

---

This is an electronic reprint of the original article.  
This reprint may differ from the original in pagination and typographic detail.

Slotte Dufva, Tomi

## Feministinen ohjelmointikieli

*Published in:*  
Avantgarde, tiede ja teknologia

Julkaistu: 01/12/2019

*Document Version*  
Publisher's PDF, also known as Version of record

*Published under the following license:*  
Määrittelemätön

*Please cite the original version:*  
Slotte Dufva, T. (2019). Feministinen ohjelmointikieli. teoksessa H. Sederholm, & I. Hautamäki (Toimittajat), *Avantgarde, tiede ja teknologia* (Sivut 44-49).  
<https://finnishavantgardenetwork.files.wordpress.com/2019/12/avantgarde-tiede-ja-teknologia-final-3.12.pdf>

---

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

# Feministinen ohjelmointikieli

*Tomi Slotte Dufva*

Vuonna 2013 Georgia Institute of Technologyn väitöskirjatutkija Ari Schlesinger toi esille ajatuksen feministisestä ohjelmointikielestä, eli feministiseen teoriaan perustuvasta tavasta rakentaa ohjelmia (Schlesinger, 2013). Viime vuosina ohjelmistot, erityisesti koneajattelua ja isoja datasettejä käyttävät ohjelmat, ovat osoittautuneet sovinnistisiksi ja rassistisiksi vinouttaen näkemyksiämme useammastakin asiasta ja vähemmistöstä (Apprich, Chun, Cramer, & Steyerl, 2018; Devlin, 2017; McQuillan, 2016). Esimerkiksi kuvantunnistusalgoritmi luokittelee keittiössä seisovan ihmisen naiseksi ja vasaraa heiluttelevan mieheksi (Devlin, 2017). Tämän lisäksi esimerkiksi liikkeentunnistusta hyödyntävät laitteet ovat kalibroituja valkoihoisille, mikä johtaa tilanteisiin, jossa saippuan annostelija ei tunnista muita etnisyyksiä. Vastaava ongelma löytyy myös aktiivisuusrannekkeista tai sykkeentunnistimista, jotka käyttävät samankaltaista optista tunnistusta (Fussel, 2017). Schlesinger ei kuitenkaan puhunut tämän tapaisista vinoutumista ohjelmistojen rakenteessa tai käytössä, vaan itse ohjelmointikielestä; Hän kysyi, olisiko käyttämäämme ohjelmointikieltä mieltä ajatella uudella tavalla?

Schlesingerin kommentti sai vastaansa vihapostin hyökyaallon (“r/ProgrammerHumor - A feminist programming language called C Plus Equality.” 2014; Schlesinger, 2014). Väitettiin, että koodi ja sen alla hyrräävä logiikka on neutraalia, tämän vuoksi feminismiin mukaan tuonti olisi aivan naurettavaa. Ajatus sai vastaväittäjät jopa luomaan kokonaisen vitsikielen, C++:n Githubiin, ohjelmoijien suosimaan koodin jakopaikkaan (Blasar, 2013). Vitsinä tehdyn ohjelmointikielen nimi C++ itsessään on koodia osaaville aukeava sisäpiirivitsi, jonka voi lukea sekä suositun ohjelmointikielen C++:n loogisena muunnelmana, tai C+equalitynä, tasa-arvoon viittaavana merkinä, jota tekijät ilmeisesti pitävät huvittavana. Githubiin luotu sivusto esittelee feministisen ohjelmointikielen filosofian muutaman esimerkin avulla. Koko C++ -sivusto rakentuu hyvin samalla tavoin kuin otsikkokin: se vilisee sisäänpäin lämpiävää naljailua saaden populistisia ja sovinnistisia sävyjä. Mielenkiintoista onkin, miten jotkut sinällään hyvin viattomat ehdotukset voivat saada näin paljon vastustusta aikaan.

