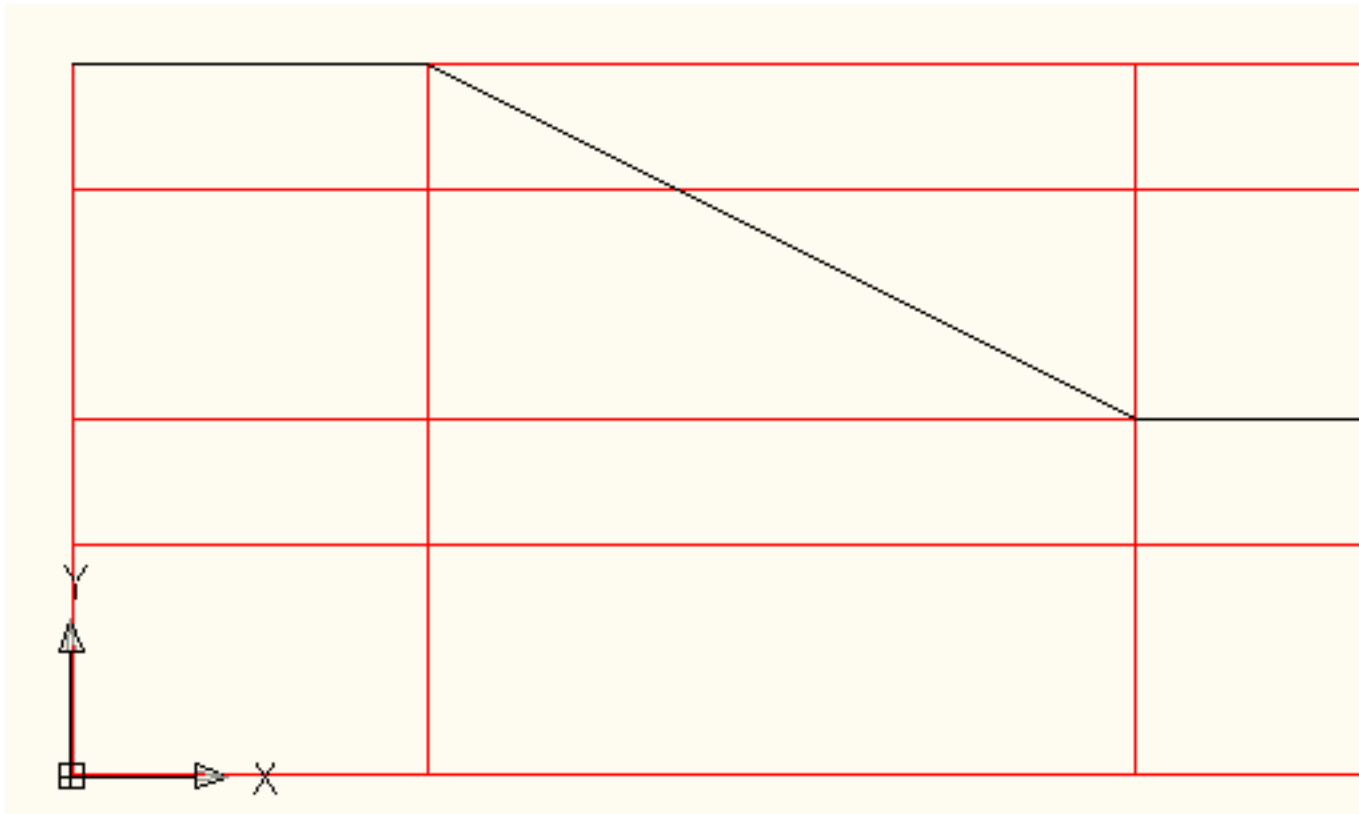
Kuva 9.9. Valmiit apuviivat

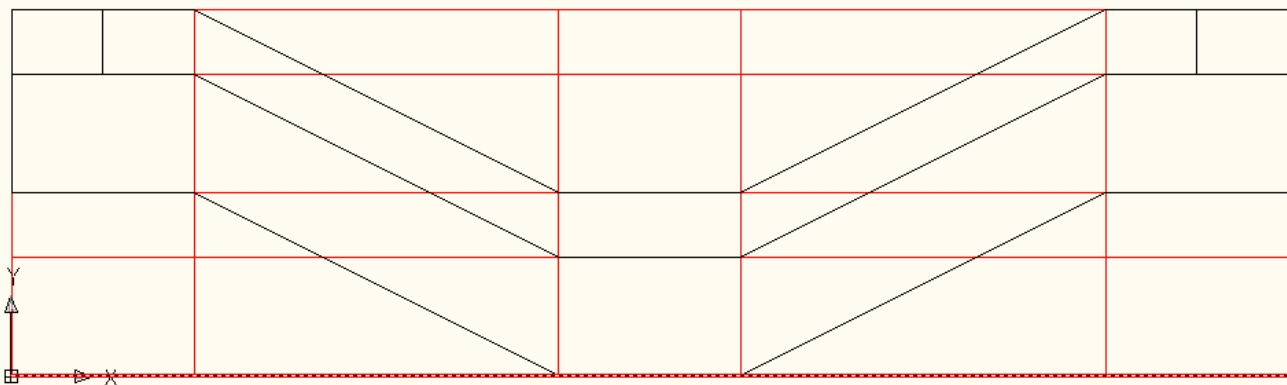
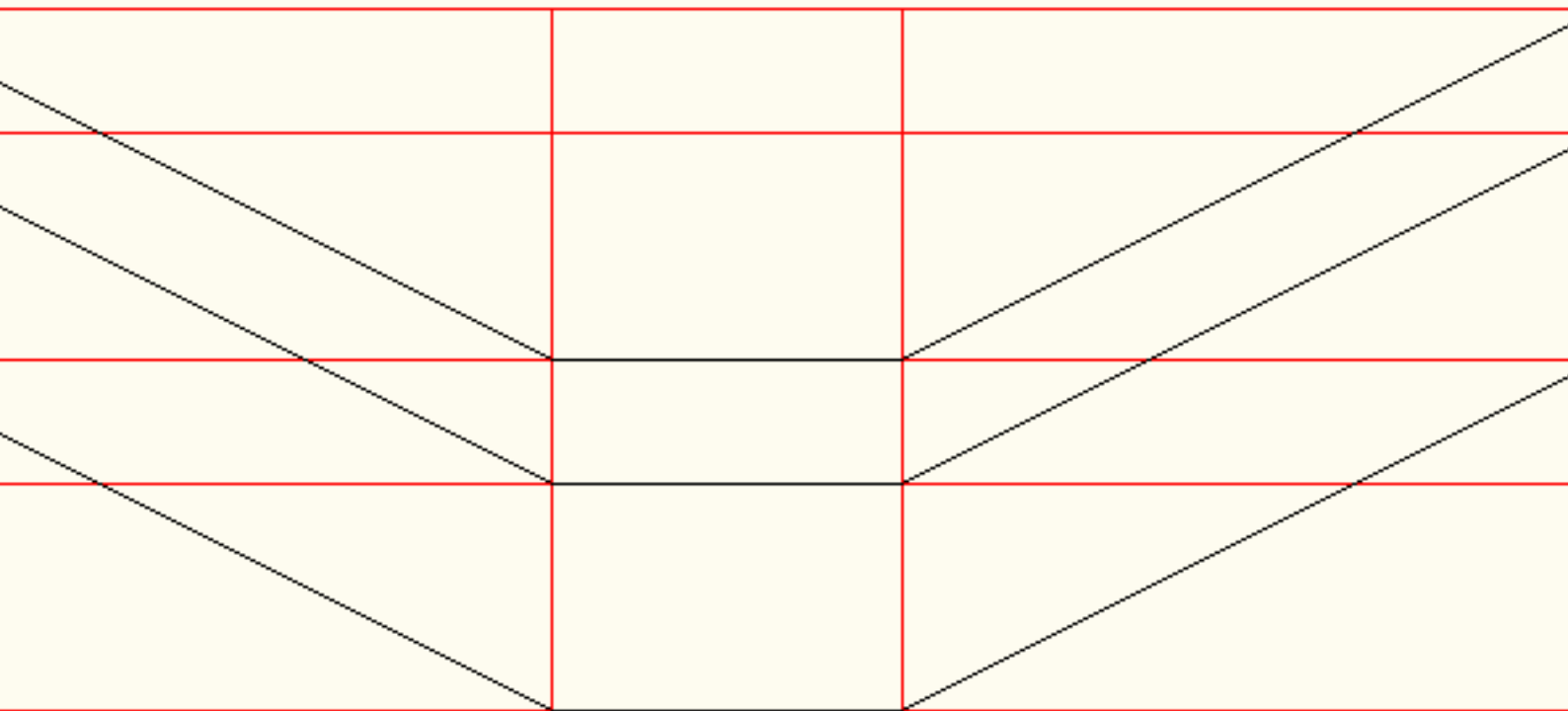
Luodaan seuraavaksi uusi taso nimeltä piirustus. Tason luonti tapahtui komennolla **la**. Anna tason nimeksi piirustus, aseta väriksi musta ja aseta se nykyiseksi tasoksi.



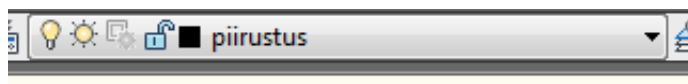
Kuva 9.10. Uuden tason luominen

Piirretään seuraavaksi varsinaisen kuvan rajat käyttäen hyödyksi apuviivoja. Kirjoitetaan komennoksi **pl** ja piirretään hiirellä osoittaen rajat seuraavien kuvien mukaan (mikäli ortho on päällä, saat sen pois päältä näppäimellä **F8**):



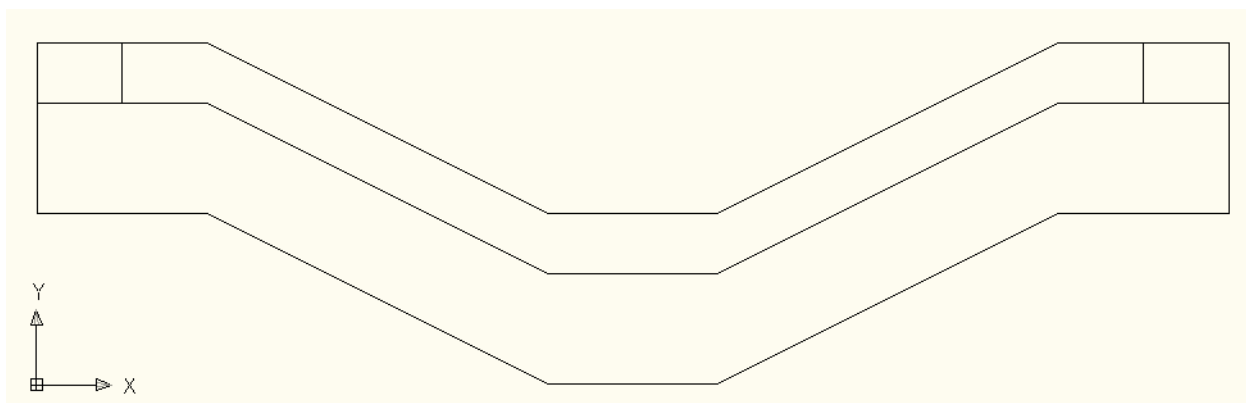


Poistetaan nyt apuviivat näkyvistä kuvan selkeyttämiseksi. Klikataan ohjelman yläreunassa olevaa tasovalikkoa, jossa pitäisi lukea tällä hetkellä käytössä olevan tason nimi, eli piirustus.

