

ELEC-C7110

Informaatioteknologian perusteet

Kalevi Kilkki

Tietojenkäsittelyn perusteista
11.2. & 12.2.2019

Kurssin rakenne

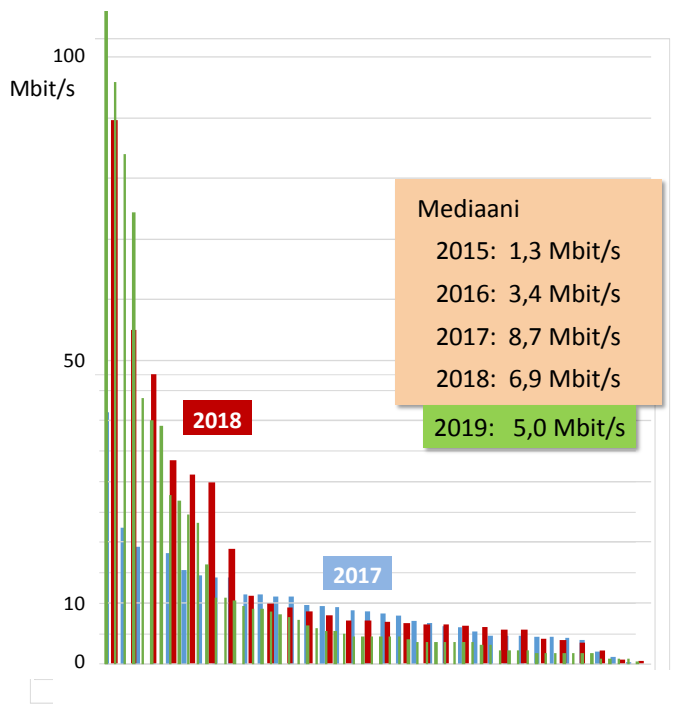


- » 7+2 asiakokonaisuutta = jakso (viikko)
- » Viikko: 2+2 tuntia luentoja ma 14:15, TU1/1017, ti 14:15, E/Y124
2 tuntia harjoituksia ke 12.15, TU1/1017
 1. Johdanto ma 7.1. & ti 8.1.
 2. Tietoliikennealan palvelut ma 14.1. & ti 15.1.
 3. Tiedonsiirto ma 21.1. & ti 22.1.
 4. Kiinteät verkot ma 28.1. & ti 29.1.
 5. Matkaviestintä ma 4.2. & ti 5.2.
 6. Tietojenkäsittely ma 11.2. & ti 12.2.
 7. Internet ma 25.2. & ti 26.2.
 8. Kommunikaatioakustiikka ma 4.3. & ti 5.3.
 9. Puheteknologia ma 11.3. & ti 12.3.

