

Kari Kuutti
Tutkimusassistentti, FM
Oulun Yliopisto
Tietojenkäsittelyopin laitos
Linnanmaa, 90570 OULU

SAMPO- ja LOGO-kielet tietotekniikan opetuksessa

1. Lausekielinen ohjelmointi - koulua vai elämää varten?

1.1 Mitä elämässä tarvitaan?

Jonkinlaisen kuvan siitä, minkälaisia asioita ihmisten pitäisi osata, saa tarkastelemalla yritysten ja yhteisöjen tällä hetkellä työntekijöilleen järjestämää tai ostamaa koulutusta. Kun varsinaisille ammattilaisille tarkoitettu koulutus karsitaan pois, voidaan jäljelle jäävä "käyttäjä"-koulutus luokitella vaikkapa seuraavasti:

- yleiskatsaukset tietojenkäsittelyyn
- tietokoneen operointi ja valmisohjelmien käyttö
- tietotyön työkaluohjelmien käyttö (tekstinkäsittely, taulukkolaskenta jne.)
- tiedonhaku ja tiedonhallinnan alkeet
- valmistautuminen tk-kehitysprojektiin käyttäjän edustajana.

1.2 Mitä koulu pyrkii opettamaan?

Tietotekniikan opetus kouluissa on perinteisesti keskittynyt ohjelmointikieliin, eritoten BASIC-kieleen. Ankaraa polemiikkaa on käyty siitä, että opetettavan ohjelmointikielen tulisi olla "ammattimaisempi". "Edistyskellisissä" kouluissa onkin siirrytty "parempaan" PASCAL:iin. Seurauksena

on voinut olla, että muille kysymyksille on jäänyt aikaa entistä vähemmän.

Vasta valmistuneet peruskoulun ja lukion oppimäärät ovat melkoinen harppaus oikeaan suuntaan: tietotekniikka on muutakin kuin ohjelmointia, vieläpä suurimmalta osaltaan. Inertia ja ohjelmoinnin opetuksen järjestämisen suhteellinen helppous vaikuttanevat kuitenkin sillä tavoin, ettei nykytilanne tule kovin nopeasti olennaisesti muuttumaan.

Muistin virkistämiseksi viralliset kurssien yleisotsikot:

Peruskoulu:

- tietotekniikan perusteet
- tietokone ilmaisuvälineenä
- tiedonhallinta ja tiedonsiirto
- ohjelmointi
- automaatio
- tietokoneen toiminta

Lukio:

- tietotekniikan peruskurssi
- ohjelmointi

1.3 Vertailu

Koulussa opetetaan tai voitaisiin ainakin oppimäärien puitteissa opettaa suurinta osaa niistä asioista, joita työelämässä tällä hetkellä näyttää tarvittavan - painotus tuntuu vain kovin erilaiselta: kouluissa ohjelmointi on tärkeimpiä ellei tärkein opetettava asia, kun taas työelämän käyttäjäkoulutuksesta se puuttuu tyystin. Mistä tämä johtuu?

Työelämän asenteelle löytyy kaksikin selitystä: 1) Lausekielistä ohjelmointia ei voi oppia kunnolla lyhyillä kursseilla ja 2) Normaali käyttäjä ei tarvitse sitä taitoa.

Koulujen tilanteelle on historiallisia syitä: mikäli tietokoneella halusi aikaisemmin tehdä ylipäänsä jotakin, oli kirjoitettava oma sovellutus. Siksi ohjelmoinnin opettaminen oli aivan luonnollista. Ohjelmointi on myös pitkään ollut lähes ainoa ala, jolta on ollut saatavissa edes jonkinlaista oppimateriaalia. Lisäksi ohjelmointia ympäröi ehkä vielä jossakin tietty "ammattilaisuuden sädekehä".