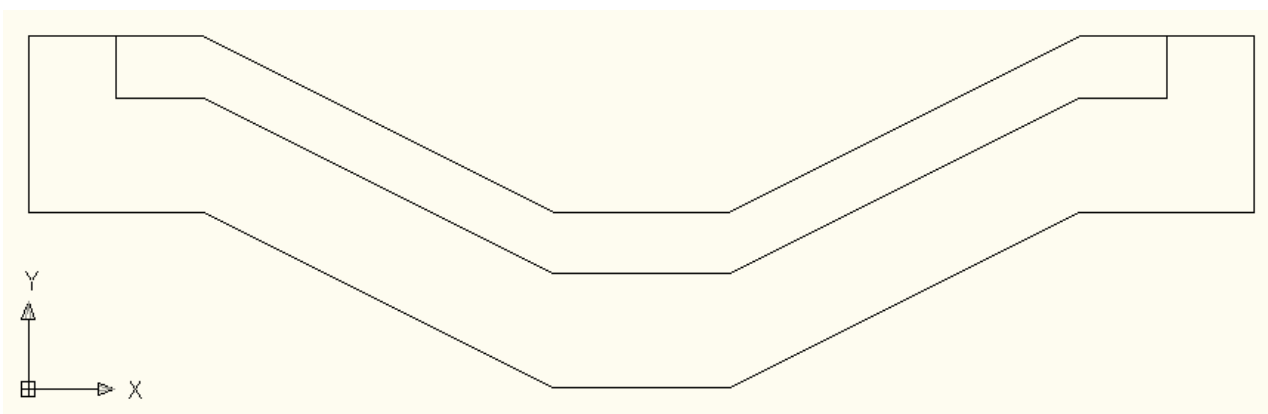


Kuva 9.11. Tasovalikko

Klikataan näkyviin ilmestyvästä tasoluettelosta apuviivat-rivin alussa olevaa lampun kuvaa ja klikataan tasoluettelo kiinni. Apuviivojen pitäisi nyt kadota näkyvistä ja kuvan näyttää seuraavalta:



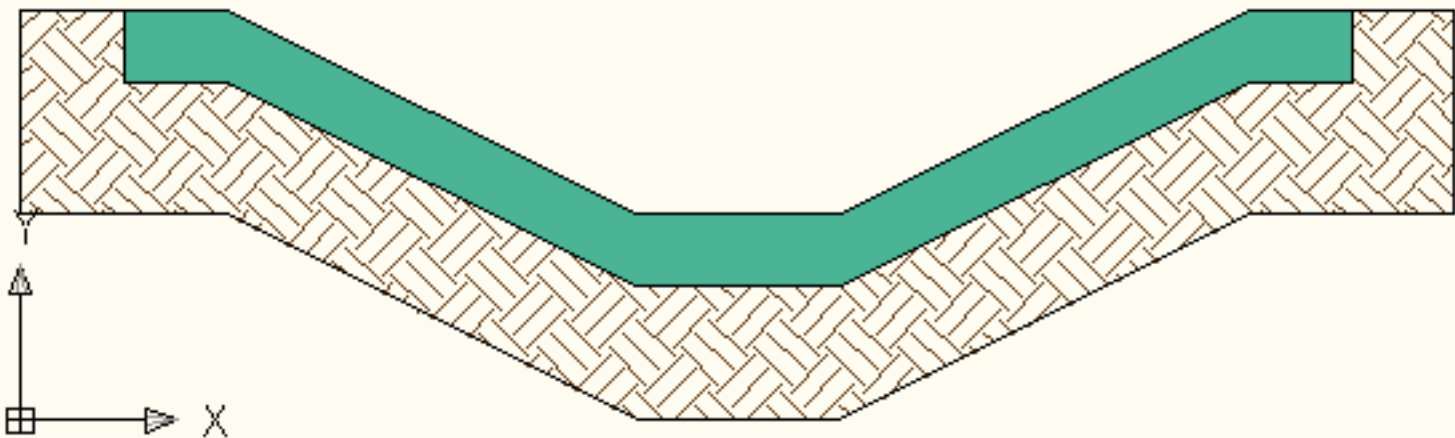
Poistetaan seuraavaksi kaksi viivaa, jotta saadaan kuva koostumaan neljän sijasta vain kahdesta alueesta ja näin väritettyä halutut alueet kerralla. Komennolla **trim** voidaan rajata tai katkaista kuva-alkio muilla kuva-alkioilla. Kirjoitetaan komennoksi **trim**, valitaan kuva-alkiot joiden välistä viiva halutaan poistaa eli kaksi ylintä vasemmanpuoleisinta pystyviivaa, painetaan **Enter**, klikataan vaakaviivaa niiden välissä ja painetaan vielä lopuksi **Enter**. Toteutetaan sama komento kuvan oikeaan ylänurkkaan ja tehdään kuvasta symmetrinen. Kuvan tuli nyt näyttää seuraavalta:



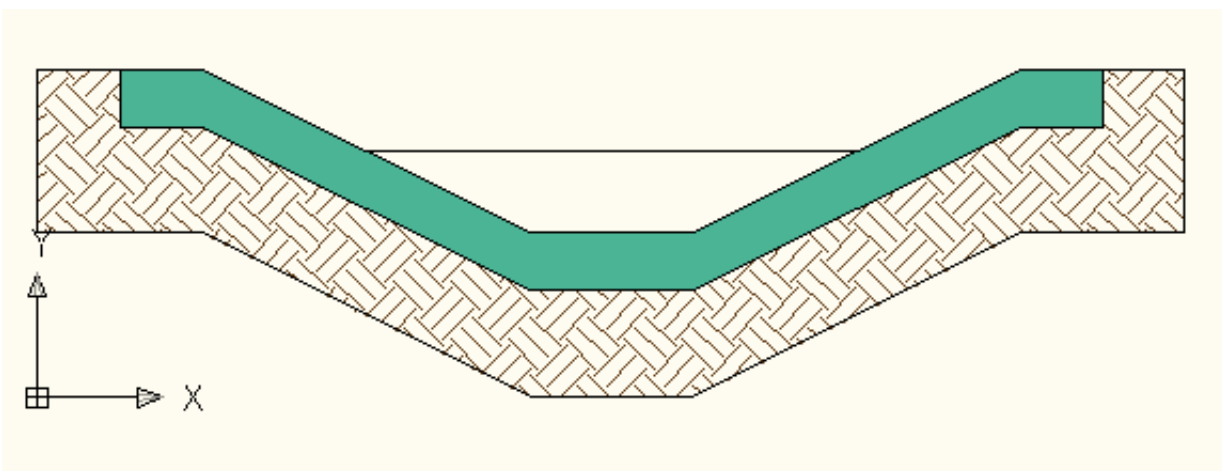
Alueiden väritys voidaan valita nopeasti esimerkiksi Tool Palettes valintaikkunasta välilehdeltä Hatches. Jos ikkuna ei ole näkyvässä, sen saa näkyviin klikkaamalla yläpalkista Tools --> Palettes --> Tool Palettes.

Valitaan Hatches välilehdeltä Imperial Hatches -otsikon alta EARTH-niminen väritys, klikataan sitä ja klikataan täytettävää aluetta, eli kuvan alemmaa aluetta.

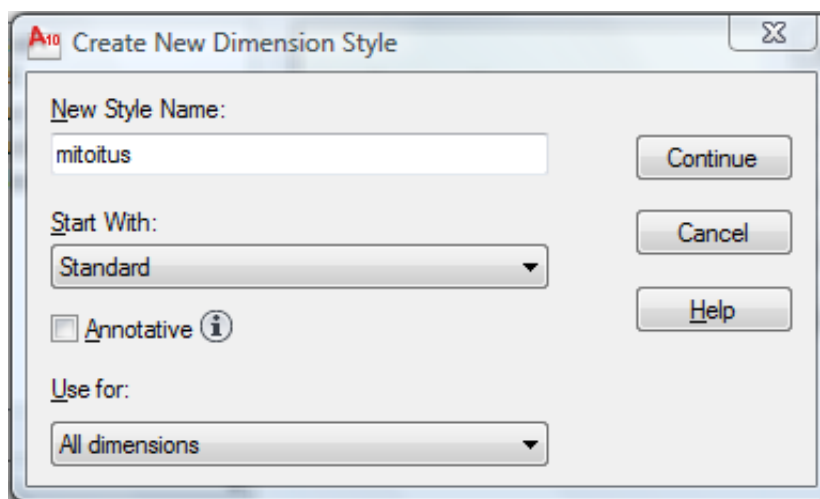
Väritetään samalla tavalla ylempi alue, mutta väritetään se vaalean siniseksi.