Tämän vuoden alussa New Yorkin yliopiston professori Alex Galloway kirjoitti artikkelin, jossa tuki Schlesingeriä (Galloway, 2019). Galloway toi tekstissään esiin, miten Schlesingerin teesi

matematiikan ja ohjelmoinnin logiikan mahdollisista vinoutumista on yleisesti tyrmätty; ajatusta koodin, matematiikan tai logiikan feministisyydestä on pidetty miltei mahdottomana. Galloway nostaa esiin seitsemän yleistä tapaa kieltää ajatus koodin vinoutuneisuudesta. Gallowayn tapaukset kattavat C++-vitsikielen tapaiset totaaliset kieltämiset edeten vivahteikkaampiin kieltoihin, joissa ohjelmointikieli ja / tai matematiikka erotetaan kulttuurista ja politiikasta, jolloin ohjelmointikielten loogista ja matemaattista rakennetta pidetään neutraalina, se ikään kuin kumpuaa puhtaasta platonilaisesta ideoiden maailmasta, eikä sitä sovi liata ihmisten kulttuurilla. Galloway nimeää nämä kieltämiset median peloksi; matematiikkaa, logiikkaa tai ohjelmointikieliä ei haluta nähdä mediana, koska tämä toisi väistämättä esiin median erilaiset kielen ja rakenteen tulkitsemisen tavat, genret, eri tavat representoida ja merkitä jotain. Kuitenkin neutraalius itsessään on yksi vallankäytön muoto ja tapa vaientaa vähemmistöjä. Galloway tulkitseekin median pelon olevan itseasiassa feminiinisyyden pelkoa.

## Mitä on ykkösen ja nollan välillä?

Mitä feministinen ohjelmointikieli voisi sitten olla? Kaikesta saamastaan vihapostista huolimatta Schlesinger on jatkanut ajatuksen eteenpäin viemistä tuoden esille erilaisia tapoja kontekstualisoida ideaa feministisestä ohjelmointikielestä. Hän käsittelee feministisessä kontekstissa toteutettua suoritettavaa koodia kirjallisena tekstinä, taiteena, sekä ideoi alkuperäisen idean mukaista, täysin uutta feminististä ohjelmointikieltä (Schlesinger, 2014). Feministisen ohjelmointikielen ajattelu ja luominen on kuitenkin haastavaa. Vallitseva kulttuuri, jossa suurin osa ohjelmoijista on valkoisia miehiä (Stack overflow, 2017) ja joita alan homogeneisyys ei edes häiritse (Collins, 2017), ei varmasti auta asiaa. Tämän lisäksi feministinen filosofia vaatii radikaalisti uudenlaista suhtautumista koodiin, koodin rakenteeseen ja rakennustapaan.

Otan tässä tekstissä lähteekseni aiemmin mainitun feministisen ohjelmointikielen pilaversion C++:n ja tarkastelen sitä tosissani feminististä teoriaa vasten. Pyrin toisin sanoen miettimään, onko vitsissä totuuden siemen, ja olisiko meidän todellakin aika miettiä aivan uudella tavalla ohjelmointikieltä, teknologiaa, jonka varaan rakentuu yhä suurempi osa koko yhteiskunnastamme. Tarkastelen erityisesti kolmea kohtaa C++-kielen Githubissa esitetystä kolmentoista kohdan filosofiasta käsitellen siihen sisältyviä ajatuksia ohjelmoinnista. Muut filosofian kolmestatoista kohdasta osoittavat auttamaton tietämyksen puutetta feministisestä teoriasta, ja/tai ovat niin sovinistisia ja rasistisia laadultaan, että niiden esiin nostaminen olisi tässä tekstissä täysin hedelmätöntä.

1. The language is to be strictly interpreted using feminist theory. Under no circumstances should the language be compiled, as compilation and the use of a compiler imposes an oppressive and toxic relationship between the high-level descriptive language and the low-level machine code that does all the labo(u)r. Instead, C+= is interpreted, which fosters communication, itself a strong female trait (Blaster, 2013).