Tenttiviikko (18.2. - 20.2.) ei opetusta

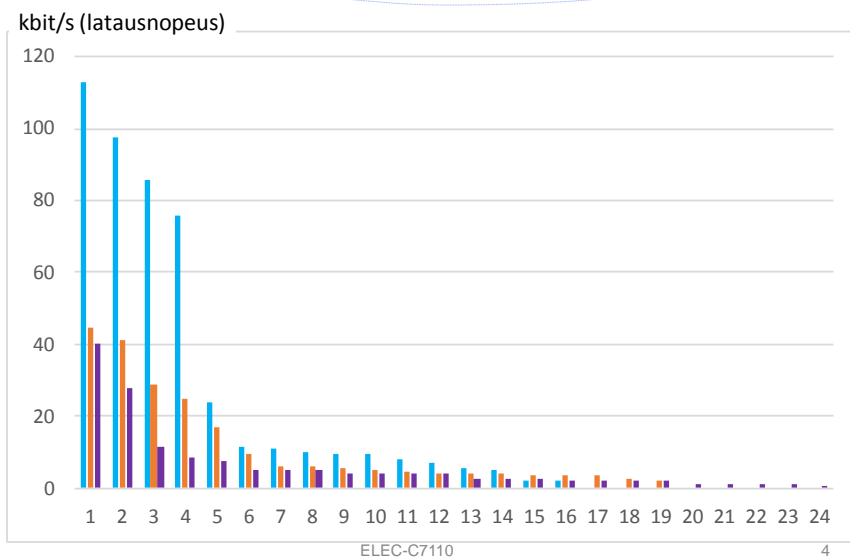
HUOM

TENTTI 10.4.2019



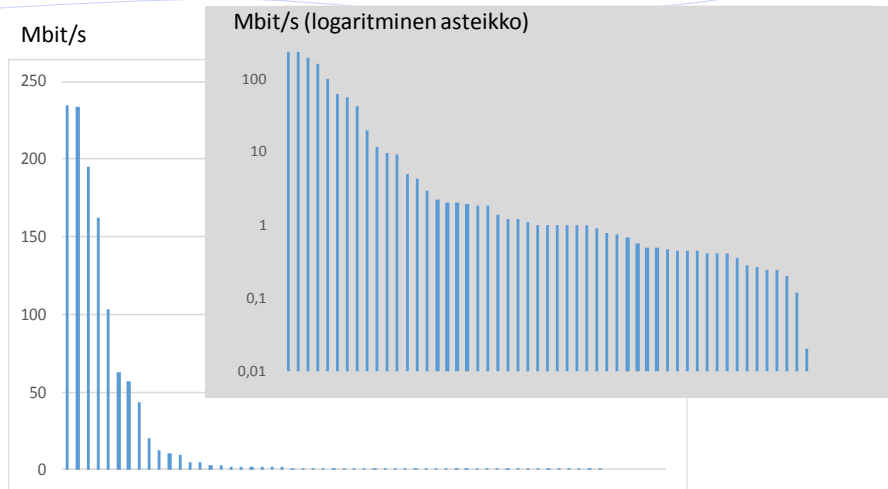
3

Nopeustestitulokset eri operaattoreille



4

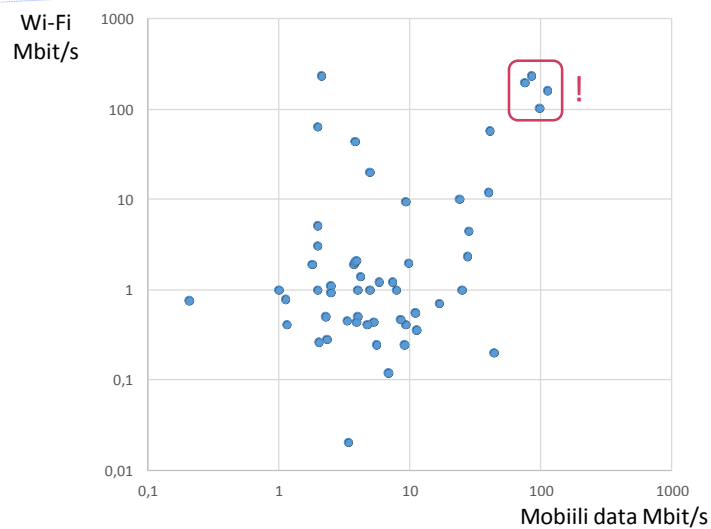
Wi-Fi nopeudet



ELEC-C7110

5

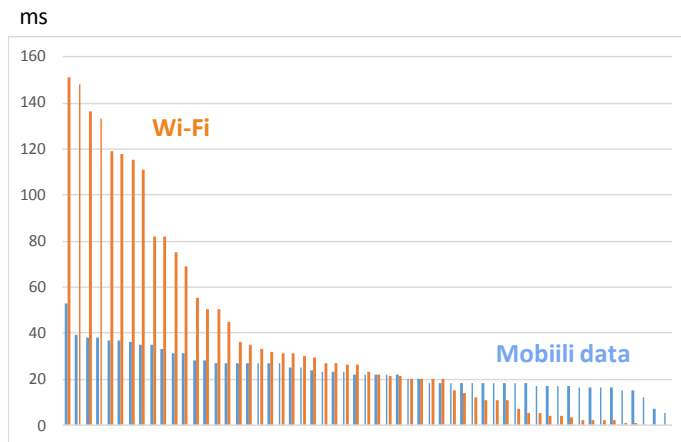
Mobiili data vs. Wi-Fi (per käyttäjä)



ELEC-C7110

6

Viiveet



ELEC-C7110

7

Tämän viikon kysymyksiä

1. Mooren laki ja sen vaikutukset
 - Mikä tarkoitetaan Moore lailla tietojenkäsittelyn yhteydessä? Pohdi miten Moore laki jatkuessaan vaikuttaa informaatioteknologian rooliin yhteiskunnassa.
2. Tietokoneen rakenneosat ja toimintaperiaate
 - Mitkä ovat tietokoneen keskeisimmät rakenneosat ja mitkä ovat niiden tehtävät?
3. Informaation esitystavat mukaan lukien negatiiviset luvut
 - Miten negatiiviset luvut yleensä esitetään binäärilukuina ja miten kyseinen esitystapa vaikuttaa yhteen- ja vähennyslaskuun?
4. Boolean algebra ja loogiset piirit
 - Miten looginen operaatio $(a \vee b) \wedge (a \vee c)$ voidaan toteuttaa loogisilla piireillä mahdollisimman yksinkertaisesti? Piirrä kuva!
5. Ohjelmointikielten tasot ja käyttöalueet
 - Miten ohjelmointikielten eri tasot eroavat toisistaan ja miksi ylipäätään on kehitetty eri tason kieliä?