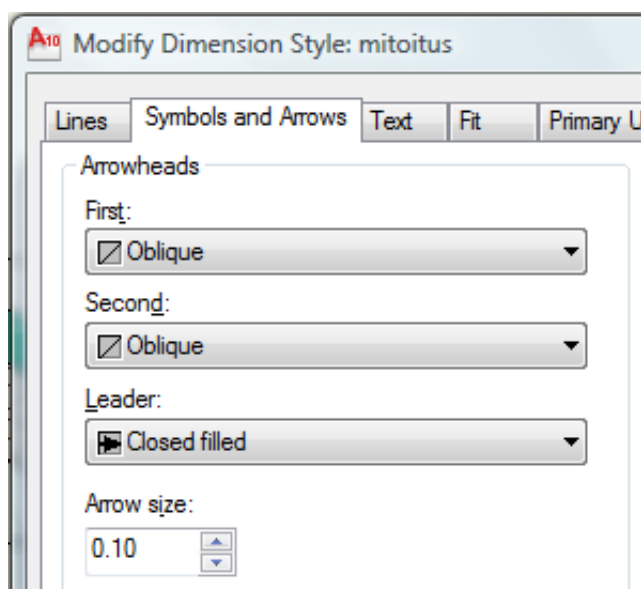
Seuraavaksi piirretään kuvaan vedenpinnantas. Piirretään se esimerkiksi **line** komennolla puoleenväliin, jolloin viiva on helppo piirtää käyttämällä aikaisemmin määriteltyä pakotusta Midpoint.



Varsinainen kuva on nyt valmis ja seuraavaksi lisätään mitoitukset ja tekstit. Ennen mitoituksen aloittamista luodaan uusi taso mitoitukseen, jonka ominaisuudet ovat samat kuin tasolla piirustus. Mitoitus aloitetaan luomalla uusi mitoitustyyli komennolla **dimstyle**. Uusi tyyli luodaan painikkeella **New**. Annetaan tyylin nimeksi mitoitus, otetaan Standard-mittatyyli malliksi ja valitaan, että tyyliä käytetään kaikkiin mitoihin. Mitoitustyylin luontia jatketaan Continue-painikkeella.



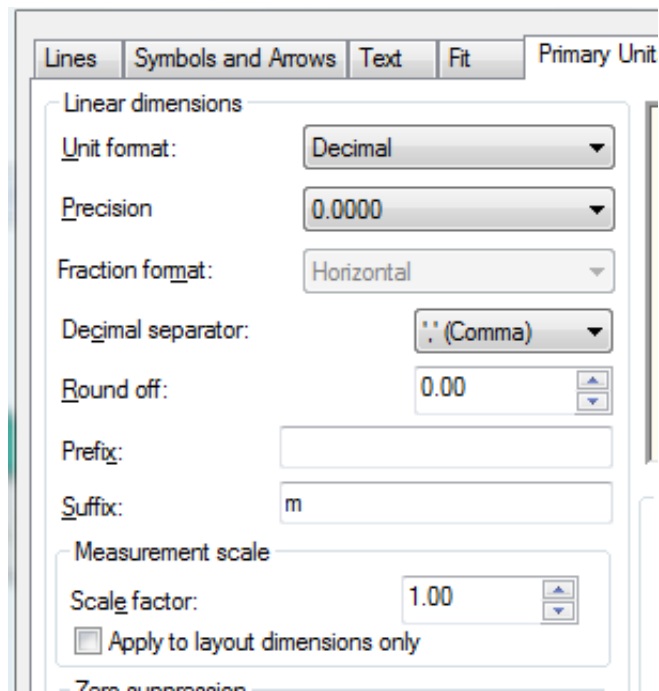
Käydään avautuvan ikkunan välilehdet järjestyksessä läpi vasemmalta oikealle. Ensimmäisellä välilehdellä määritellään mittaviivojen asetukset, niihin ei tehdä muutoksia. Toisella välilehdellä määritellään esimerkiksi mittaviivojen päiden asetuksen. Vaihdetaan ensimmäisen ja toisen alasetusvalikon määrittysten Closed filled tilalle Oblique ja asetetaan päiden kooksi 0.1. Muut asetukset jätetään ennalleen.



Text-välilehdellä määritellään mittatekstien asetukset. Asetetaan tekstin korkeudeksi 0.12, muut asetukset jätetään ennalleen.

Fit-välilehdellä määritellään, mitä mitoissa merkitään näkyviin, jos sekä tekstille että nuolille ei ole tilaa. Tärkein kohta on oikeassa alakulmassa oleva Scale for dimension features, jossa määritetään käytettävä mittakaava. Tämä työ on piirretty mittakaavaan 1:1, joten kohtaan *Use overall scale of* voidaan jättää 1.

Primary units välilehdellä määritellään mitoituksessa käytettävät yksiköt. Asetetaan desimaalimääräksi kaksi, valitaan erotukseksi pilkku ja kirjoitetaan päätteeksi m (jokaisen mittayksikön perään siis lisätään päätte m, jolloin tiedetään yksiköiden tarkoittavan metriä).

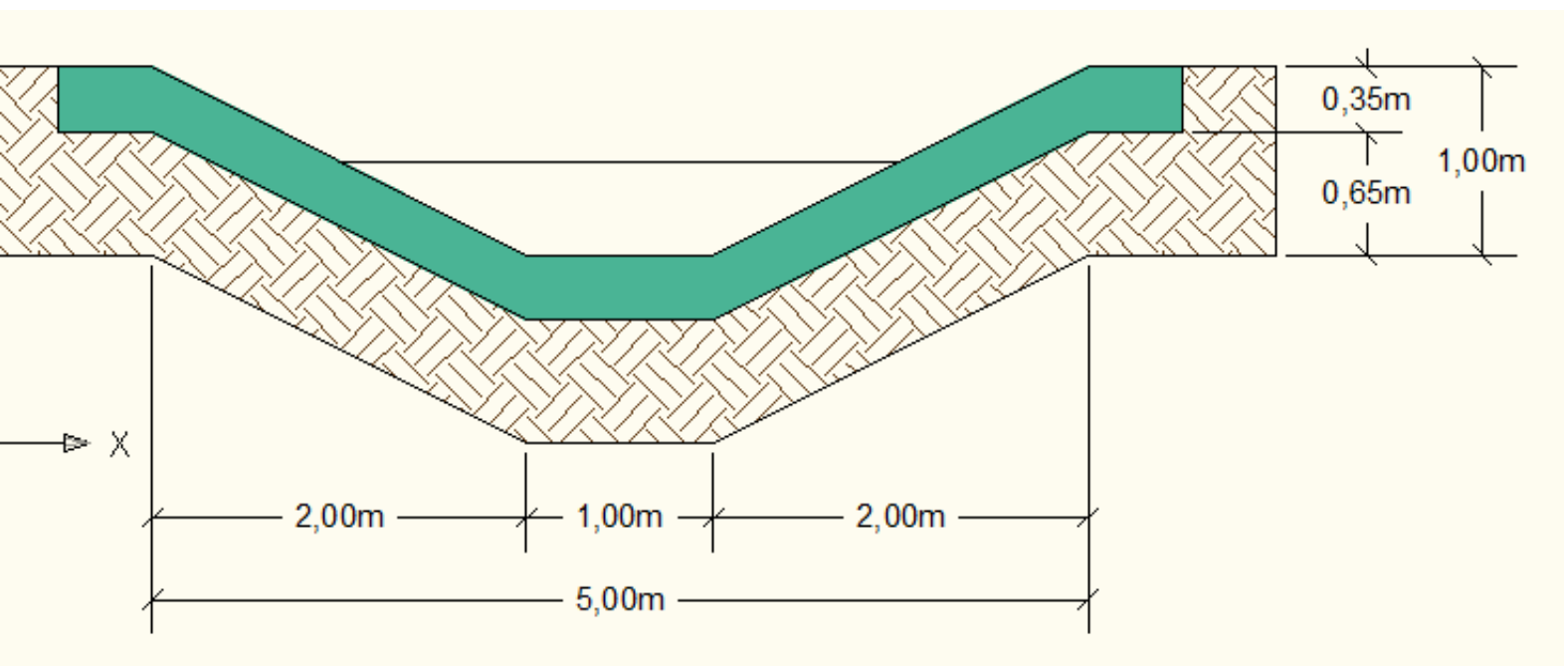


Kun mitoitustyyli on määritelty, painetaan OK, valitaan tyyli käyttöön klikkaamalla Set Current ja suljetaan ikkuna.

Mitoitusta varten valitaan **osnap** komennolla käyttöön pakotukset **Endpoint** ja **Perpendicular**. Kun halutaan mitoittaa koordinaattiakselien suuntaisia alkioita, aloitetaan mitoitus komennolla **dimlinear**. Tämän jälkeen klikataan mitoitettavan välin reunapisteitä ja asetetaan mitta haluttuun kohtaan.

Mitoitusta jatketaan **dimcontinue**-komennolla, joka jatkaa mitoitusta edellisestä mitasta haluttuun pisteeseen.

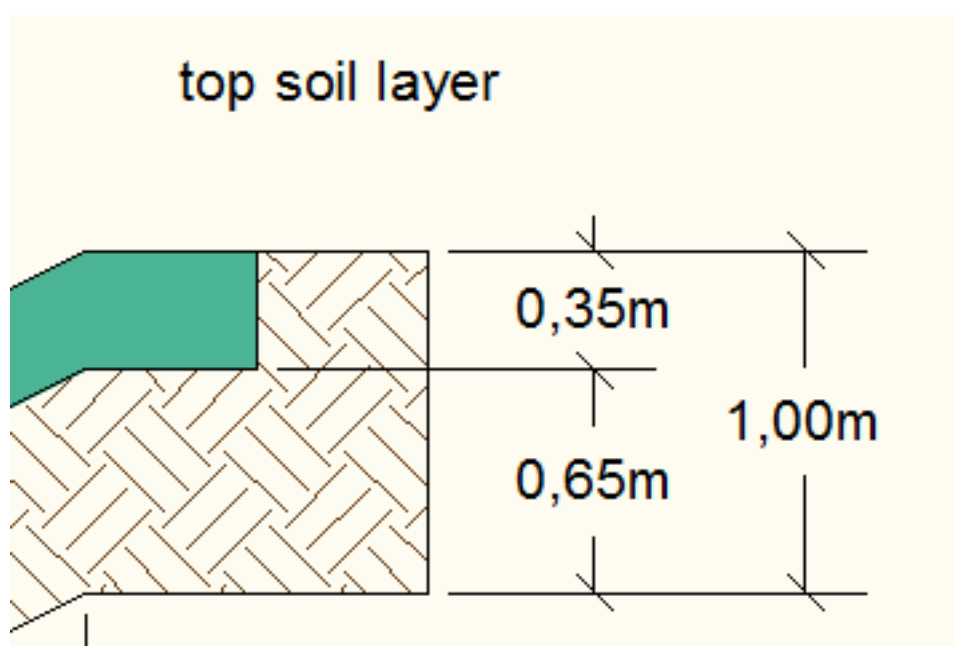
Aseta mitat kuvan osoittamalla tavalla.