C+= ohjelmointikielen filosofia alkaa esityksellä siitä, miten kieli pitää tulkita vain feminististen teorioiden mukaan, ilman kääntämistä, koska kääntäminen sisältää aina alistamista<sup>1</sup>. Vaikka tämä huomio ensin näyttäytyy (ja on) huonona vitsinä sekä feministisen teorian ymmärtämättömyytenä, tuo tämä huomio esiin mielenkiintoisia lähtökohtia koko digitaalisesta teknologiasta. Muun muassa mediatutkija Douglas Rushkoff on pannut merkille, miten digitaalisen teknologian binaarinen rakenne aiheuttaa itsessään vinoutumia ja kuinka nämä vinoutumat kasautuvat binäärisyyden päälle luoduissa ohjelmistoissa sekä toisintuvat ohjelmistojen käyttäjien kulttuureissa (Rushkoff, 2010; 2013). Yksi mielenkiintoinen kysymys onkin, ovatko feministinen filosofia ja binaarinen logiikka millään muotoa yhteensopivia? Kysymystä voi laajentaa myös pohtimalla, miten hyvin humanististen tieteiden esiin tuomat asiat ylipäätään ovat käännettävissä binaariseen logiikkaan. Miten ohjelmoidaan epävarmuutta, dekonstruoituneita hierarkioita ja monin tavoin taipuneita rihmastoja systeemillä, joka viimekädessä haluaa selkeän vastauksen, kyllä vai ei, yksi vai nolla? Mies vai nainen?

Toisaalta kääntäminen tuo esiin kysymyksen siitä, mitä kieli on ja miten käänämme sitä koneiden ymmärtämäksi kieleksi. Toisin sanoen, kääntämisessä on kysymys älykkyyden ja ajattelun siirtämisestä ihmisen kielestä koneen kieleksi. Hayles onkin todennut, että yksi digitaalisen teknologian mukanaan tuoma asia on kehon redusoiminen pois sellaisesta älykkyyden määritelmästä, jonka mukaan älykkyys on formaalia symbolien vaihtoa, ei eletty kehollinen kokemus (Hayles, 2008). Kehollisuuden näkökulmasta digitaalinen teknologia ja sen älykkyys näyttäytyy auttamattomasti vinoutuneena, jolloin kääntämisen uudelleen ajattelu voisi tuoda mielenkiintoisia tapoja ymmärtää ja luoda ohjelmistoja. Suuri osa ohjelmistoista esimerkiksi rakentuu tietyn hierarkian varaan; ohjelmistoissa on keskus, jonka varaan muut osat rakentuvat ja määrittelevät itsensä. Millainen olisi feministinen ohjelmointikieli, joka pystyisi ottamaan huomioon vailla keskusta olevat, monimutkaiset, eri osioiden väliset suhteet?

---

<sup>1</sup> Kääntäminen (englanniksi compiling) merkitsee tässä tapauksessa ohjelmointikielissä tapahtuvaa kielen kääntämistä ihmisten ymmärrettävistä ohjelmointikielistä (high-level languages) koneen ymmärrettäväksi kieleksi (machine language). Kääntämisen ymmärtämiseen tarvitaan hyvä annos teknologista osaamista, mutta tiivistettynä voi todeta kääntämisen olevan binäärisen konekielen ja ihmisille ymmärrettävän ohjelmointikielen välillä tapahtuva etukäteen suunniteltu ja ohjelmoitu käänös. Tietokone muodostuu miljoonista, tai miljardeista transistoreista, -sähkövirtaa kahteen suuntaan ohjaavista komponenteista- jotka siis muodostavat digitaalisen logiikan binaarisen perustan. Koska miljardien ykkösten ja nollien lukeminen on suurimmalle osasta ihmisistä mahdotonta, on tietokoneiden ohjelmointiin luotu useita kieliä, jotka muistuttavat enemmän tai vähemmän kirjoitettuja kieliä. Näiden kielten konventioiden kautta kääntäjä sitten tulkitsee kielen binaariksi.

## Muutoksen polut

2. No constants or persistence. Rigidity is masculine; the feminine is fluid. I.e., fluid mechanics is hard for men 'because it deals with "feminine" fluids in contrast to "masculine" rigid mechanics' (Blaster, 2013).