ELEC-C7110

8

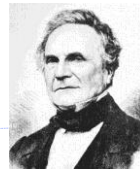
Muutama sana sanoista

- » **Tietojenkäsittely**
 - › ”Ammatillinen termi” (AMK): ohjelmointi ammattina
 - › Tietojenkäsittelytiede: mitä voidaan laskea ja miten se tehdään tehokkaimmin (yliopistot)
- » **Ohjelmointi**
 - › Sanasta ”ohjata” kehitetty sana (vuodelta 1880!)
 - *Ahokas, Osmo*: Normaalilyhtälöiden ratkaisun **ohjelmointi** matematiikkakone Eskolle. Diplomityö, Maanmittausosasto, 1959.
- » **Tietokone**
 - › Suomen Akatemian kielilautakunnan kokous joulukuun 1959
 - Vaihtoehtoja: matematiikkakone, tiedin, sähköaiivot, numerokone
 - *Mård, Matti*: Taloudellisen muuntajan laskeminen **tietokoneen** avulla. Diplomityö, Sähkötekniikan osasto, 1961. - 89 s.
- » **Tietotekniikka**
 - › Vasta 1980-luvulla valittu sana (TKK:n tarpeita varten?)
 - › Tietotekniikan koulutusohjelma aloitti 1985

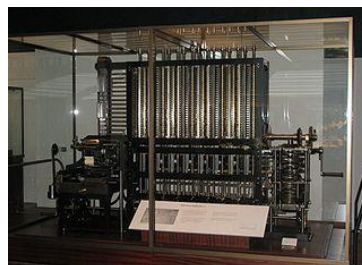
ELEC-C7110

9

Historiaa: Charles Babbage



- » Englanti, 1791-1871
- » Logaritmitaulukoiden laskemiseen
 - › Sai rahoitusta £17000 (suuri summa!)
 - › Mutta Babbage ei saanut koskaan laitteita täysin valmiiksi
 - Ei kehitysryhmää...
- » Babbagen esimerkki vaikutti vielä 1940-luvulla!



The [Science Museum's](#) Difference Engine No. 2 rakennettu Babbagen suunnitelman mukaan

ELEC-C7110

10

Ada Lovelace



Diagram for the computation by the Engine of the Numbers of Bernoulli



- » 1815-1852
- » Kehitti ajatusta ohjelmoitavasta laskentakoneesta
 - › Ensimmäinen algoritmi Bernoullin lukujen laskemiseksi (1842)
- » Ohjelmointi kehittyi eteenpäin vasta 100 vuotta myöhemmin
 - › Silloinkin ensin naisten toimesta!

Diagram for the computation by the Engine of the Numbers of Bernoulli. See Note G. (page 4)

Number of Operations	Variables used	Variables receiving results	Indications of change to the value of any Variable.	Statement of Results.	Data.												Working Variable.	
					x	y	z	w	v	u	t	s	r	q	p	o	n	m
1	$x = 1$	$y = 1$	$z = 1$	$x = 1$														
2	$x = 2$	$y = 2$	$z = 2$	$x = 2$														
3	$x = 3$	$y = 3$	$z = 3$	$x = 3$														
4	$x = 4$	$y = 4$	$z = 4$	$x = 4$														
5	$x = 5$	$y = 5$	$z = 5$	$x = 5$														
6	$x = 6$	$y = 6$	$z = 6$	$x = 6$														
7	$x = 7$	$y = 7$	$z = 7$	$x = 7$														
8	$x = 8$	$y = 8$	$z = 8$	$x = 8$														
9	$x = 9$	$y = 9$	$z = 9$	$x = 9$														
10	$x = 10$	$y = 10$	$z = 10$	$x = 10$														
11	$x = 11$	$y = 11$	$z = 11$	$x = 11$														
12	$x = 12$	$y = 12$	$z = 12$	$x = 12$														
13	$x = 13$	$y = 13$	$z = 13$	$x = 13$														
14	$x = 14$	$y = 14$	$z = 14$	$x = 14$														
15	$x = 15$	$y = 15$	$z = 15$	$x = 15$														
16	$x = 16$	$y = 16$	$z = 16$	$x = 16$														
17	$x = 17$	$y = 17$	$z = 17$	$x = 17$														
18	$x = 18$	$y = 18$	$z = 18$	$x = 18$														
19	$x = 19$	$y = 19$	$z = 19$	$x = 19$														
20	$x = 20$	$y = 20$	$z = 20$	$x = 20$														
21	$x = 21$	$y = 21$	$z = 21$	$x = 21$														
22	$x = 22$	$y = 22$	$z = 22$	$x = 22$														
23	$x = 23$	$y = 23$	$z = 23$	$x = 23$														
24	$x = 24$	$y = 24$	$z = 24$	$x = 24$														
25	$x = 25$	$y = 25$	$z = 25$	$x = 25$														
26	$x = 26$	$y = 26$	$z = 26$	$x = 26$														
27	$x = 27$	$y = 27$	$z = 27$	$x = 27$														
28	$x = 28$	$y = 28$	$z = 28$	$x = 28$														
29	$x = 29$	$y = 29$	$z = 29$	$x = 29$														
30	$x = 30$	$y = 30$	$z = 30$	$x = 30$														

Here follows a repetition of Operations destined to steady them.