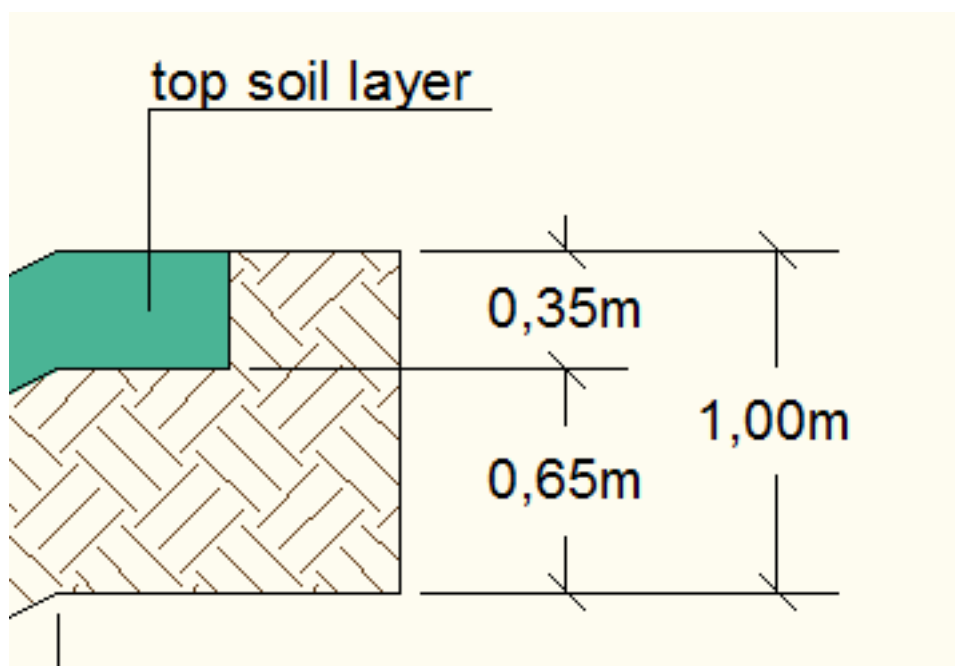
Kuva 9.12. Valmis mitoitus



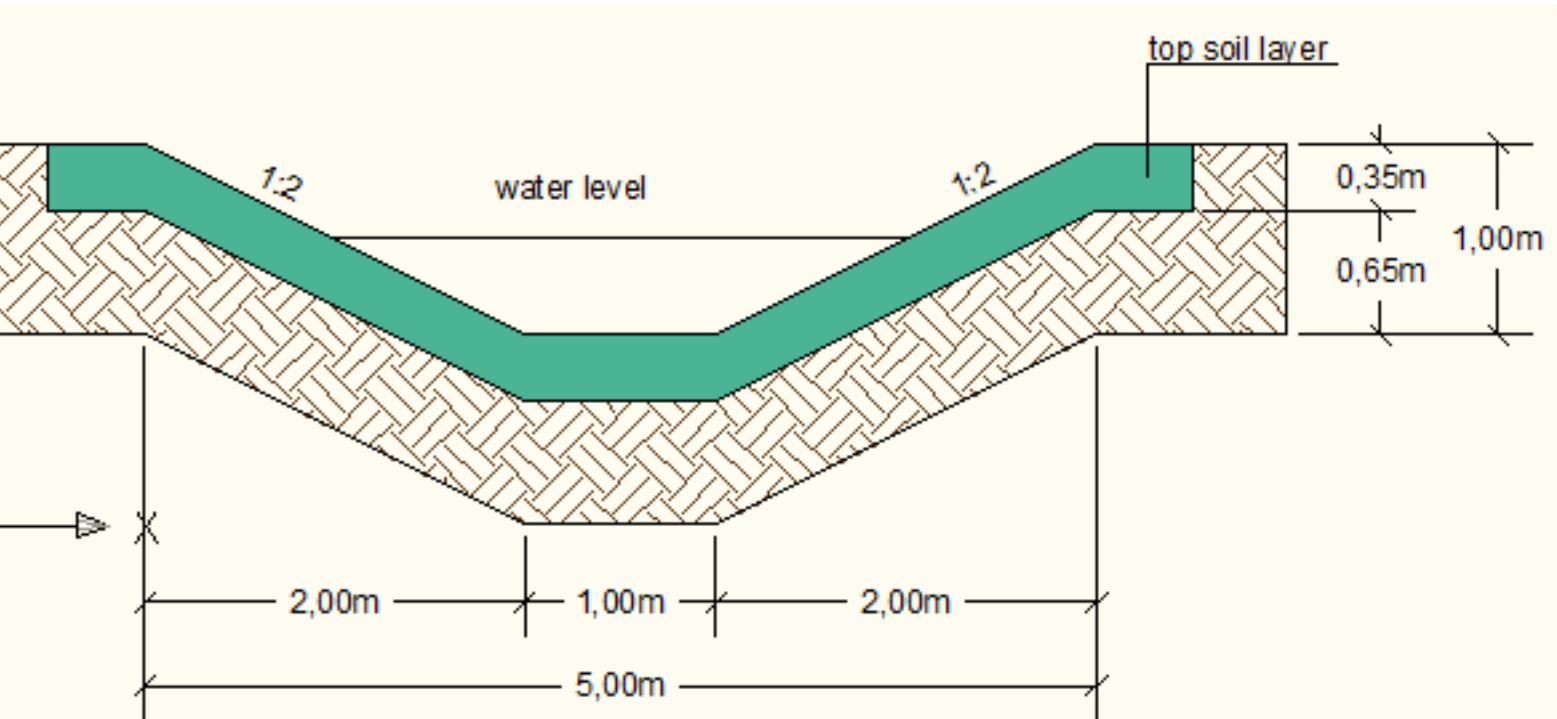
Seuraavaksi lisätään kuvaan teksti. Yksirivisten tekstien kirjoittamiseen käytetään komentoa **text**. Ensin annetaan aloituspiste, tämän jälkeen tekstin korkeus (0.12) ja viimeiseksi suuntakulma, joka on 0. Nyt voidaan kirjoittaa haluttu teksti. Lisää teksti piirroksen kuten se on seuraavassa kuvassa.



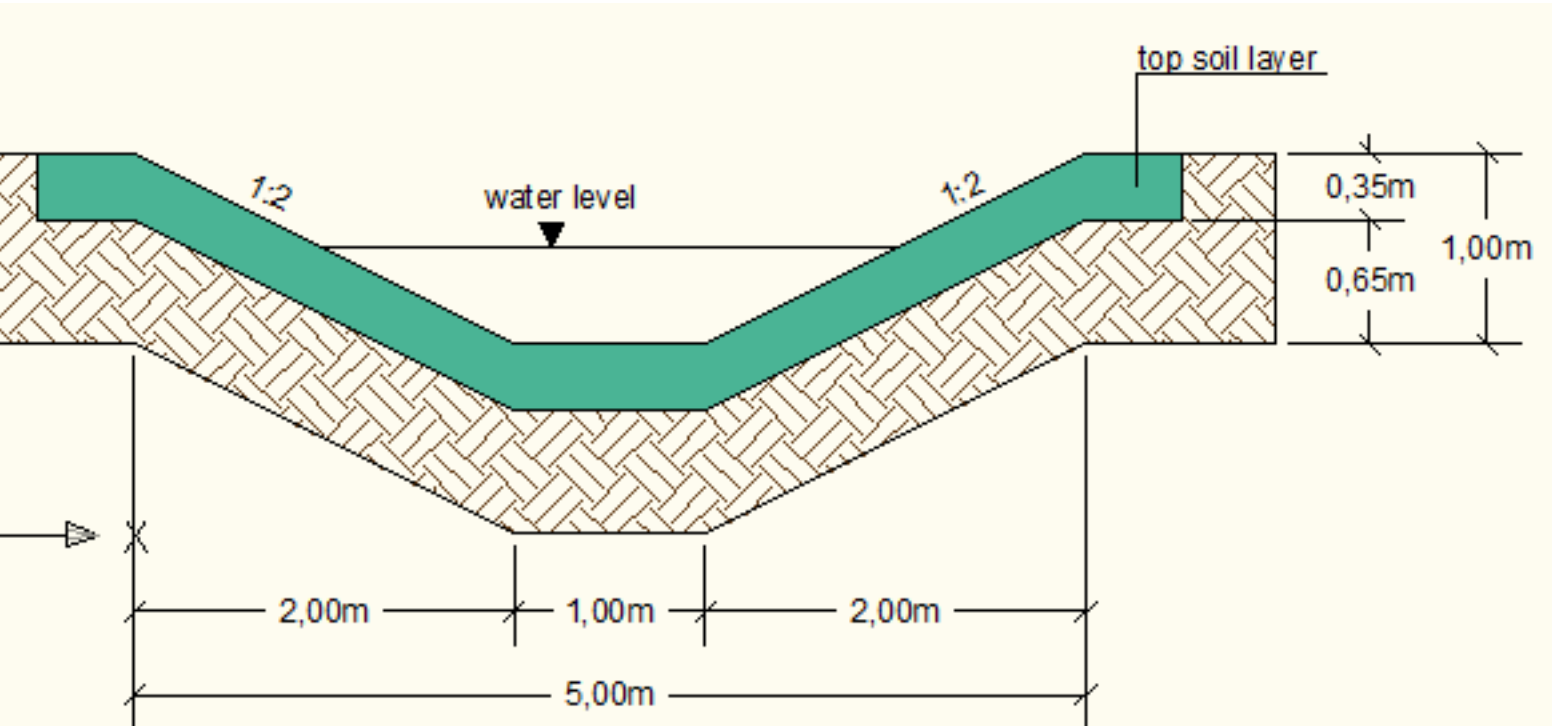
Teksti saadaan yhdistettyä kuvaan viivalla komennolla **mleader**. Osoitetaan ensin pistettä tekstin vasemmassa alakulmassa ja tämän jälkeen sinistä maakerrosta edellä osoitetun pisteen alapuolella. Seuraavaksi komento tarjoaa mahdollisuutta kirjoittaa tekstiä. Teksti on kuitenkin kirjoitettu jo, joten painetaan Esc. Kuten huomataan, on tekstiä alleviivaava viivan osa liian pitkä. Sitä voidaan muokata lyhyemmäksi klikkaamalla viivaa kerran ja vaihtamalla ilmestyvän valikon kohtaan Landing distance sopiva pituus, tässä tapauksessa esimerkiksi 1.0. Valikko voidaan sulkea raksista ja painaa tämän jälkeen Enter. Kuvan pitäisi nyt näyttää seuraavalta:



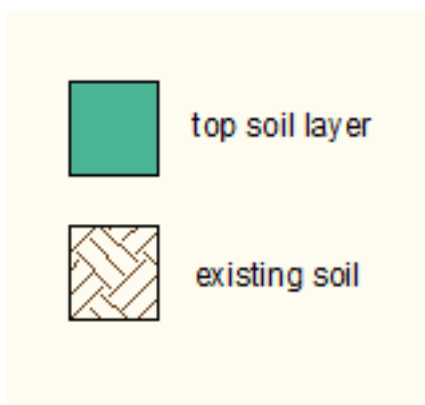
Merkitse kuvaan vielä vedenpinnan taso ja luiskien kaltevuuskulma **text**-komennolla.