2. Missä ohjelmointitietoja ja -taitoja tarvitaan?

Vaikka jokaisen työelämässä ei selvästikään tarvitse osata ohjelmointia, ei tämä silti merkitse sitä, etteikö ohjelmointiin liittyvillä valmiuksilla olisi käyttöä. Alussa esitellystä käyttäjäkoulutuksesta voidaan löytää ainakin seuraavia tilanteita, joissa ohjelmointitiedoista on apua:

- tietokoneen operointi ja valmisohjelmien käyttö - **kun edetään alkeistasoa pidemmälle** (esim. käyttöjärjestelmän käsky tiedostot jne.)
- Tietotyön työkaluohjelmien käytössä - **kun edetään alkeistasoa pidemmälle** (esim. tekstinkäsittelyssä postituslistat ja kehittynyt editointi, taulukkolaskennassa itsetehdyt makrokäskyt jne.)
- tiedonhaussa ja -hallinnassa - **kun edetään alkeistasoa pidemmälle** (esim. käskysarjat toistuvia hakuja varten jne.)
- toimittaessa osallisena ("käyttäjänä") tietojärjestelmän kehittämisessä (kaikkalainen kommunikointi ohjelman tai ohjelmiston varsinaisien rakentajien kanssa)

Edelläolevasta huomataan, että joillakin ohjelmointiin liittyvillä asioilla on keskeinen merkitys menestykselliselle tietotekniikan käyttämiselle. Edelleen huomataan, että nämä asiat eivät mitenkään liity jonkin tietyn lausekielen hallintaan, vaan kyse on algoritmisen ajattelun ja ohjelmoinnin perusideoista ja -käsitteistä.

JOHTOPÄÄTÖS: KOULUN TULISI OPETTAA JUURI NÄITÄ ASIOITA - OHJELMOINNIN YLEISSIVISTYSTÄ.

3. Ohjelmoinnin yleissivistys

Seuraavassa on oma, tiivistetty näkemykseni useimpiin tämän päivän tarpeisiin riittävästä, ohjelmointiin liittyvästä yleistietoudesta.

1) Tietokonetta ohjataan yksinkertaisilla **peruskäskyillä**. Tietokone on neuroottisen tarkka siitä, että käskyjen muoto on oikea.

2) Peruskäskyjä yhdistelemällä saadaan **rakenteita**. Rakenteita on kol-

mentyyppisiä:

- peräkkäisrakenne
- toistorakenne
- ehtorakenne.

3) Jonkin tehtävän suorittamista varten tarvitaan **ohjelma**. Ohjelma syntyy liitettäessä yhteen joukko rakenteita, jotka yhdessä toteuttavat tarvittavan **käsittelyalgoritmin**. Algoritmin lisäksi ohjelmaan kuuluvat **tiedot**, joita algoritmilla käsitellään.

4) Ohjelman laatiminen edellyttää, että on olemassa jokin **tavoite**, johon ohjelmalla pyritään. Mitä tarkemmin tämä tavoite kyetään ilmaisemaan, sitä helpompaa on itse ohjelman kehittäminen.

5) Ohjelman monimutkaisuutta voidaan hallita **modularisoinnilla ja hierarkisuudella**.

6) Ohjelmointityössä ei useinkaan heti ensimmäisellä kerralla saavuteta toivottua tulosta, vaan sitä joudutaan lähestymään **kokeilun ja virheiden korjaamisen** kautta. Virheitä voi syntyä useilla eri tasoilla: käskyjen syntaksissa, rakenteiden logiikassa, käsittelyalgoritmissa tai ohjelman tavoitteen määrittelyssä. Mitä ylemmällä tasolla virhe tehdään, sitä vaikeampi se on korjata.

Edellisen luvun perusteella voi tietysti nyt kysyä: Jos tämä kerran on yleisivistystä, niin eikö se sovellu erittäin hyvin opiskeltavaksi juuri lausekielen avulla?

Vastaus on - kyllä, mutta kahdella ehdolla:

1) Opettajan on kyettävä opettamaan yleisiä periaatteita ao. lausekieltä välineenään käyttäen.

2) Aikaa on oltava riittävästi.

Mikäli ehdot eivät toteudu, jää metsä puilta näkemättä: yleisivistyksen sijasta opetellaan syntaksia.