⇒ 1930-luvun loppu



- » Tekniikan taso
 - › sähkömekaniikkaa (puhelinkeskukset) ja tyhjiöputket (radio, TV)
- » Tarve
 - › Ihmisen (computers!) suorittaman mekaanisen laskennan nopeuttaminen
 - Esim. ballististen taulukoiden laskenta 170 henkilöä (naisia)
- » Rinnakkaisia polkuja
 - › K. Zuse (Saksa, mekaaninen)
 - › J. Atanasoff (Iowa, tyhjiöputket + mekaniikka)
 - › Bell Labs (sähkömekaaninen)
 - › IBM: Mark I (sähkömekaaninen)
 - › Britannia: Colossus (salauksen purkuun, sähköinen)

Ensimmäinen tietokone

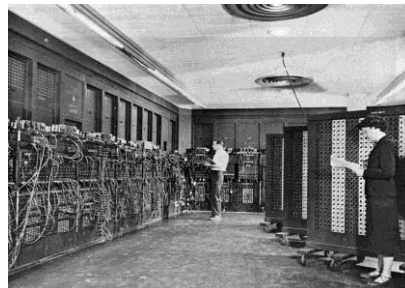
- » **Kun** kriteerit ovat:
 - › digitaalinen,
 - › toiminnaltaan täysin sähköinen,
 - › ohjelmoitava,
 - › yleiskäyttöinen ja
 - › toimiva,
- » **niin** ensimmäinen on ENIAC
- » muilla kriteereillä vastaus voi olla erilainen

ELEC-C7110

13

ENIAC

- » Ballististen lentoratojen laskemiseen (armeijan rahoittama)
 - › Valmistui 1945 ~ 6 000 000 \$
 - Ei aivan ehtinyt valmiiksi ennen sodan päättymistä
 - Laskettiin ensimmäiseksi vetypommin yhtälöitä heti sodan jälkeen
- » Rakenneosat
 - › 17 468 tyhjiöputkea
 - › 7 200 diodia
 - › 1 500 relettä
 - › 70 000 vastusta
 - › 10 000 kondensaattoria
 - › 5 miljoonaa liitosta



ELEC-C7110

14

Laskenta (käytti 10-järjestelmää!)

- » Yksi jakso = 0,2 ms
 - › 10-numeroinen kertolasku 2,8 ms
 - › Jakolasku tai neliöjuuri 28 ms
- » Luotettavuus
 - › Tyhjiöputkien vuoksi laite oli toimintakuntoinen n. 50% ajasta
- » Ohjelmointi
 - › Kytkimien ja kaapeleiden avulla (päivien työ)
 - › Datan syöttö reikäkorttien avulla
 - › Mahdollisesti silmukat, haarautumiset ja ”aliohjelmat”

ELEC-C7110

15

Kehitysaskel: transistori

- » Transistori 1947 (Bell Labs)
 - › John Bardeen, Walter Brattain ja William Shockley
- » Miksi Bell Labs?
 - › Monopolin asema
 - › Muutama visionäärihenkilö
 - › **Mutta Bell (AT&T) ei kaupallistanut kehitystä**
- » Tietokoneisiin vasta 1950-luvun lopulta alkaen
 - › (meilläkin oli kotona putkiradio ja tyhjiöputkiin perustuva TV pitkälle 1970-luvulle)



Although Shockley was not involved in the invention, and has never been listed on patent applications, Bell Labs decided that Shockley must appear on all publicity photos along with Bardeen and Brattain.

ELEC-C7110

16

Muisti – hankala haaste

Ensimmäinen kiintolevy
(pyörivä magneettinen levy
ja liikkuva lukupää)

- 1956
- *IBM 305 RAMAC-tietokoneeseen*
- 5 MB
- \$50000 (silloista rahaa)
- > 1000 kg
- voisi tallentaa yhden digitaalisen valokuvan



<http://www.retronaut.com/2011/12/5mb-hard-disk-drive-1956/>

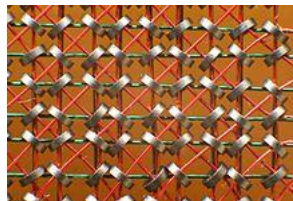
ELEC-C7110

17

Keskusmuisti

Ennen RAM-muisteja (1955 - 1975)

- » Ferriittirengasmuisti
 - › Haihtumaton (eli ei vaadi virkistystä)
 - › Lukuaika alle 1 μ s (1970-luvulla)
 - › Muistin koko suurimmillaan kymmeniä kilotavuja



ELEC-C7110

18

FBI fingerprint file room (~1964)



ELEC-C7110

19

Sormenjälkien digitalisointi (FBI/USA)

- » 83 miljoonaa korttia
- » ”laatikoiden aika on päätymässä v. 2014, 20 vuoden työn jälkeen”



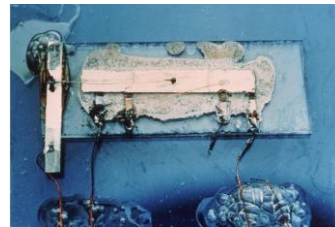
ELEC-C7110

20

Mikropiiri

» Ongelmaksi muodostui lukuisten transistorien liittäminen yhteen

- › Mikropiiri jossa samalle piirille tehdään transistorit, passiiviset komponentit ja johdotus
- › Samaan aikaan sekä Texas Instruments että Fairchild Semiconductor
 - Pitkällisiä patenttikiistoja... samoin tietokoneen toimintaperiaatteesta