Filosofian toinen kohta kiinnittyy ohjelmointikielen muuttujien pysyvyyteen. Muuttuja eli variaabeli, on ohjelmoinnissa ja matematiikassa käytetty symboli, jolle voidaan asettaa erilaisia arvoja, kuten  $x=1$ . Muuttujan arvo voi muuttua, mutta vain määrittelemällä tämä muutos. Edelleen enimmäkseen maskuliinista typeryyttä huokuva C++:n filosofian kohta tuo kuitenkin esille jo myös ensimmäiseen kohtaan liittyvän seikan: miten määritellään jokin, jossa muuttuja ei ole selvä tai muutos on moniulotteinen tai jatkuvassa muutoksessa? Muutoksen uudelleenajattelun tuominen mukaan ohjelmointikieliin voisi itseasiassa olla hyvinkin mielenkiintoista. Esimerkiksi filosofi Gilles Deleuze on tuonut esiin, miten jokainen toisto pitää sisällään muutoksen luoden virtuaalista potentiaalisuuksien kenttää, mikä mahdollistaa asioiden muuttumisen (Deleuze, 1994). Entä miten voisi ottaa huomioon eri muutoksien välisen eron, esimerkiksi intersektionaalisen feminismin kautta? Millainen olisi ohjelmointikieli tai millaisia ohjelmia rakentaisimme, jos ottaisimme huomioon muuttujien toistojen jatkuvan muutoksen sisäsyntyisenä osana ohjelmointikieltä?

9. Women are more social than men. Hence, social coding should be the only option. The code only runs if it is in a public repo (Blaster, 2013).

Viimeinen esille nostamistani kohdista ei sinällään liity ohjelmointikielen rakenteeseen, mutta tuo hyvin esille ohjelmoinnin käytänteitä ja kulttuureita, sitä miten koodia luodaan ja millaisiin tarkoituksiin. Jos sivuutamme stereotyyppioinnin, tuo tämä kohta esiin vapaan ohjelmiston ideologian (Stallman, Gay, & Lessig, 2009) kaltaisen tarpeen koodin saavutettavuudelle. Jokaisella tulisi olla mahdollisuus halutessaan katsoa ja nähdä käyttämiensä ohjelmistojen rakenne ja halutessaan mahdollisuus muokata sitä. Tämä vapaan ohjelmiston radikaali ideologia pitää sisällään täysin erilaisen sosioekonomisen mallin, joka varmasti romuttaisi useammankin ohjelmistotalon. Toisaalta, suurin osa internetiä ei toimisi ilman avoimen lähdekoodin Linux-palvelimia. Avoimuus, jakaminen ja laajempi kommunikaatio ohjelmoijien ja muiden välillä voisi myös muuttaa koko ohjelmoinnin tyyliä ja tarkoitusta. Morozov on esimerkiksi arvostellut ohjelmistofirmoja aina start-upeista mammuttimaisiin ohjelmistotaloihin asti digitaalisesta solutionismista, tavasta ratkaista monimutkaisia ongelmia yksinkertaistamalla ne yksinkertaisiksi algoritmeiksi (Morozov, 2014). Solutionismi saattaa luoda hetkellisesti taloudellisesti menestyviä yrityksiä, mutta yleensä se siirtää tai pahentaa jo olemassa olleita ongelmia (Morozov, 2014; 2015). Kenties moisia ongelmia ei syntyisi, jos ohjelmoinnin käytännöt olisivat jo sisäsyntyisesti kommunikoivia ja monialaisia?

Tämän lisäksi kohta tuo myös esille ohjelmoijien homogeenisyyden: miespuolisten osuus ohjelmoijista on edelleen korkea, ja jopa avoimen lähdekoodin puolella esimerkiksi naisten osuus on vain 1,5% (Collins, 2017; Stack overflow *Stack Overflow Developer Survey 2017*, 2017). Ja kuten Schlesingerin vastaanottama palaute osoittaa, muutoksen aikaan saaminen on vaativaa. Voikin kysyä, miten ohjelmointikieli voisi muuttua, kun koko kulttuuri sen ympärillä on näin yksipuolista.

