

Ohjelmoinnin perusteet



Otavan Opiston julkaisusarja Jari Sarja 2015
Creative Commons Nimeä 4.0 Kansainvälinen (CC BY 4.0)

Sisällysluettelo

Yleistä.....	2
Tehtävä 1.....	3
Tehtävä 2.....	3
Historia.....	3
Tehtävä 3.....	4
Tehtävä 4.....	5
Vuokaavio.....	5
Kuinka vuokaavio piirretään?.....	6
Tehtävä 5.....	7
Tehtävä 6.....	7
Algoritmi.....	8
Tehtävä 7.....	9
Ohjelmointikielen osia.....	9
Lähteet.....	12

Yleistä

Kun haluamme koneen tekevän jotain meidän puolesta, on käytettävä apuna ohjelmointia. **Kone ei ymmärrä tavallista puhuttua kieltä vaan sille on annettava ohjeet koodattuina.** Siksi joissakin yhteyksissä puhutaan koodauksesta. Myös ohjelmoijista käytetään toisinaan nimitystä koodari, jolla viitataan heidän tekemään koodaustyöhön (kirjoittavat koodia, joista muodostuu ohjelma).

Ohjelmointi on yksinkertaisimmillaan toimintaohjeiden antamista ennalta määrätyn toimenpiteen suorittamista varten. Ohjelmointia esiintyy meidän jokaisen arkielämässä lähes päivittäin. Ohjelmoinniksi voidaan laskea esimerkiksi, että annamme jollekulle puhelimesta ajo-ohjeet. Myös mikroaaltouunin käyttö on ohjelmointia, koska annamme uunille ohjeet siitä, kuinka kauan ja kuinka suurella teholla sen tulee toimia. Lopputulos on kiinni siitä kuinka olet onnistunut ohjelmoimaan. Mikroaaltouuni toteuttaa ohjelmointisi prikulleen, jäinen pulla ja räjähtynyt tomaatti ovat huonon ohjelmoinnin seurauksia.



Kuva 1, Flickr: Sharp Carousel Microwave by Chris Kelly (CC BY 2.0)

Tehtävä 1

Kuinka ohjelmoisit mikroaaltouunin (teho ja ika) siten, että pakastettu pizzapala olisi sopivan lämpöinen syöntiä varten.

Tehtävä 2

Kuinka ohjelmoisit pesukoneen pesemään villapaitasi?