J. Kilbyn (Texas Instruments) ensimmäinen IC-piiri, 1958

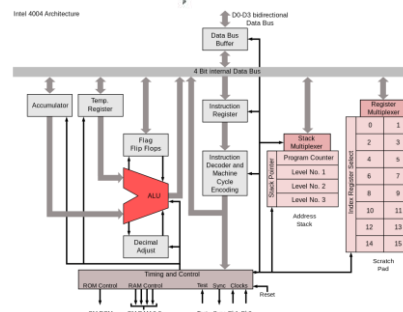
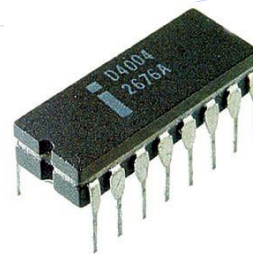
ELEC-C7110

21

Mikroprosessori

» Intel 4004

- › 1971
- › Alun perin Busicom Corp. (Japani) yhtiölle taskulaskinta varten
- › Kellotaajuus 740 kHz
- › 46300 tai 92600 käskyä/s
- › 16 pin
- › 46 käskyä
- › 16 rekisteriä á 4 bittiä



ELEC-C7110

22

Henkilökohtainen tietokone

DEC:n pääjohtaja 1974:
Ei mitään syytä olettaa,
että ihmiset haluaisivat
oman tietokoneen

2057 – 2019 = 2019 - 1981
Mikä muuttuu ja mikä ei?



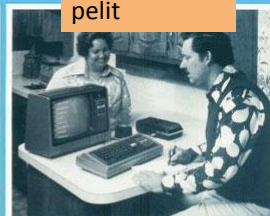
1981

Toimisto:
Paperityö ja
suunnittelu



BUSINESS: Significantly reduces the time spent on tedious paperwork and serves as a planning aid.

Koti:
Pankkiasiat ja
pelit



HOME: Computes budgets and bank balances, stores data, entertains with exciting games.

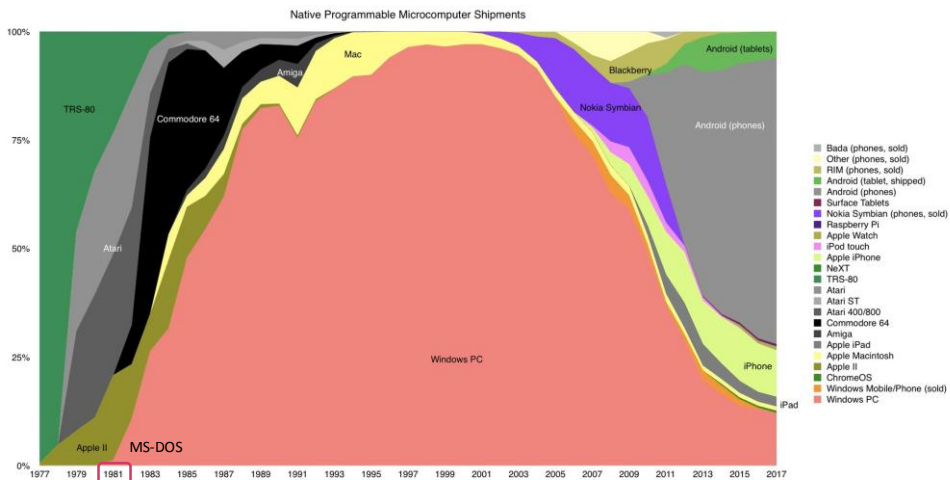


Opetus:
Opiskelijatiedot ja
matematiikan opetus



Laboratorio:
Edullinen
tietojenkäsittely

Käyttöjärjestelmien markkinaosuudet



Tulevaisuus = kvanttilaskenta – vai mitä?



Mikko Hyppönen @mikko · 19 min

Interesting point raised in a reddit thread:

Satoshi's original bitcoins are now a quantum canary. Once we see them moving, we'll know that someone has a functioning advanced quantum computer. It's just too big a prize not to be the first thing you'd do with a quantum computer.



25

Tietojenkäsittelyn vaikutus työelämään



1. Jos kehitys jatkuu entisellään, tietojenkäsittelyn tehokkuus (per €) kasvaa nykyisestä yli miljardikertaiseksi vuoteen 2057 mennessä. Miten tämä muuttaa työelämää?

Keskustelu!

