

Kysymyksiä ja vastauksia ohjelmoinnista:

ONKO SE VAIKEAA?

ONKO OHJELMOINTI VAIKEAA?

Ohjelmointi on ohjeiden antamista tietokoneelle – ja sen pohtimista, miten ohjeet kannattaa antaa. Kun kysytään, onko ohjelmointi vaikeaa, on kysymys hiukan samanlainen kuin se, onko kirjoittaminen vaikeaa.

Lapsikin oppii kirjoittamaan yksinkertaisia asioita. Toisaalta tullakseen kirjallisuuden Nobel-voittajaksi tulee tuntea laaja valikoima käytössä olevia kirjoittajan työkaluja ja niiden parhaita käyttötapoja. On osattava peilata omaa tekemistä olemassa olevaan ja toisaalta oltava tarpeeksi luova tehdäkseen omia oivalluksia.

Siis: Ohjelmointi ei välttämättä ole vaikeaa, mutta se tarjoaa loputtomasti haasteita. Ohjelmoinnissa ei ole koskaan mielestään täysin oppi-

nut samalla tavalla kuin kirjailija ei ole koskaan mielestään täydellinen kirjoittaja.

MITÄ OHJELMOIJA TEKEE TYÖKSEEN?

Ohjelmoijia ja ohjelmointiin liittyviä työtehtäviä on hyvin erilaisia. Jotkut ohjelmoijat keskittyvät kirjoittamaan ohjelmakoodia ja miettimään, miten tietty sovellus tai ohjelmisto tai sen osa kannattaa järkevimmin toteuttaa. Toisena päivänä sama ihminen voi keskittyä testaamaan toisen tekemää ohjelmaa tai etsimään kirjoitetusta koodista ajatus- tai muita virheitä.

Joskus ohjelmoija keskittyy pohtimaan projektin pohjapiirrustusta: mikä on paras yhdistelmä erilaisia palikoita, jotka yhdessä muodostavat sujuvasti, nopeasti ja virheettä toimivan ohjelmiston.

Ohjelmointi on toisaalta ongelmanratkaisua, toisaalta taidetta.

Ohjelmointiin kuuluu myös paljon yhteistyötä: yhdessä pohtimista ennen koodauksen aloittamista, sen aikana sekä ohjelmiston testausvaiheessa.

Kun ohjelmoijan asiantuntijuus kasvaa, hänestä voi tulla esimerkiksi muista ohjelmoijista koostuvan tiimin vetäjä. Tällöin päätehtävä on auttaa muita suoriutumaan parhaansa mukaan, kysyä kriittisiä kysymyksiä ja tarjota pohdinta-apua pulmatilanteisiin.

KUKA PÄÄTTÄÄ, MILTÄ VALMIS OHJELMA NÄYTTÄÄ KÄYTTÄJÄLLE?

Tyypillinen tietokoneohjelma, esimerkiksi älypuhelimella käytettävä sovellus, jakaantuu eri osiin. Yksi ”osa” sovellusta on se, miltä se näyttää älypuhelimien ruudulla sitä käyttävälle ihmiselle. Tuota osaa kutsutaan käyttöliittymäksi.

Käyttöliittymällä tarkoitetaan esimerkiksi älypuhelimien Facebook-sovelluksessa kaikkea sitä, minkä käyttäjä näkee ja mitä hän voi sormillaan

koskea tehdäkseen Facebookissa asioita.

Käyttöliittymän suunnittelu on yhden tai useamman käyttöliittymäsuunnittelijan tehtävä.

Käyttöliittymän suunnittelijan vastuulla on paitsi se, miltä valmis ohjelmisto näyttää käyttäjälle, myös se, kuinka nopeaa ohjelmiston käyttö on oppia ja kuinka tehokasta ja tarkoituksenmukaista käyttö on pidemmällä aikavälillä.

Käyttöliittymäsuunnittelija ei välttämättä ohjelmoi. Hänen tehtävänsä saattaa olla esimerkiksi puhua ohjelmiston tulevien käyttäjien kanssa ja piirtää – joko käsin tai tietokoneella – keskustelujen pohjalta ehdotuksia mahdollisiksi käyttöliittymämalleiksi.

MIKÄ TEKEE HYVÄN OHJELMOIJAN?

Hyviä ohjelmoijia on hyvin erilaisia. Kuitenkin ainakin seuraavat piirteet voidaan useimpien mielestä liittää hyvään ohjelmoijaan:

Halu ja kyky oppia uutta. Ohjelmointi on toisaalta käsityötä, toisaalta taidetta ja toisaalta ongelmanratkaisua. Kuten kaikissa noissa osaluissa, myös ohjelmoinnissa tulee sitä paremmaksi, mitä enemmän sitä tekee. Ihminen oppii esimerkiksi näkemään erilaisia tapoja ohjeistaa tietokonetta yhä tarkemmin, lyhyemmin ja elegantimmin.