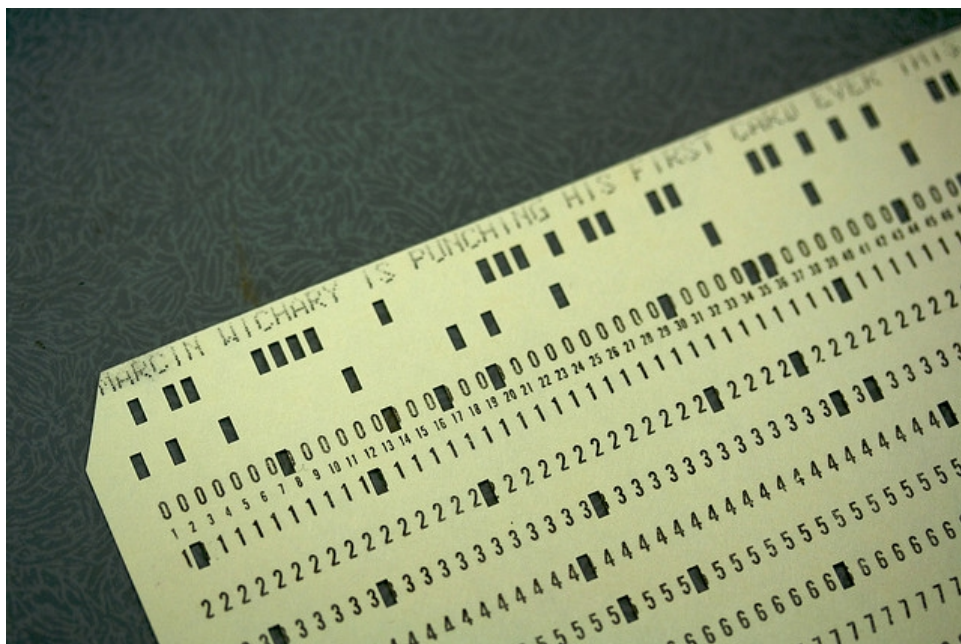
Historia

Ohjelmointi on liittynyt läheisesti siihen, että on ollut jokin iso ratkaisematon ongelma. Tässä alla on kuvaus yhdestä sellaisesta ongelmasta.

”Vuonna 1880 Amerikassa oli 50 miljoonaa asukasta ja tietojen kokoaminen ja analysointi kesti kaikkiaan seitsemän vuotta. Koska väestön määrän arvioitiin nousevan 60 miljoonaan, pelättiin virastossa, että vuoden 1890 väestölaskenta kestäisi yli kymmenen vuotta eikä siis valmistuisi edes seuraavan väestölaskennan alkuun mennessä.” (Lindström 2002, 3)

Kun vuoden 1890 väestölaskennan aika lähestyi, Väestölaskentavirasto järjesti kilpailun löytääkseen uuden nopean menetelmän väestölaskennan suorittamiseksi. Kilpailun voitti **Herman Hollertin keksintö**, joka perustui reikäkortteihin. Henkilön tiedot syötettiin reikäkortteihin, joissa oli kenttiä. Jokaisessa positiossa oli kaksi vaihtoehtoa eli siinä oli joko reikä tai ei. Esim. sukupuoli voitiin merkata siten, että reikä tietyissä positiossa kuvasi henkilön olevan nainen ja viereisessä positiossa hänen olevan mies. (Lindström 2002, 4)

”Hollerithin menetelmää käyttäen Väestölaskentavirasto säästi henkilötyön vähenemisellä 5 miljoonaa dollaria, joka oli 10 kertaa enemmän kuin oli arvioitu” (Lindström 2002, 4)



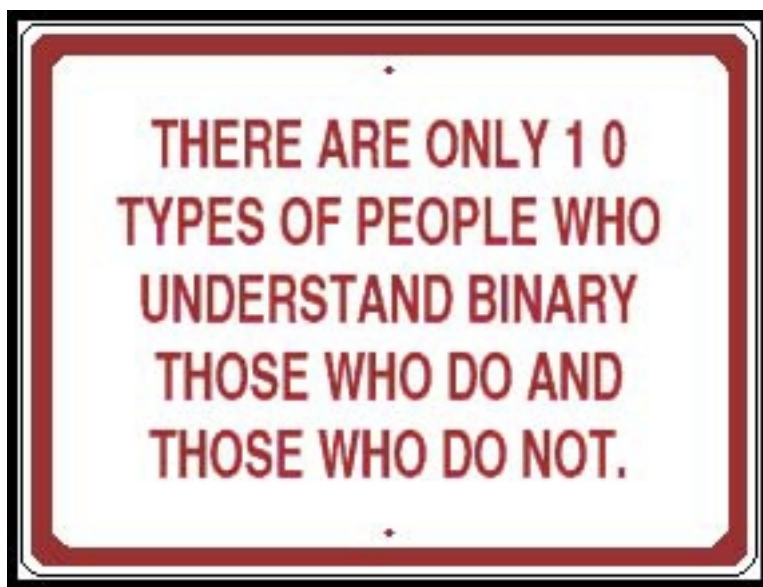
Kuva 2. Flickr: My first punched card by Marcin Wichary (CC BY 2.0)

Reikäkortti perustuu binäärilukujärjestelmään, jossa on käytössä vain kaksi lukua eli 0 ja 1. Nykyään ohjelmointi on aiempaa helpompaa, koska on olemassa useita apuvälineitä. Lisäksi ohjelmointikielet ovat kehittyneet inhimillisempään suuntaan, ne ovat siis helpommin tulkittavissa kuin ennen.