Lukekaa myös

Martti Mäntylä: Big data, IoT, lohkoketju, tekoäly ja koneoppiminen: Post-digitaalista elämää määrittävät taianomaiset kyvyt

<https://www.tekniikkatalous.fi/tekniikka/ict/big-data-iot-lohkoketju-tekoaly-ja-koneoppiminen-post-digitaalista-elamaa-maarittavat-taianomaiset-kyvyt-6704315>

Suomi

- » 1958: Ensimmäinen (toimiva) tietokone
 - › IBM 650, Postisäästöpankki
- » 1960: 8 tietokonetta
- » 1968: 155 tietokonetta
 - › niistä 58 IBM System/360
- » 1977
 - › TKK hankki DEC-20
 - Ensimmäinen **osituskäyttö!**
 - › Noin 1 MIPS
(million instructions per second)



ELEC-C7110

27

Moderni tietokone

Raspberry Pi 2

- Neljä ydintä
- Muisti: 1 GB
- Kellotaajuus: 900 MHz
- Tehonkulutus: 5W
- Hinta (2015) 31 €

~ 1000 kertaa tehokkaampi kuin DEC-20




ELEC-C7110

28

ENIAC vs. TRS-80 vs. Raspberry


ENIAC

- » 1947
- » 160 kW
- » 20 * 10 numeron desimaaliluku
- » 5 kHz
- » 6 000 000 \$
- » 30 000 kg



TRS-80 III

- » 1980
- » 200 W ?
- » 48 kB
- » 2 MHz
- » ~ 3000 €
- » ~ 10 kg



Raspberry Pi 2

- » 2014
- » 5 W
- » 1 GB
- » 900 MHz
- » 31 €
- » ~ 0.2 kg



ELEC-C7110

29

1 Gigaflop



Cray X-MP	DEC AlphaServer	Power Mac G4	G4 Cube
1982	1995	1999	2000
7,000,000 cm ³ (= 7 m ³)	100,000 cm ³ (= 0,1 m ³)	40,000 cm ³	8,000 cm ³

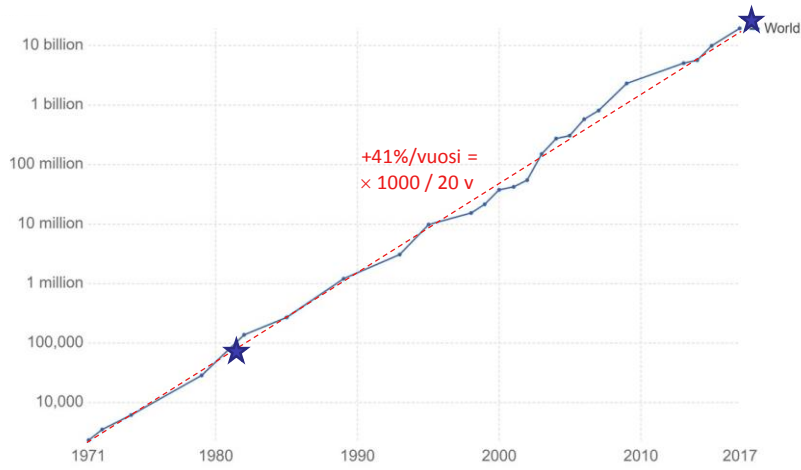
PowerBook G4	iPhone	iPad	Apple Watch
2001	2007	2010	2015
2,200 cm ³	85 cm ³	390 cm ³	14 cm ³