Kunnianhimo. Hyvä ohjelmoija miettii aina, mi-

**Monilla osa-alueilla
voi pärjätä ohjelmoijana,
vaikka ei olisi ollut luokan
matikkanero.**

ten tietyn koodinpätkän voisi kirjoittaa vieläkin paremmin. (Tässä ”paremmin” tarkoittaa samaa kuin yllä: tarkemmin, lyhyemmin, elegantimmin.)

Luovuus. Ohjelmoinnissa samaan lopputulokseen voi yleensä päätyä kymmenillä eri tavoilla, ja näistä tavoista useakin erilainen voi olla vieläpä täysin järkevä. Hyvä ohjelmoija ei jumitu toistamaan itseään vaan pohtii ongelmaa monilta eri suunnilta. Jos ongelma on vähänkin monimutkainen, kaksi huippuohjelmoijaaakin ratkaisee sen todennäköisesti hiukan eri tavoilla.

Tarkkuus. Kun tietokoneelle antaa ohjeita, on otettava huomioon monia asioita oikeakielisyydestä siihen, että kone ei varmasti voi ymmärtää eri tavoin ihmisen tarkoitusperiä. Siksi ohjelmoijan on oltava tarkkana kuin porkkana. Tarkkuus kehittyy kuitenkin tekemisen kautta. Ohjelmoijaksi haluavan ei tarvitse olla valmiiksi erityisen pikkutarkka luonne.

PITÄÄKÖ OHJELMOIJAN OSATA MATEMATIIKKA?

Ohjelmointitaito ja matemaattiset kyvyt liitetään usein toisiinsa. Joissain asiayhteyksissä matematiikasta onkin hyötyä, sillä ohjelmoija saattaa joutua kertomaan tietokoneelle, millainen käyrä sen on osattava piirtää tai ymmärrettävä, että tietty ongelma ratkeaa parhaiten hyödyntämällä vaikkapa jakojäännöksen käsitettä laskutoimituksissa.

On kuitenkin tärkeää ymmärtää, että ohjelmoinnissa oleellisempaa on yleinen kyky pohtia asioiden loogisia syitä ja seurauksia kuin laskennallinen matemaattinen osaaminen. Monilla osa-alueilla voi pärjätä ohjelmoijana, vaikka ei olisikaan ollut luokan matikkanero. Monille halu oppia ohjelmoimaan paremmin antaa kuitenkin lisämotivaatiota ymmärtää myös matematiikkaa paremmin.

MITÄ EROA ON ”OHJELMOINNILLA” JA ”KODAAMISELLA”?

Ohjelmointi ja koodaaminen tarkoittavat käytännössä samaa asiaa eli tietokoneen ymmärtämän ohjelmakoodin kirjoittamista.

Tämä ei tarkoita, etteikö termeihin liittyisi sekaannusta:

Tvt, atk, it, ict, tietotekniikka, tietojenkäsittelyoppi, tietojenkäsittelytiede, digitaalinen lukutaito... Tietokoneisiin liittyvä sanasto on ongelmallista, sillä käytämme eri ilmaisuja sekaisin.

Myös ”tietotekniikkataidot” voivat nykyisin olla mitä tahansa lähtien hiiren käyttötaidosta ja taulukkolaskennasta aina siihen, että rakentaa itse tietokoneen, kirjoittaa ohjelman tai pystyttää lähiverkon. Samalla tavalla tietotekniikka koulussa voi tarkoittaa oppiainetta jossa opetetaan turvallista internetin käyttöä tai sitä, miten lähiverkko yhdistää printterit ja koneet toisiinsa.

MIKSI ON NIIN MONIA OHJELMOINTIKIELIÄ? MITEN NE EROAVAT TOISISTAAN?

Ohjelmoijat rakastavat työkaluja, ja samalla tavalla kuin puusepät tai arkkitehdit rakentavat omia erikoistuneita työkalujaan, ohjelmoijat voivat kirjoittaa uusia ohjelmointikieliä ratkaisemaan erilaisia ongelmia.

Jotkut kielistä antavat ammattilaiselle vapauden lähes kaikkeen (C), toiset ovat hyviä verkkosivustokäyttöön (JavaScript) ja kolmannet helppoja aloittaa (Ruby). Jotkut ovat hyviä matematiikassa (Haskell), toiset tilastotieteissä (R). Jotkut sopivat Applen älypuhelimille (Objective-C), toiset vaikkapa puhelinverkkojen toteuttamiseen (Erlang).