4. SAMPO- ja LOGO-kielet yleisivistuksen opetuksessa

SAMPO- ja LOGO-kielet tarjoavat yleistietouden oppimiseen vaihtoehdon perinteisille lausekielille. Tämän vaihtoehdon voima perustuu seuraaville tekijöille:

- 1) Oppiminen on "tekemällä oppimista", mikä on ihmiselle kaikkein luonnollisin tapa.
- 2) Liikkeelle päästään tutuista, "arkijärjellä" välittömästi ymmärrettävistä ilmiöistä, ilman että tietokoneesta tarvitsee ennakkoon tietää tai opetella yhtään mitään.
- 3) Kyetään heti ensihetkestä käskemään konetta ja tekemään jotakin näkyvää.
- 4) Kielten syntaksin alkeet on helppo oppia - etenkin SAMPOssa, missä käskysanat ovat suomenkielisiä.
- 5) Oppimiseen tarvittava virhe-korjaus-kokeilu - sykli on hyvin nopea.
- 6) Kielet tukevat yleisivistuksen oppimista - esim. modulaarisuus ja hierarkisuus tulevat käyttöön "luonnostaan".

Millä tavalla nämä hyvät ominaisuudet on sitten voitu saavuttaa? Tekemällä oppiminen on kaiken ydin. Jotta ohjelmointikieli tukisi sitä, täytyy siinä olla ainakin nämä SAMPO:n ja LOGO:n ominaisuudet:

- tulkkityyppinen käsittely, jotta palaute saadaan välittömästi.
- ns. konnagrafiikan tuki, eli kielen täytyy tukea sellaista vuorovaikutteista grafiikkaa, jossa komennot annetaan suhteessa nykypisteeseen (ei absoluuttikoordinaatteja). Tällöin voidaan lähteä liikkeelle arkikokemuksella ymmärrettävistä asioista.
- laajentuvuusominaisuus, eli kielen sanastoa on voitava joustavasti laajentaa.

5. Didaktiikasta

Nykyinenkin ohjelmoinnin peruskäsitteiden opettaminen perustuu itse asiassa tekemällä ja kokeilemalla oppimiseen, koska vasta niiden kautta kyetään uudet asiat todella sisäistämään. Pulmana on vain se, etteivät laajimmin käytössä olevat välineet tätä juuri tue, eikä koko asiaa opetuksessa muutenkaan tiedosteta.

SAMPO- ja LOGO-kielet, joissa kaikki perustuu oppijan omaan, aktiiviseen toimintaan, mahdollistavat opetuksen keskittymisen ydinasioihin. Tämä vaatii kuitenkin uudenlaista opetuksellista käsittelyä. LOGO-kielen kehittäjä S. Papert on ilmaissut epäilynsä siitä, tarvitaanko opettajaa enää ollenkaan. Tämä on varmasti virheellinen arvio. Pikemminkin opettajan rooli korostuu entisestään, mutta nykyisen detaljitason asemasta laajemmalla, opetuksen suunnittelun tasolla.

Mikäli ohjelmoinnin yleissivistystä halutaan opettaa, täytyy opettajan tietää, missä asioissa nämä yleisivistävät piirteet ilmenevät ja minkälaisen tehtävien avulla oppija nopeimmin ja helpoimmin muodostaa juuri oikean käsityksen niistä. Ilman kokonaisohjausta käyttöön väistämättä tuleva sokea-yrityksen- ja erehdyksen menetelmä ei varmasti johda juuri minnekään. Myös oppimisen motivoinnissa on opettajan osuus ratkaiseva.

Tällaista kokonaisohjauksen käsikirjaa ei valitettavasti vielä ole olemassa, vaan SAMPO:n ja LOGO:n tehokkaaseen hyödyntämiseen tarvittaisiin lisää tietoa tietotekniikan didaktiikasta.

Tiedon vähyys ei kuitenkaan ole yksin SAMPO:n ja LOGO:n ongelma, vaan koko tietotekniikan koulutus kärsii didaktisten ohjenuorien puutteesta. Näkymä on huolestuttava: kuinka suuri osa julkisen vallan ja yritysten tällä hetkellä vuosittain tietotekniikan koulutukseen sijoittamista sadoista miljoonista menee suoraan Kankkulan kaivoon - opetusmenetelmien vuoksi?