Lisätään kuvaan vielä nuolenpää osoittamaan vedenpintaa. Valitaan se Tool Palettes -ikkunan välilehdeltä Structural. Valitaan Imperial Samples -otsikon alta oikealle osoittava kolmio ja klikataan se paikalleen water level -tekstin alle. Käännetään kolmio osoittamaan alaspäin klikkaamalla sitä kerran ja vaihtamalla näkyviin ilmestyvään ikkunaan kohtaan Rotation 270. Siirretään kolmio lopuksi oikealle paikalle.



Vaihdetaan tasoksi piirustus ja piirretään seuraavaksi oheisen kuvan mukaiset kuvat ja tekstit. Käytetään hyväksi **rec**-komentoa (rectangle) sekä aikaisemmin opittuja asioita. Kuvan sijainnilla tai koolla ei ole väliä.



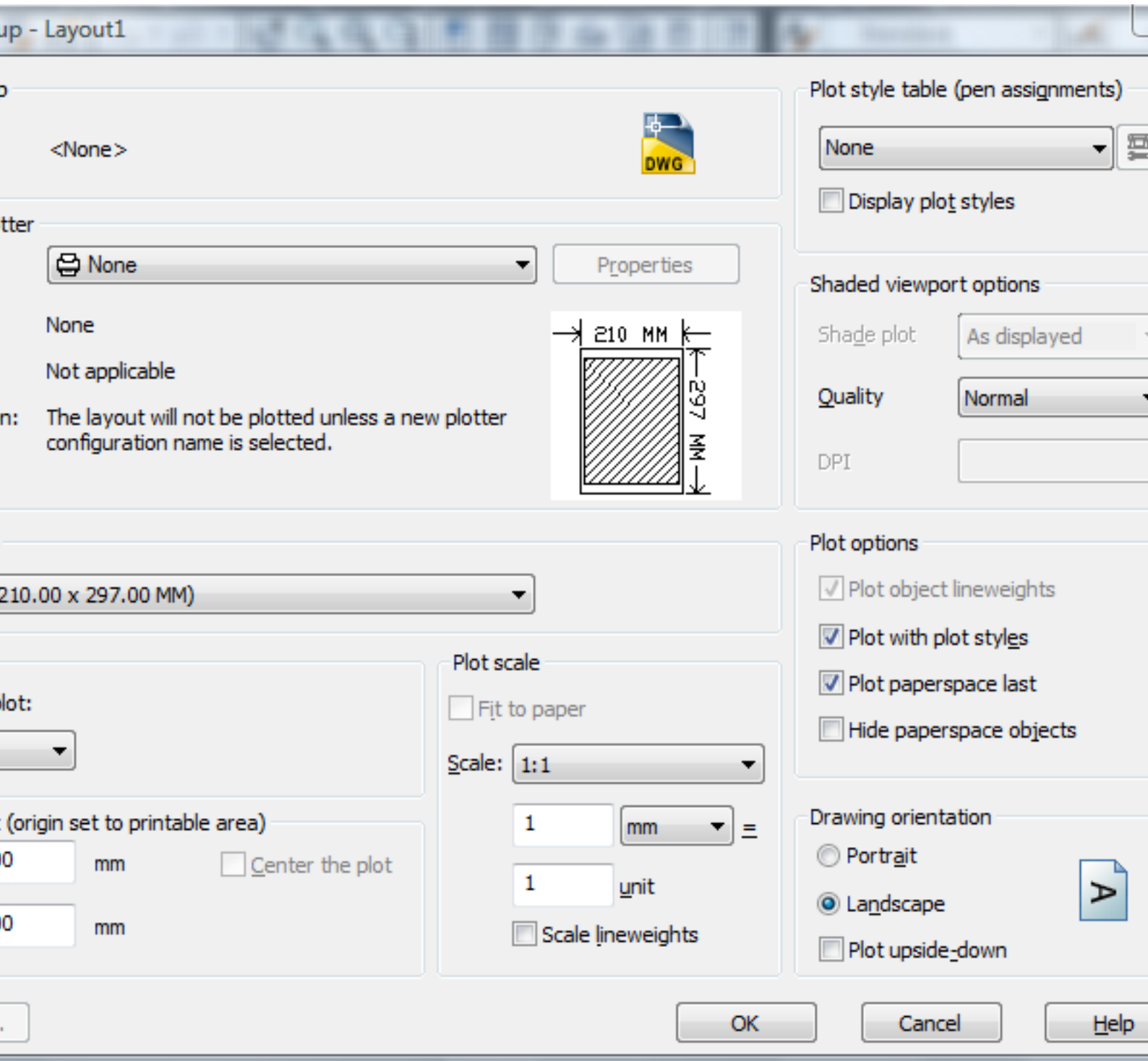
Lopuksi luodaan vielä nimiö. Käytetään komentoa **pl** ja tehdään vaakasuuntainen esimerkiksi 2\*4 metrin kokoinen neliö. Jaetaan laatikko vaakasuunnassa neljään osaan ja kaksi alinta riviä pystysuunnassa kahteen osaan. Taulukon tulisi näyttää seuraavalta:


Täytetään taulukkoon **text**-komennolla seuraavan kuvan mukaiset tekstit:

TKK Department of Civil and Environmental Engineering	
<b>CROSS SECTION SWALE</b>	
Editor Oma Nimi	Scale 1:50
Date pp.kk.vvvv	Sheet no. 1

## Luku 9. AutoCAD käyttöharjoitus

Kun kaikki piirustukseen liittyvät objektit on saatu piirrettyä, luodaan niistä tulostusasettelu. Piirustusalueella on kaksi tilaa, suunnittelutila (välilehti Model) ja tulostustila (Layout -välilehdet). Klikataan välilehteä Layout 1, jotta päästään tulostusasetteluun. Klikataan seuraavaksi hiiren oikealla Layout 1 -tekstin kohdalta ja klikataan ilmestyvästä valikosta Page Setup Manager... ja avautuvasta ikkunasta Modify. Valikosta päästään muuttamaan paperitilan asetuksia. Tässä työssä käytetään A4-kokoista vaakasuuntaista paperia ja skaala on 1:1. Klikataan lopuksi OK ja Close.



Kuva 9.13. Paperitilan asetukset

Poistetaan automaattisesti muodostettu näkymä **erase**-komennolla. Luodaan uusi taso nimeltä näkymät, josta otetaan tulostus pois päältä klikkaamalla tulostimen kuvaa.. Tällöin tasolla piirretyt objektit eivät näy tulostuksessa. Asetaan näkymät-taso tämänhetkiseksi piirtotasoksi.

Näkymät luodaan **mview**-komennolla tulostusasetteluvälilehdellä. Rajataan hiirellä vaakasuuntainen suorakulmio paperin yläosaan varsinaista kuvaa varten. Tuplaklikkaamalla näkymän sisään saadaan oikea kohta suunnittelutilan piirustuksesta näkyviin zoomaamalla ja liikuttelemalla. Kuva voidaan asettaa mittakaavaan komennolla **zoom** ja antamalla tämän jälkeen mittakaava.

$$\text{scale factor } XP = \frac{[\text{mm}] \text{ in Layout}}{[\text{draw units}] \text{ in Model space}}$$

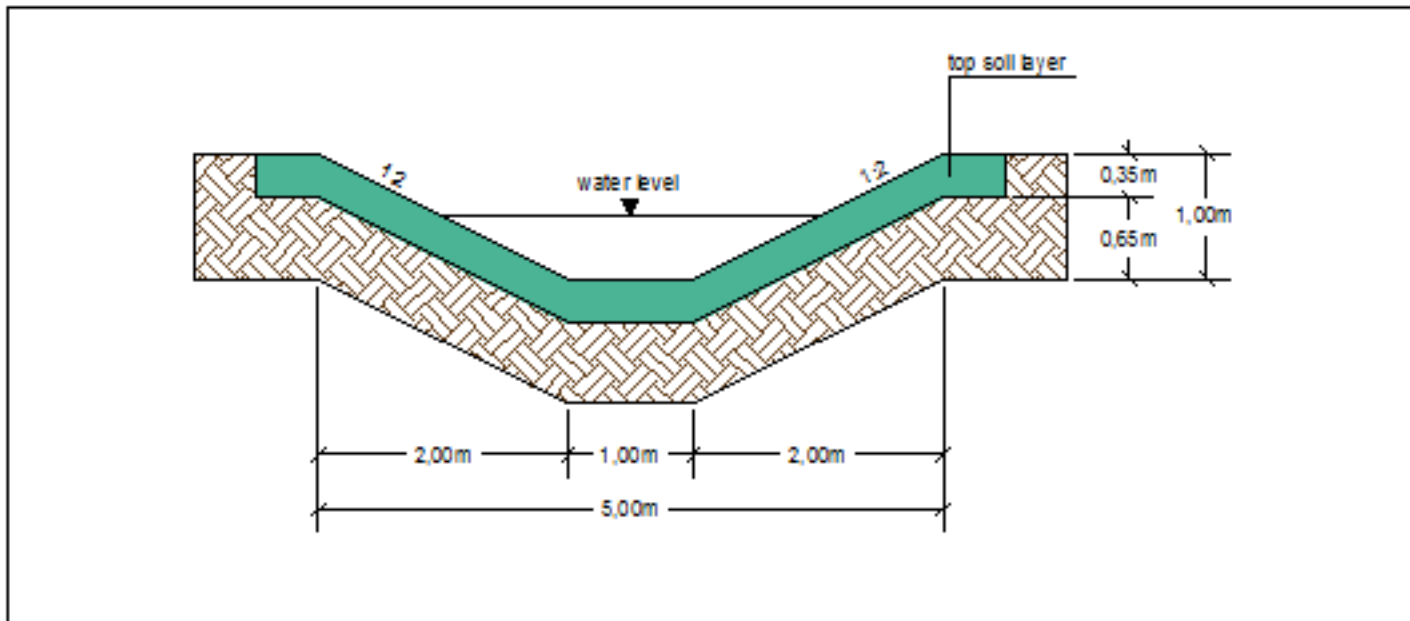
(How many mm on the paper should equal 1 draw unit)