Peruskoulussa hyviä perusvarmoja ja monipuolisia aloituskieliä ovat esimerkiksi JavaScript, Ruby ja Python (ks. myös sivut 40–44 ja seuraava kohta.)

**Peruskoulussa hyviä
aloituskieliä ovat
esimerkiksi JavaScript,
Python ja Ruby.**

MIKÄ OHJELMOINTIKIELI ALOITTELIJAN KANNATTAA VALITA?

Aloittelijan kannattaa valita kieli riippuen siitä, mitä haluaa saada aikaan. Tietokoneen työpöydällä, internetissä ja puhelimessa olevat sovellukset on saatettu koodata eri kielillä. Jos tietää mitä haluaa tehdä, kannattaa googlata mistä ideaasi muistuttava sovellus koostuu. Kaikki kielet muistuttavat kuitenkin perusperiaatteiltaan toisiaan, joten aloittaa voi melkein mistä vain.

Monille hyvä ensimmäinen ohjelmointikieli on sellainen, jolla saa nopeasti aikaan jotain näkyvää, eikä asentamiseen ja työympäristöön hallintaan tarvitse käyttää paljon aikaa. Jos haluaa rakentaa verkkosivuston, riittää opetella HTML:ää ja

CSS:ää, jotka eivät ole edes varsinaisia ohjelmointikieliä (ks. sivu 122). Jos verkkosivustosta haluaa interaktiivisen, kannattaa opetella JavaScriptiä. Lisää vinkkejä ohjelmoinnin aloittamiseen löytyy tämän oppaan osiosta 3 alkaen sivulta 76.

VOIKO SAMALLA OHJELMOINTIKIELELLÄ OHJELMOIDA TIETOKONEELLE JA ÄLYPUHELIMELLE?

Riippuu tapauksesta, sillä eri älypuhelimien käyttöjärjestelmille ohjelmoidaan eri ohjelmointikielillä. Esimerkiksi Android-nimistä käyttöjärjestelmää käyttäville Samsungin tai HTC:n puhelimille tarkoitetut sovellukset ohjelmoidaan Java-nimisellä ohjelmointikielellä, jota käytetään myös muualla. Applen valmistamien iPhone-puhelimien iOS-käyttöjärjestelmälle taas ohjelmoidaan Objective-C:llä.

Eri älypuhelimille ohjelmoidaan eri ohjelmointikielillä.

MIKSI TÄMÄ ON TÄRKEÄÄ?

KOODI2016:N TUKIJAT KERTOVAT

”JOKA PÄIVÄ MIETIN, ETTÄ OLISIPA KIVA, KUN YMMÄRTÄISI PAREMMIN OHJELMOINTIA”

Taru Rastas, viestintäneuvos, Liikenne- ja viestintäministeriö



”**D**igitaalisuus tulee toimialasta riippumatta arkeen. Julkiset palvelut, yritysten liiketoimintaprosessit, sairaalat, kaupat... Maailma vain digitalisoituu, ja meidän pitää huolehtia siitä, että meillä on tekijöitä.

Jotta voimme vastata tarpeeseen, meidän pitää tarttua asiaan kaikilla ikäasteilla. Lapset on saatava digitaaliseen maailmaan koulusta lähtien siten, että he kykenevät ottamaan tekijän roolin.

Ja mihin ikinä se nuori haluaa tulevaisuudessa erikoistua – digitaalisuus tulee opinnoissa hyödyksi. Luonnontieteissä tai bisnesmaailmassa.

Liikenne- ja viestintäministeriö kehittää digitaalista Suomea. Me luomme mahdollisuuksia sille, että kuluttajilla ja yrityksillä olisi käytössään paras mahdollinen tieto- ja viestintätekniikka, ja siitä löydettäisi uutta kasvua Suomelle.

Usein kuullaan, että Nokia on romahtanut, ja tän alan insinööreille ei ole kysyntää. Mutta kyse on siitä, että IT:tä hyödyntävät alat kehittyvät valtavasti, ja sitä osaamista pitää koko ajan kehittää.

Usein uuden oppiminen vaatii poisoppimista siitä, että kun on aina tehty yhdellä tavalla, niin miksi pitäisi alkaa nyt tehdä asioita uudella tavalla.

Sen auttamiseksi pitäisi saada läpi ajatus siitä, että ohjelmointi ei ole erityistaitoa jollekin tietylle pienelle porukalle.

Joka päivä mietin itse, että olisipa kiva, kun ymmärtäisi paremmin jotain ohjelmointiin liittyvää. Ei ole tarkoitus, että jokaisesta meistä tulee hyvä koodari, mutta sen ymmärtäminen, että mihin kaikkeen sillä pystytään, ja millaisia asioita ratkotaan, ja miten se linkittyy erilaisiin toimintoihin – se on tärkeää.”