Vaikka itse käytännön konstruktio feministisestä ohjelmointikielestä saattaa olla nykyteknologiassa ja vallitsevassa ilmapiirissä mahdoton, on ajatus feministisestä ohjelmointikielestä mielestäni erittäin tärkeä. Se tuo esille ohjelmoinnin moniulotteiset ja syvät kytkökset muuhun maailmaan: ohjelmoinnin tulee jollain keinoin ottaa huomioon humanististen tieteiden ja taiteen esiintuoma muuttuva monisyinen maailma. Feministinen ohjelmointikieli tuo esille, miten ohjelmoinnin ajatukset oppimisesta, älykkyydestä ja maailmasta itsessään ovat vanhentuneita ja rajallisia. Tämän vuoksi tavassamme ajatella digitaalisuutta ja digitaalista teknologiaa tarvittaisiinkin ei pelkästään pientä linjanmuutosta vaan suuri korjausliike. Tähän feministinen ohjelmointikieli tuo perustavanlaatuisia ideoita. Ja kuten Schlesingerin tutkimus osoittaa, itse käytännön konstruktio, toimiva feministinen ohjelmointikieli ei kuitenkaan ole mahdoton.

## Kirjallisuus

- Apprich, C., Chun, W. H. K., Cramer, F., & Steyerl, A. H. (2018). Pattern Discrimination (pp. 1–139). Minneapolis: University of Minnesota press. [http://doi.org/ 10.14619/1457](http://doi.org/10.14619/1457)
- Blastar, E. (2013, December 15). ErisBlastar/cplusequality. Retrieved September 25, 2019, from <https://github.com/ErisBlastar/cplusequality>
- Collins, K. (2017, March 29). Tech is overwhelmingly male and men are just fine with that. Retrieved August 10, 2019, from <https://qz.com/940660/tech-is-overwhelmingly-male-and-men-are-just-fine-with-that/>
- Deleuze, G. (1994). *Difference and Repetition*. London: Continuum.
- Devlin, H. (2017, April 13). AI programs exhibit racial and gender biases, research reveals. Retrieved April 16, 2017, from <http://www.theguardian.com/technology/2017/apr/13/ai-programs-exhibit-racist-and-sexist-biases-research-reveals>
- Fussel, S. (2017). Why Can't This Soap Dispenser Identify Dark Skin?
- Galloway, A. R. (2019, September 26). Are Algorithms Biased? Retrieved February 2, 2019, from <http://cultureandcommunication.org/galloway/are-algorithms-biased>
- Hayles, N. K. (2008). *How We Became Posthuman*. University of Chicago Press.
- McQuillan, D. (2016). Algorithmic paranoia and the convivial alternative. *Big Data & Society*, 3(2), 205395171667134–12. <http://doi.org/10.1177/2053951716671340>

- Morozov, E. (2014). *To Save Everything, Click Here. The Folly of Technological Solutionism*. PublicAffairs.
- Morozov, E. (2015, February 1). Why cities need to fight Uber and give people a real transport choice. Retrieved May 10, 2017, from <http://www.theguardian.com/commentisfree/2015/feb/01/cities-need-to-fight-uber-transport-choice-evgeny-morozov>
- r/ProgrammerHumor - A feminist programming language called C Plus Equality. Some changes include renaming of "class" to "social\_construct", the prohibition of "y" variables, and "bugs" are called "snowflakes". (2014). r/ProgrammerHumor - A feminist programming language called C Plus Equality. Some changes include renaming of "class" to "social\_construct", the prohibition of "y" variables, and "bugs" are called "snowflakes".
- Rushkoff, D. (2010). *Program Or Be Programmed*. OR Books.
- Rushkoff, D. (2013). *Present Shock*. Penguin.
- Schlesinger, A. (2013, November 26). Feminism and Programming Languages. Retrieved October 20, 2017, from <https://www.hastac.org/blogs/ari-schlesinger/2013/11/26/feminism-and-programming-languages>
- Schlesinger, A. (2014, July 14). A Feminist Programming Language? – FemTechNet. Retrieved September 25, 2019, from <https://femtechnet.org/2014/07/a-feminist-programming-language/>
- Stack Overflow, Developer Survey Results 2017*.
- Stallman, R. M., Gay, J., & Lessig, L. (2009). *Free Software, Free Society*. Boston: Free Software Foundation.