Tosin siellä ihan ytimessä tietokoneet ovat edelleenkin ykkösiä ja nolliä. Prosessoreissa ja muissa digitaalipiireissä nolla ja yksi esitetään usein kahtena eri jännitetasona.

Tehtävä 3

Mitä alla olevassa kyltissä lukee suomeksi?



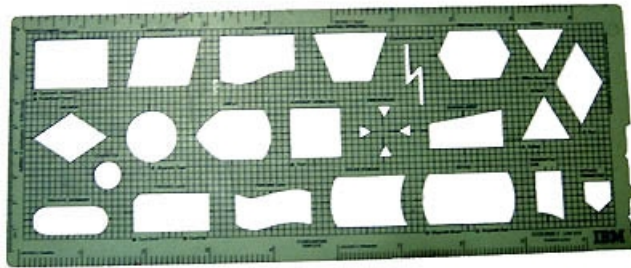
Kuva 3. Flickr: Binary by Xerones (CC BY-NC 2.0)

Tehtävä 4

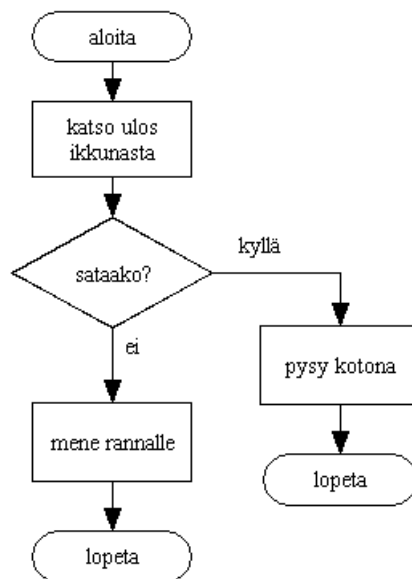
Pinääripentti väittää, että $1+1$ on 10 . Hän on oikeassa, miten se on mahdollista?

Vuokaavio

Vuokaaviota on käytetty pitkään. Jopa niin pitkään, ettei ketään ole nimitetty vuokaavion isäksi tai keksijäksi. Yleensä vuokaaviolla kuvataan prosesseja. Vuokaavio tarjoaa yhteisen symbolisen kielen mille tahansa ilmiölle, joka sisältää prosesseja. Vuokaavioita käytetäänkin kuvaamaan mitä erilaisemmissa yhteyksissä. Helpon vuokaavion voi piirtää lyijykynällä. Tyylikkäämmän kaavion saa aikaiseksi käyttämällä apuna valmista sabluunaa.



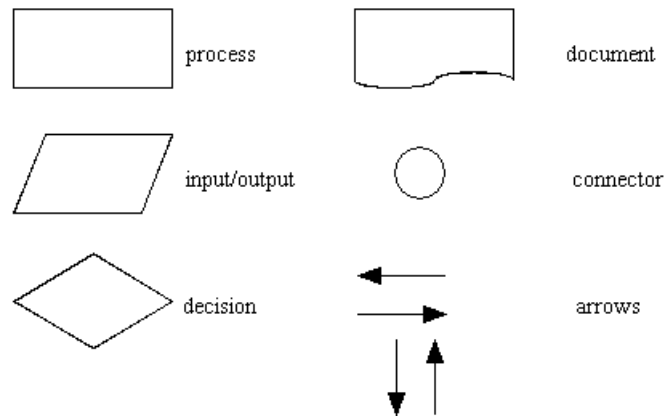
Alla oleva vuokaavio kuvaa kuinka syntyy päätös jäädä kotiin tai lähteä rannalle. Kumpi valinnoista toteutuu riippuu siitä sataako ulkona vai ei.



Kuinka vuokaavio piirretään?