draw unit [U] in model space	desired scale	print out	[mm] in Layout for 1 draw unit	XP factor
1 U = 1mm	1: <u>1</u>	in real size	1mm/1 = 1 mm	1 mm/ <u>1</u> = 1XP
1 U = 1mm	1: <u>100</u>	100 times smaller	1mm/100 = 0.01 mm	1 mm/ <u>100</u> = 0.01XP
1 U = 1mm	1: <u>50</u>	50 times smaller	1mm/50 = 0.02 mm	1 mm/ <u>50</u> = 0.02XP
1 U = 1cm (=10 mm)	1: <u>100</u>	100 times smaller	10mm/100 = 0.1 mm	10 mm/ <u>100</u> = 0.1XP
1 U = 1m (=1000 mm)	1: <u>100</u>	100 times smaller	1000mm/100 = 10 mm	1000 mm/ <u>100</u> = 10XP
1 U = 1m (=1000 mm)	1: <u>50</u>	50 times smaller	1000mm/50 = 20 mm	1000 mm/ <u>50</u> = 20XP

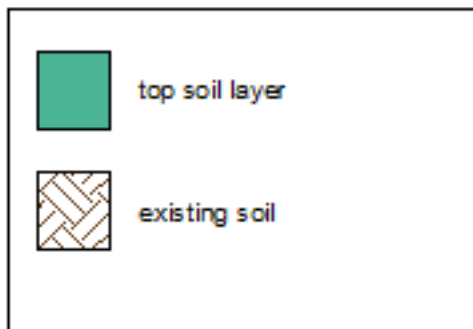
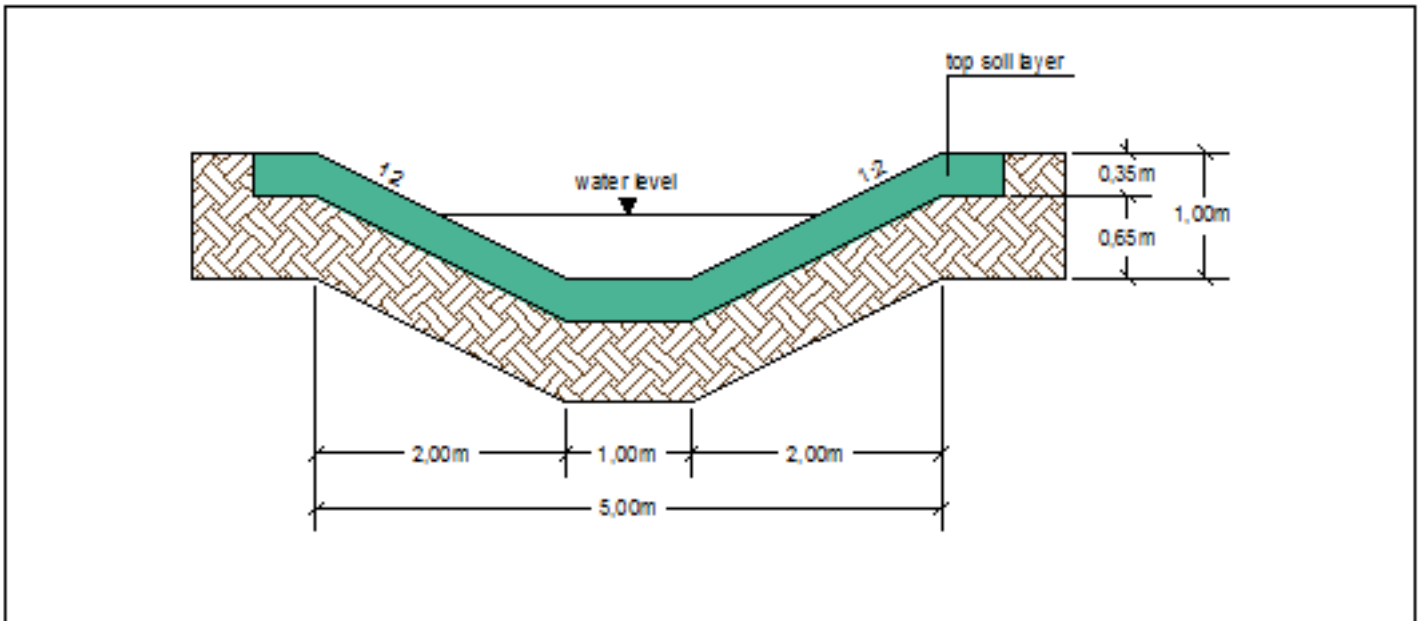
Kuva 9.14. Mittakaavan määrittäminen

Piirrettäessä kuvaa on yksikkönä käytetty metriä ja kuva halutaan mittakaavaan 1:50. Mittakaavan kerroin saadaan siis alimmalta riviltä, eli kerroin on 20. Kertoimen perään kirjoitetaan xp, mikäli kuvaa halutaan zoomata verraten todellisiin mittoihin ja x, mikäli kuvaa halutaan zoomata verraten senhetkiseen näkymään. Kirjoitetaan siis komennon zoom jälkeen **20xp**. Kun mittakaava on asetettu, ei kuvaa saa enää zoomata, sillä tällöin mittakaava muuttuu.

Näkymän muokkauksesta päästään pois tuplaklikkaamalla hiirellä näkymän ulkopuolella. Näkymän kokoa saadaan muokattua klikkaamalla kerran näkymän rajoja ja vetämällä ilmestyvistä nurkkapisteistä. Tulostusasettelun tulisi nyt näyttää seuraavalta:

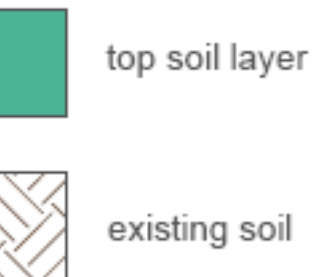
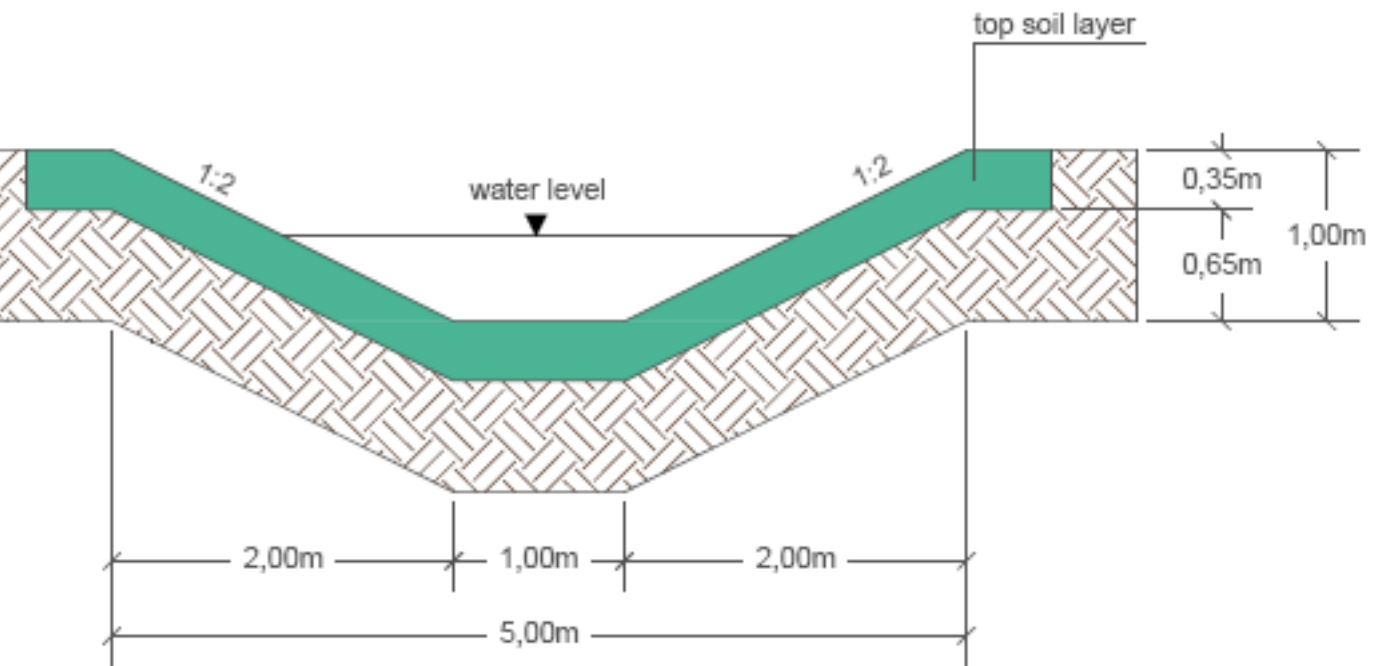


Lisätään aseteluun vielä kaksi näkymää, toinen vasemmalle alas maakerroksille ja toinen oikealle alas nimiölle. Nämä kuvat asetetaan halutun kokoisiksi zoomaamalla, mittakaavalla ei ole väliä. Kuvan tulisi näyttää valmiina suurinpiirtein seuraavalta:



TKK Department of Civil and Environmental Engineering	
<b>CROSS SECTION SWALE</b>	
Editor Oma Nimi	Scale 1
Date pp.kk.vvvv	Sheet no.

Valmis asettelu voidaan tulostaa PDF-tiedostoksi valikosta File - Plot. Printteriksi valitaan DWG To PDF.pc3, paperin kooksi valitaan A4 ja skaalaksi 1:1. Lopuksi klikataan OK.



TKK Department of Civil and Environmental Engineering		
<b>CROSS SECTION SWALE</b>		
Editor	Oma Nimi	Scale
Date	pp.kk.vvvv	Sheet no.