ELEC-C7110

30

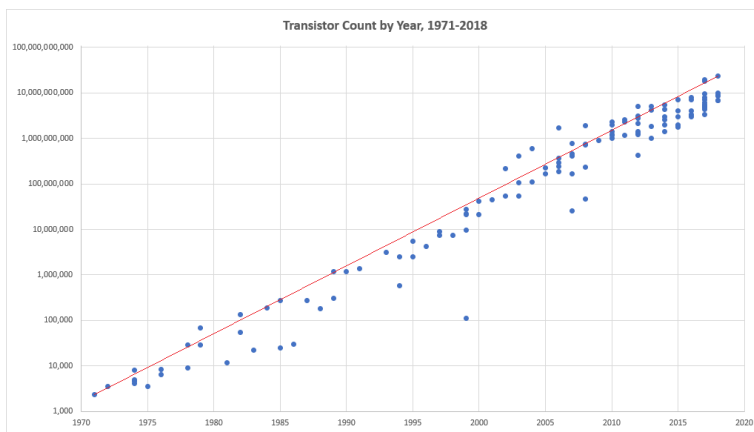
Mooren laki Transistoreja / mikropiiri



ELEC-C7110

31

⇒ 2018

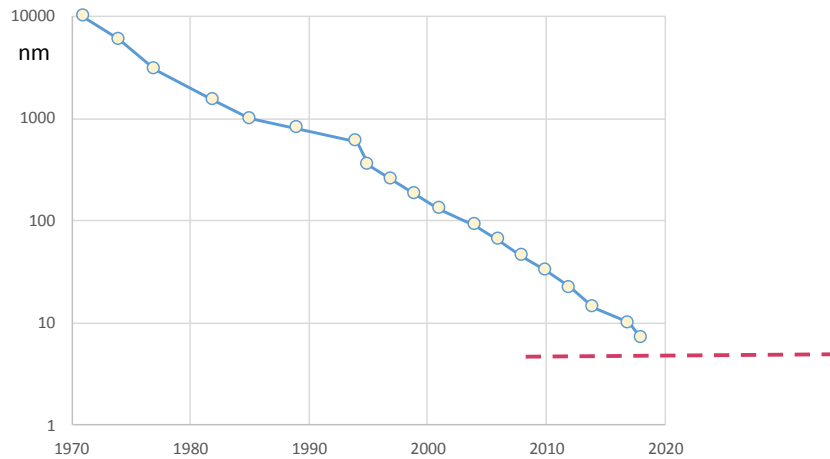


<https://hackernoon.com/moores-law-is-alive-and-well-adc010ea7a63>

ELEC-C7110

32

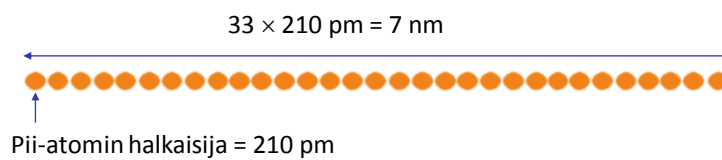
Mikropiirien viivanleveyden kehitys



ELEC-C7110

33

7 nm viivanleveys mahdollinen



Pii-atomeja mahtuu 1 cm:lle 48 miljoonaa

Jos pii-atomin halkaisija olisi 1 cm,
niin vastaava määrä pii-atomeja veisi 476 km

ELEC-C7110

34

Tulevaisuus (M. Muller, S. Yeric / Arm)



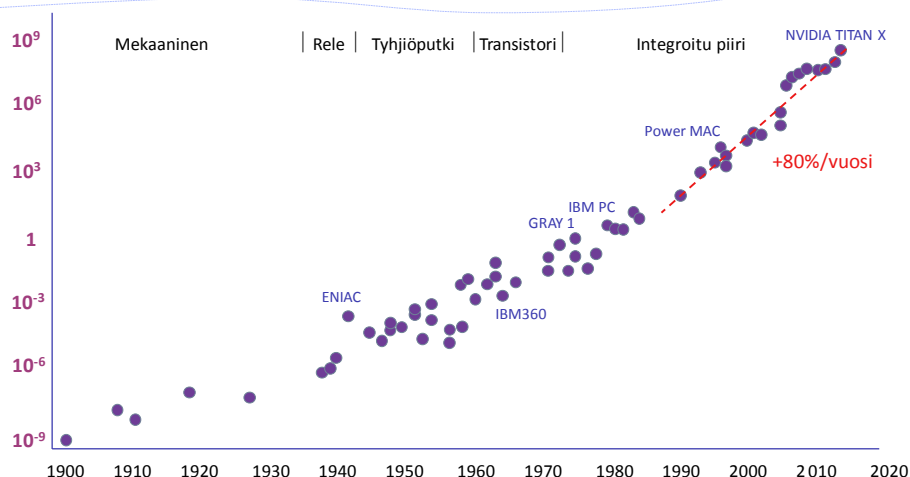
<https://www.techrepublic.com/article/moores-law-is-dead-three-predictions-about-the-computers-of-tomorrow/>

- » **3D chips will continue to improve processor performance.**
 - › Muller believes chip designers will continue to squeeze more power from processors by stacking more transistors and processor dies on top of each other.
- » **Computers will rely on increasingly specialized chips**
 - › "The Google TPU is a great example. You don't need massive amounts of accuracy to do these calculations. You can cut back on power and delay by not counting as many bits in the decimal points. Simple things like that."
- » **Computers will move beyond silicon chips**
 - › "We're going to start to see accelerators that are hardware differentiated," says Yeric.
 - › "So, you can have a special kind of transistor that does one thing really well, [which] doesn't necessarily do all compute well, and you can have a chip that bolts onto a regular CMOS chip."

ELEC-C7110

35

Laskutoimituksia/s/\$

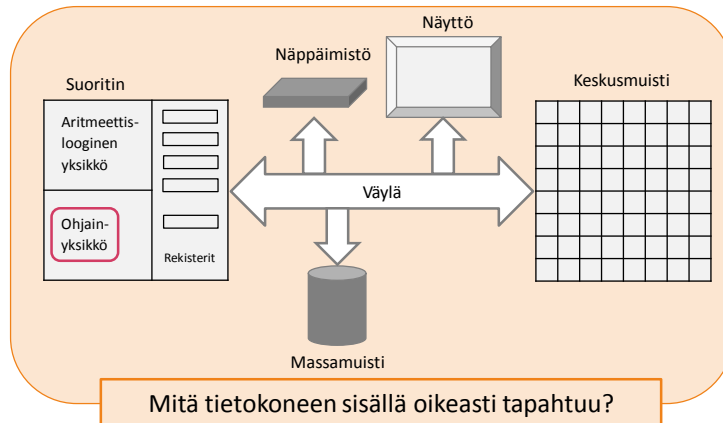


ELEC-C7110

36

Tietokoneen rakenneosat

Mitkä ovat **tietokoneen** keskeisimmät **rakenneosat** ja mitkä ovat niiden tehtävät?