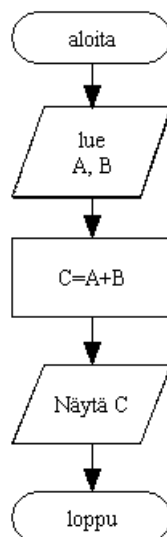
Ohjelmointia käsittelevissä vuokaavioissa ei tarvitse opetella kuin kuusi erilaista symbolia. Suorakaiteen muotoinen symboli (process) kuvaa tietokoneen prosessointia. Tietokoneen suorittama tehtävä kuvataan kirjoittamalla suorakaiteen sisään suoritettava tehtävä. Suorakaide (input/output) kuvaa sitä, että tietokoneelle syötetään tietoa tai että tietokone

tuottaa tietoa. Salmiakkikuvion tehtävä on kuvata päätöksentekoa, joka perustuu ennalta määriteltyihin valintoihin. Suorakaide, jonka alareuna on aaltomainen, kuvaa tiedon varastointia esim. tulostamista kuvataan tällä symbolilla (papeille tallentuu tulos). Pieni ympyrä kuvaa yhdistämistä. Tätä tarvitaan kun esim. A4:n loppuu kesken ja on jatkettava seuraavalle sivulle. Silloin ympyrän sisään kirjoitetaan kirjan (esim. A) ja nuoli viiva alaspäin, ja jatkettavalle sivulle myös A-kirjain ja viiva ylöspäin. Nuolet kuvaavat prosessin suuntia eli mikä seuraa mitäkään vaihetta.



Seuraavaksi esittelen vuokaavion, jonka pitää suorittaa yhteenlaksutehtävä. Ohjelma koostuu seuraavista vaiheista:

- 1) lue luvut
- 2) laske luvut yhteen
- 3) kirjoita tulos
- 4) loppu



Ohjelmaksi muutettuna yllä olevasta vuokaviosta voisi saada aikaiseksi vaikka tällaisen lyhyen JavaScriptillä kirjoitetun ohjelman.

```
<script language="javascript">
var a=333
var b=644
var c=a+b
document.write(c)
</script>
```

Tehtävä 5

Piirrä vuokaavio onkimisesta.

Tehtävä 6

Piirrä vuokaavio lottorivin kirjoittamisesta.

Vinkki. Hyvä ohjelma vuokaavioiden piirtämiseen on Dia Portable.
Ohjelman voi asentaa esim. muistitikulle eli sitä ei varsinaisesti asenneta.
Riittää, kun ohjelman lataa ja purkaa.

Algoritmi

Kirjoitettaessa koneelle kelpaavia ohjeita, joudutaan suoritettavana oleva toimenpide pilkkomaan osiin yksinkertaisiksi toimenpiteiksi.

Toimenpidesarjan tulee olla yksikäsitteinen eikä siinä saa esiintyä ristiriitaisuuksia tai rinnakkaisuuksia. Tällaista yksikäsitteistä kuvausta tehtävän ratkaisuun tarvittavista toimenpiteistä kutsutaan algoritmiksi. Algoritmi on siis joukko ohjeita, jotka ohjaavat tehtävän suorittamista. Alla on kuvattuna esimerkki algoritmista, jonka aiheena on kahvinkeitto. (Männistö 2001.)

Kahvinkeitto:

1. Täytä pannu vedellä.
2. Keitä vesi.
3. Lisää kahvijauhot.
4. Anna tasaantua.
5. Tarjoile kahvi. (Männistö 2001.)

Kahvinkeittoa kuvaava algoritmi voidaan vielä purkaa vaihe vaiheelta pienempiin osiin.

Kahvinkeitto:

1. Täytä pannu vedellä:
 - 1.1. Pistä pannu hanan alle.
 - 1.2. Avaa hana.
 - 1.3. Anna veden valua, kunnes vettä on riittävästi.
2. Keitä vesi:
 - 2.1. Aseta pannu hellalle.
 - 2.2. Kytke virta keittolevyyn.
 - 2.3. Anna lämmitä, kunnes vesi kiehuu.
3. Lisää kahvijauhot:
 - 3.1. Mittaa kahvijauhot.
 - 3.2. Sekoita kahvijauhot kiehuvaan veteen.
4. Anna tasaantua:
 - 4.1. Odota, kunnes suurin osa kahvijauhoista on vajonnut pannun pohjalle.
5. Tarjoile kahvi:
 - 5.1. laita pöytään kupit, lusikat ja lautaset
 - 5.2. kaada kahvi kahvikuppiin. (Ks Männistö 2001.)