Kuva 9.15. Valmis kuva

### 9.3. Itsenäisesti tehtävä harjoitus

Harjoituksen tavoitteena on palauttaa mieleen ohjatussa harjoituksessa opittuja asioita. Tehtävänäsi on piirtää AutoCAD:llä jokin yhdyskunta- ja ympäristötekniikkaan liittyvä elementti tai kohde. Piirustuksen mittojen ei tarvitse olla täysin todenmukaisia, mutta realismi on suotavaa. Pääasia kuitenkin



on, että työstä käy ilmi taitosi toteuttaa ohjelmalla kuva ja saattaa se pdf-muotoon. Työn ei tarvitse olla monimutkainen tai laaja.

Vaatimukset tehtävän suorittamiseksi:

- Piirustuksen tulee kuvata jotakin todellista
- Piirustuksessa tulee olla käytettynä vähintään kahta tasoa
- Piirustuksessa tulee olla vähintään yksi mitoitettu objekti
- Piirustuksessa tulee olla tekstiä
- Piirustuksesta on tehty tulostusasettelu ja se on saatettu pdf-muotoon

Ohjatusta ja itsenäisestä harjoituksesta palautetaan yksi raportti, joka tehdään valmiin raporttipohjan mukaisesti. Ohjatusta harjoituksesta riittää, että mainitset siitä alun johdannossa ja liität tuottamasi kuvan raporttiin (usean pdf-tiedoston yhdistämisestä annetaan erilliset ohjeet). Itsenäisesti tehtävässä harjoituksessa tuottamasi kuva tulee myös liittää raporttiin ja lisäksi raportissa tulisi vastata itsenäisesti tehtävän harjoituksen osalta mm. seuraaviin kysymyksiin:

- Mitä piirtämäsi kuva esittää?
- Miten työsi eteni? Missä järjestyksessä teit kuvassa näkyvät asiat?
- Oliko harjoitus helppo vai vaikea toteuttaa?
- Tukiko harjoitus AutoCADin käytön oppimista?
- Tukiko ohjattu harjoitus riittävästi itsenäistä harjoitusta?

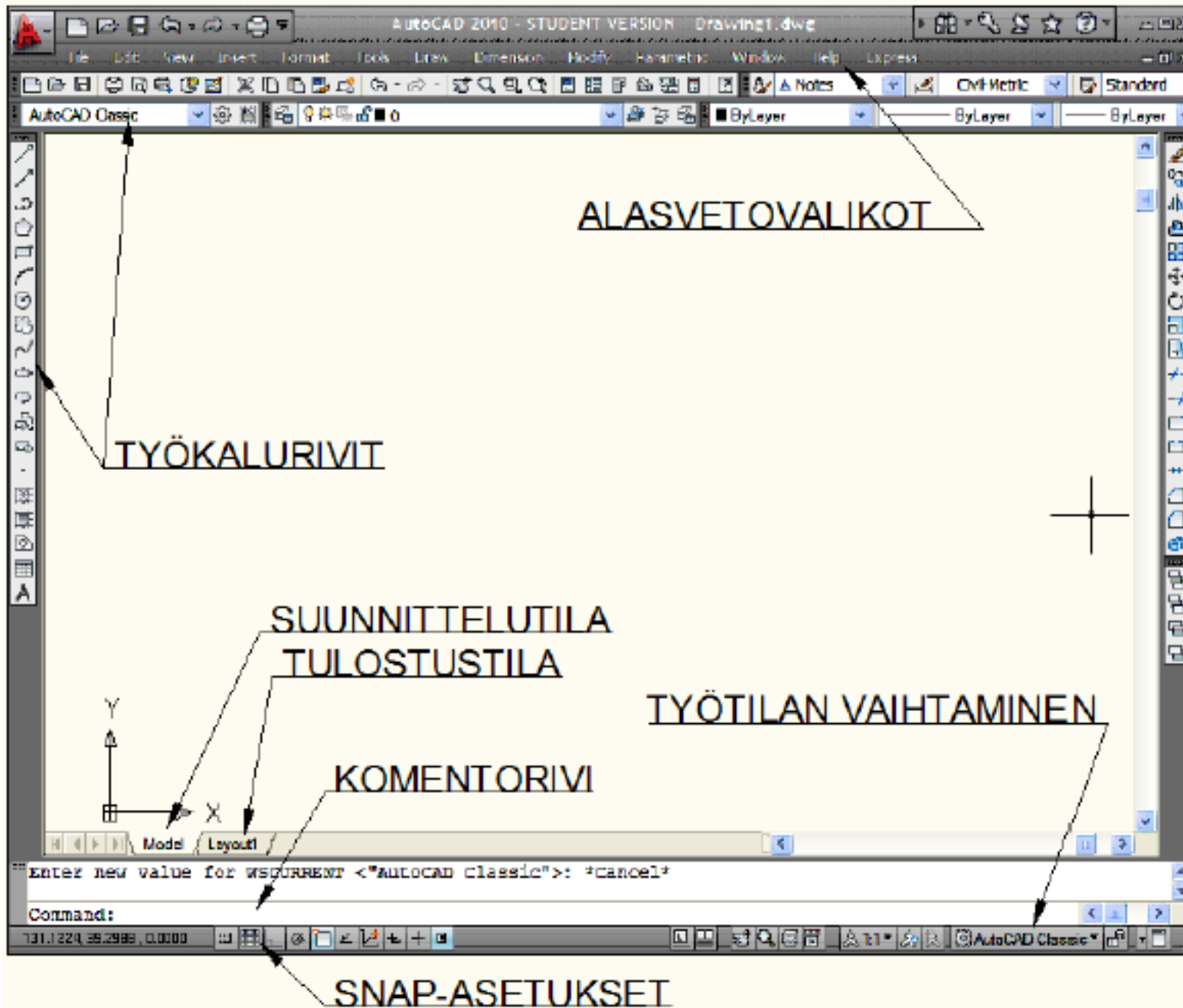
Raportin tekstin tulee olla yhtenäinen teksti. Oheisten kysymysten lisäksi voit halutessasi liittää raporttiin myös muita pohdintoja, tarkastelua ja huomioita työhön ja työn tekemiseen liittyen.



# AutoCAD pikaopas

Tässä oppaassa on käyty lyhyesti läpi AutoCADin käytön perusteita. Opas on tehty R-CAD-kurssin AutoCAD-käyttöharjoituksen pohjalta.

## 10.1. Käyttöliittymä



Kuva 10.1. Käyttöliittymän ominaisuudet

Valikko	Sisältö
File	tiedostojen hallinta
Edit	muokkaustoiminnot
View	näytön näkymien hallinta
Insert	blokkien, viitepiirustusten ja objektien käyttö
Format	piirustusalkioiden asetukset
Tools	piirtämisen aputoiminnot
Draw	piirto- ja mallinnuskäskyt
Dimension	mitoitus
Modify	geometrian muokkaus
Window	näkymäikkunoiden hallinta ja järjestely
Help	ohjeet

Kuva 10.2. Alasvetovalikot

## 10.2. Komentojen antaminen ja peruminen

Komentoriviä käytetään komentojen antamiseen. Komennot annetaan joko kokonaisina tai lyhenteinä eli aliaksina. Komentojen käyttö nopeuttaa ohjelman käyttämistä ja komentoriviltä voidaan myös seurata mitä seuraavaksi tulisi tehdä.

Komennot voidaan antaa joko näppäimistöltä, alasvetovalikoista tai työkalupainikkeilla. Komento voidaan perua komennolla **undo** tai **u** tai näppäinyhdistelmällä **Ctrl+z** ja viimeisin peruttu komento voidaan palauttaa komennolla **redo**. Meneillään oleva komento keskeytetään **Esc**-näppäimellä ja komennot hyväksytään painamalla **Enteriä**, **space**- eli välilyöntinäppäintä tai **hiiren oikeanpuoleista painiketta**.

Funktionäppäimiä kannattaa käyttää erityisesti snap-asetusten hallintaan työskentelyn nopeuttamiseksi. Alla olevassa taulukossa on esitetty funktionäppäinten toiminta.

<b>F1</b>	Aktivoi AutoCADin help-toiminnot
<b>F2</b>	Tekstinäyttö päälle ja pois
<b>F3</b>	Object Snap päälle ja pois
<b>F6</b>	Koordinaattinäytön esitystavan muutos
<b>F7</b>	Grid päälle ja pois
<b>F8</b>	Ortho päälle ja pois
<b>F9</b>	Snap päälle ja pois
<b>F10</b>	Polar Tracking päälle ja pois

Kuva 10.3. Funktionäppäinten toiminta

Hyödyllisiä funktionäppäimiä ovat etenkin F3, jolla voidaan määrittää pakotukset ja F8, jolla saadaan käyttöön kohtisuorat suunnat.

### 10.3. Zoomaus ja panorointi

Zoomaus on helpointa suorittaa pyörittämällä hiiren rullaa, mutta siihen voidaan käyttää myös komentoa **zoom** tai **z**.

**ZOOMAUS-VALINNAT****Zoom Window**Command: `zoom (Optio:Window)`

Zoom Window -komennolla voidaan osoittaa kahden murkkapisteen avulla lähemmin tarkasteltava alue. Zoom-komennon oletusoptio.

**Zoom Previous**Command: `zoom (Optio:Previous)`

Zoom Previous -komento palauttaa edellisen näkymän. Palautettua näkymää ei voi enää palauttaa takaisin päin.

**Zoom Dynamic**Command: `zoom (Optio:Dynamic)`

Zoom Dynamic -komennolla voidaan määrittellä dynaamisesti zoomausikkunan koko ja paikka. Zoomausikkuna on näkymäikkunan muotoinen.

**Zoom Scale**Command: `zoom (Optio:Scale)`

Zoom Scale -komennolla näkymää voidaan suurentaa tai pienentää kerrointen avulla. Zoomauskerrointen avulla voidaan viitata:

- piirustuksen rajoihin: syötetään pelkkä zoomauskerroin
- nykyiseen näkymään: syötetään zoomauskerroimen perään x
- tulostustilan (PaperSpace) näkymäikkunan mittakaavaan: haluttu mittakaava n:m syötetään muodossa n/mxp (esim. mittakaava 1:5 muodossa 1/5xp)

**Zoom In, Zoom Out**Command: `zoom (Optio automaattisesti: scale)`

Zoom In ja Zoom Out ovat Zoom Scale -komennon erikoistapauksia. Zoom In suurentaa näkymän kaksinkertaiseksi (2x) nykyisestä näkymästä ja Zoom Out pienentää näkymää kertoimella 0.5x.

**Zoom Center**Command: `zoom (Optio:Center)`

Zoom Center zoomaa hiirellä osoitetun keskipisteen suhteen määritellyllä zoomauskerroimella. Zoomauskerroin voidaan antaa kuin Zoom Scale -komennossakin, esim. 2x, tai haluttuna näkymäikkunan korkeutena käytössä olevilla piirtoyksiköillä, esim. 30. Oletusarvona on nykyisen näkymäikkunan korkeus piirtoyksiköissä.

**Zoom All**Command: `zoom (Optio:All)`

Zoom All zoomaa näkymäikkunaan piirustuksen kaikki kuva-alkiot tai piirustuksen rajojen määrittämän alueen. Jos kuva-alkiot mahtuvat piirustusrajojen sisään, näkymä sovitetaan rajojen mukaan. Jos rajojen ulkopuolelle jää kuva-alkioita, näkymä sovitetaan kauimmaisten alkioiden mukaan ( vastaa Zoom Extents -komentoa).

**Zoom Extents**Command: `zoom (Optio:Extents)`

Zoom Extents toimii kuten Zoom All, mutta ei huomioi piirustuksen rajoja.

Kuva 10.4. Zoomausvalinnat

Panorointi tarkoittaa näkymäikkunan liikuttelua. Rullahiirellä panorointi onnistuu painamalla rulla alas ja liikuttamalla hiirtä. Panorointi voidaan suorittaa myös komennolla **-pan**.

## 10.4. Tasojen hallinta

Tasoja päästään määrittelemään ja muokkaamaan komennolla **layer** tai **la**. Komento avaa Layer properties manager -ikkunan.

Uusi taso luodaan **New-painikkeella** tai näppäinyhdistelmällä **Alt+N**.

Tason nimi muutetaan kaksoisklikkaamalla sitä hiirellä ja muita ominaisuuksia muokataan klikkaamalla ominaisuutta kerran.

Tasojen tuhoaminen tapahtuu valitsemalla tuhottavat tasot ja painamalla Delete.

















## 10.5. Snap-asetukset

Asetusten käyttäminen on hyödyllistä kun uuden kuva-alkion piirtäminen aloitetaan olemassa olevasta kuva-alkiosta. Asetusten avulla uusi viiva saadaan alkamaan esimerkiksi olemassa olevan viivan päästä tai keskeltä.

Tartuntapisteitä eli pakotuksia päästään muokkaamaan komennolla **osnap** tai **os**. Tartuntapisteet määritellään avautuvan Drafting Settings -ikkunan Object Snap -välilehdeltä ja vain tarvittavat pakotukset kannattaa pitää käytössä.

Pakotukset saadaan käyttöön ja pois käytöstä funktionäppäimellä F3 tai komentorivin alla olevan tilarivin OSNAP-painikkeella.

Tartuntapisteet on esitetty alla olevassa kuvassa.


	<b>Loppupiste, Endpoint, end</b> Endpoint pakottaa pisteosoituksen kuva-alkion, kuten viivan tai kaaren päätepisteeseen.
	<b>Keskipiste, Midpoint, mid</b> Midpoint pakottaa pisteosoituksen kuva-alkion, kuten viivan, käyrän, kaaren tai 3D-pinnan särmän,
	<b>Leikkauspiste, Intersection, int</b> Intersection pakottaa pisteosoituksen kuva-alkioiden leikkauspisteeseen. Viivojen on leikattava samassa kuvatasossa. Jos viivoilla on eri z-koordinaatit, leikkauspistettä ei löydy, vaikka xy-tasossa viivat näyttäisivät leikkaavan.
	<b>Oletettu leikkauspiste, Apparent Intersection</b> Apparent Intersection toimii xy-tasossa kuten Intersection, mutta hakee leikkauspisteitä tarvittaessa myös 3D-koordinaatistosta.
	<b>Ympyrän, kaaren tai ellipsin keskipiste, Center, cen</b> Center pakottaa pisteosoituksen ympyrän, kaaren tai ellipsin keskipisteeseen. Osoituspisteenä on kaaren, ympyrän tai ellipsin kehä. Osoitus toimii myös 3D-kappaleilla, kuten sylinterillä ja pallolla, tai kappaleen pyöristyksellä.
	<b>Neljännespiste, Quadrant, qua</b> Quadrant pakottaa pisteosoituksen ympyrän, kaaren tai ellipsin neljännespisteisiin. Osoituspisteenä on kaaren, ympyrän tai ellipsin kehä. Neljännespisteiden paikat määräytyvät käytetyn koordinaatiston mukaan. Maailmankoordinaatistossa (WCS) neljännespisteet ovat 0, 90, 190 ja 270 astetta keskipisteestä.
	<b>Kohtisuora, Perpendicular, per</b> Perpendicular pakottaa pisteosoituksen kuva-alkiota, kuten viivaa, ympyrää, kaarta ja ellipsiä, vastaan kohtisuoraan. Kohtisuoruus voidaan määrittää myös kuva-alkion kuvitellulle jatkeelle.
	<b>Yhdensuuntainen, Parallel, par</b> Parallel-pakotuksen avulla voidaan piirtää olemassa olevan kuva-alkion, kuten viivan, kanssa yhdensuuntainen.
	<b>Tangentti, Tangent, tan</b> Tangent pakottaa pisteosoituksen ympyrän, kaaren tai ellipsin tangenttipisteeseen.
	<b>Pistealkio, Node, nod</b> Node pakottaa pisteosoituksen Point-, Divide- tai Measure-käskyillä tuotettuihin pistealkioihin. Pistealkioita voidaan blokkeihin mukaan.
	<b>Lisäyspiste, Insertion, ins</b> Insertion pakottaa pisteosoituksen tekstin, blokin tai blokin attribuutin lisäys- eli viitepisteeseen.
	<b>Lähin piste, Nearest, nea</b> Nearest pakottaa pisteosoituksen lähimpään kuva-alkion pisteeseen. Pakotuksen avulla voidaan tarttua tarkasti kuva-alkioon, kuten viivaan tai ympyrän kehään.
	<b>Ei mitään, None, non</b> None kumoaa kaikki jatkuvan pakotuksen määrittäykset, jolloin mikään pakotus ei ole voimassa. Pisteosoitus voidaan syöttää täysin vapaasti.
	<b>Jatke, Extension, ext</b> Extension pakottaa pisteosoituksen viivan tai kaaren oletettuun jatkeeseen.
	<b>Tilapäinen viitepiste, Temporary Tracking Point, tt</b> Temporary Tracking Point -toiminnon avulla viitataan olemassa olevaan geometriaan tilapäisen origon, viitepisteen avulla. Toiminnolla voidaan tehdä vain vaaka- ja pystysiirtymiä.
	<b>Tilapäinen viitepiste, Snap From, fro</b> Snap From -toiminnolla viittaukset tilapäiseen origoon syötetään suhteellisten koordinaattien avulla.

## 10.6. Piirtäminen

Seuraavana on esitely muutamia piirtämiseen ja piirretyn muokkaamiseen tarvittavia komentoja.



**POLYLINE**

 Command: **pline**, **pl** (*Draw - Polyline*)

Polyline-alkio koostuu yhdestä tai useammasta viiva- ja kaarisegmentistä. Jokaisella segmentillä voi olla oma leveys niin kuin segmenttien alku- ja päätepisteilläkin.

Polyline-komennon optioita ovat:

- **arc - a**: Kaarevan segmentin piirto
- **Close - c**: Sulkee polyline-alkion palaten lähtöpisteeseen
- **Halfwidth - h**: Segmentin leveyden asetus. Leveys kasvaa keksiviivan molemmin puolin. Alku- ja päätepisteen leveys voidaan määrittää erikseen.
- **Length - l**: Piirtää edellisen suuntaisen, määritellyn pituiseen viivasegmentin
- **Undo -u**: Peruu viimeksi piirretyn segmentin
- **Width - w**: Segmentin leveyden asetus. Määritelty leveys on segmentin reunasta reunaan. Alku- ja päätepisteen leveys voidaan määrittää erikseen.

Polyline-alkioita voidaan muokata kuten muitakin piirustusalkioita tai pelkästään polyline-alkioiden muokkaukseen tarkoitetulla pedit-komennolla. Pedit-komennolla voidaan liittää toisiinsa useita polyline-alkioita tai muuntaa viiva- ja kaari-alkiota polylineiksi.

**line, l** (*Draw - Line*)

jonka alku- ja päätepiste tunnetaan. Viivoista voidaan muodostaa monikulmio, jonka alkio on erillinen ja yksittäin muokattavissa. Viivamonikulmio voidaan muokata.

Command: **offset**, **o** (*Modify - Offset*)

Ohjelmalla voidaan luodaan yhdensuuntainen kopio viivasta, kaaresta, ympyrästä ja spline-käyrästä ja polyline-alkiosta. Kopion etäisyys voidaan määrittää. Ohjelma kysyy etäisyyden lisäksi kummalle puolelle alkuperäistä viivaa.

**TRIM**

Command: **trim**, **tr** (*Modify - Trim*)

Trim-komento rajaa tai katkaisee kuva-alkion muilla kuva-alkioilla. Trim-komennossa ensin valitaan ne kuva-alkiot, joilla halutaan leikata tai rajata, ja sen jälkeen katkaistavat kuva-alkiot.

**br** (Modify - *Break*)

Maistaan kuva-alkioita. Ensimmäinen valintapiste on samalla ensimmäinen. Valintapisteitä voidaan siirtää f-optiolla (First).

---

**nen**

**c, cha** (Modify - *Chamfer*)

Viistään viistää ei-yhdensuuntaisten viivojen, polyline-alkioiden ja kaareiden. Viisteen koko voidaan määrittää viistettävien viivojen suuntaisten avustuksella.

Käytetään käsittelyyn.

---

**nen**

Command: **fillet, f** (Modify - *Fillet*)

Pyöristetään kahden kuva-alkion leikkauskohta. Pyöristyksen koko säädetään. Pyöristyksiä voidaan tehdä viivoille, polylineille, apulinjoille, kaarille ja suorille.

Käy myös 3D-mallien käsittelyyn.

---