Tehtävä 7

Kirjoita algoritmi, joka opastaa henkilön postilta koulusi pihaan.

Ohjelmointikielen osia

On olemassa monia eri ohjelmointikieliä, mutta ne koostuvat yleensä samanlaista tai samantyyppistä osista. Tässä keskitytään muutamiin perusosiin, jotka löytyvät lähes poikkeuksetta mistä tahansa ohjelmointikielestä.

Muuttujat:

Muuttujiin ohjelma tallentaa tietoa ohjelman suorittamisen ajaksi. Niistä voi myös hakea tietoa ohjelman suorittamisen aikana. Muuttujalla on jokin nimi ja arvo.

Esim. muuttuja pisteet JavaScript -kielessä

```
<script language="javascript">
var pisteet = 0;
</script>
```

Sama Scratchillä.



Operaattorit

Operaattoreita tarvitaan erilaisten laskutoimituksien ja vertailujen suorittamiseen.

Esim. yhteenlasku JavaScriptillä.

```
<script language="javascript">
var luku1 = 149;
var luku2 = 150;
var luku3 = luku1+luku2;
document.write(luku3);
</script>
```

Sama Scratchillä ja lopputulos.



Ehtolauseet

Ehtolauseiden avulla suoritetaan valintoja. Ehto määrittää mikä valinta suoritetaan.

```
<script language="javascript">
if(9<10)
{
document.write("9 on vähemmän kuin 10")
}
</script>
```

Sama Scratchillä ja lopputulos, kun ohjelma on suoritettu.



Toistolauseet

Toistorakenteet ovat silmukoita, joiden avulla voidaan suorittaa jokin asia useamman kerran peräkkäin.

```
<script language="javascript">
for (i = 0; i <= 10; i++)
{
document.write("9 on vähemmän kuin 10 <br>");
}
</script>
```

Sama Scratchillä ja liike mukana. Ilman liikettä kirjoittaisi päälle, jolloin näyttää kirjoittavan vain kerran.



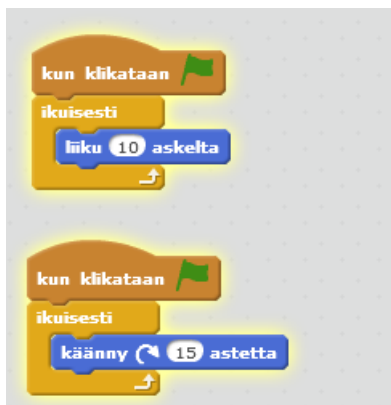
Funktiot

Funktio on aliohjelma, joka palauttaa tuloksen. Tässä alla olevassa esimerkissä laskuri laskee pinta-alan.

```
<script language="javascript">
function pintaala(luku1, luku2)
{
return luku1 * luku2;
}
```

```
var ala = pintaala(3.25, 5.5);  
document.write(ala);  
</script>
```

Scratch -ohjelmalla ei ole varsinaisia aliohjelmiä, mutta siellä voidaan pilkkoa asioita. Alla on esimerkki pyörivästä kissasta.



Lähteet

Hyvönen, M. Lappalainen, V. Ja Lakanen A-J. (2013). *Ohjelmointi 1 C#*. (3. korjattu painos). [Online <http://kurssit.it.jyu.fi/ITKP102/monistecs/html/moniste.html>]. Luettu 26.3.2015.

Lindström, M. (2002). *Reikäkortin historia*. Seminaarityö Helsingin yliopisto. Tietojenkäsittelytieteen laitos.

Männistö, T. (2001). Johdatusta ohjelmointiin. [Online: http://appro.mit.jyu.fi/2001/syksy/johdatusohjelmointiin/moniste/ciao_t.htm]. Luettu 5.12.2004.