ELEC-C7110

37

Ohjausyksikön tehtävät

- » **Lue** seuraava käsky ohjelmalaskurin ilmoittamasta paikasta
- » **Muunna** numeerinen koodi signaaleiksi muille osille
- » **Kasvata** ohjelmalaskurin arvo osoittamaan seuraavaa käskyä
- » **Lue** data muistista tai oheislaitteesta
- » **Lähetä** data aritmeettis-loogiseen yksikköön (ALU) tai rekisteriin
- » **Ohjaa** (tarvittaessa) muita yksiköitä suorittamaan vaaditut operaatiot
- » **Kirjoita** ALU:sta saatu tulos muistiin, rekisteriin tai tulostuslaitteeseen
- » **Palaa** takaisin ensimmäiseen kohtaan

ELEC-C7110

38

Mutta...

mitä **lue, muunna ja kasvata** tarkoittavat tietokoneen sisällä?

Vastaus

- » Aritmeettisiä ja
- » loogisia toimituksia
 - › Jotka ovat olennaisesti sama asia = bittien manipulointia tarkasti määriteltyjen sääntöjen mukaan

NAND-funktio

Luonnollinen kieli purettuna osiin

Lue, muunna, kasvata

F on epätosi vain jos sekä a että b ovat tosia

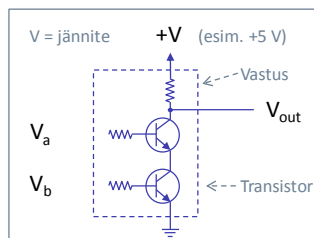
Booleen algebra

$$F = \neg(a \wedge b)$$

Looginen portti



Toteutus mikro-piirillä



Ulostulojännitteiden säätäminen sisään tulevien jännitteiden mukaan

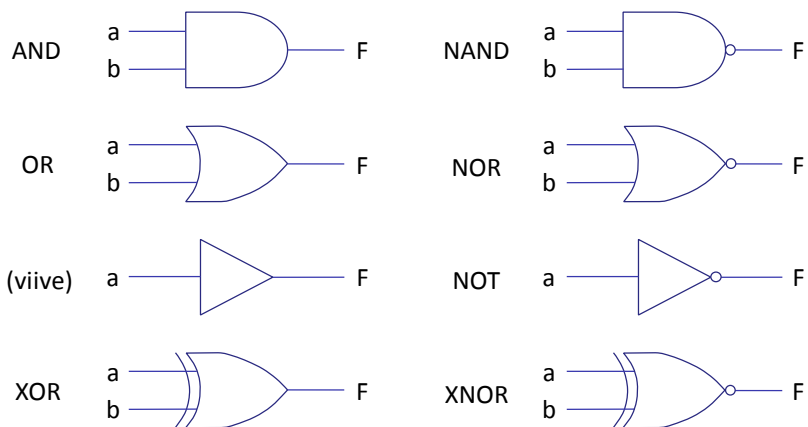
Fyysinen todellisuus

Sähkömagneettiset ja kvanttimekaaniset ilmiöt

Boolean algebra

a	b	Epätosi FALSE	JA AND \wedge	XOR \oplus	TAI OR \vee	NOR \downarrow	XNOR	NAND tai \uparrow	Tosi TRUE
0	0	0	0	0	0	1	1	1	1
0	1	0	0	1	1	0	0	1	1
1	0	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1

Loogiset portit



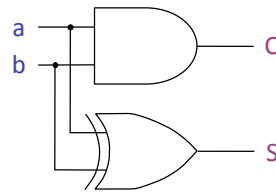
Yksinkertainen toimitus: kahden bitin (a ja b) yhteenlasku



in		out	
a	b	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$$S = a \oplus b$$

$$C = a \wedge b$$



ELEC-C7110

43

Useampi-bittisen luvun yhteenlasku



- » 10-järjestelmä: $3 + 10 = 13$
- » Binääri: $0011 + 1010 = 1101$
 - > Alin bitti: $1 + 0 = 1$
 - > Toinen bitti: $1 + 1 = 1$ 0
 - > Kolmas bitti: $1 + 0 + 0 = 1$
 - > Neljäs bitti: $0 + 1 = 1$

- » Tarvitaan siis 3 bitin yhteenlasku
 - > $z + a + b$, jossa z on alemman bitin "muistiin" laitettu bitti

ELEC-C7110

44

3 bitin yhteenlasku

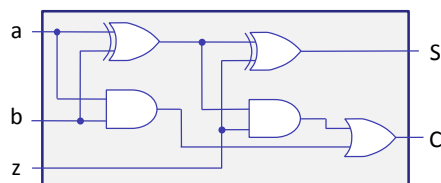
in			out	
z	a	b	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Eräs toteutus Boolean algebralla

$$S = (\neg z \wedge (a \oplus b)) \vee (z \wedge \neg(a \oplus b))$$

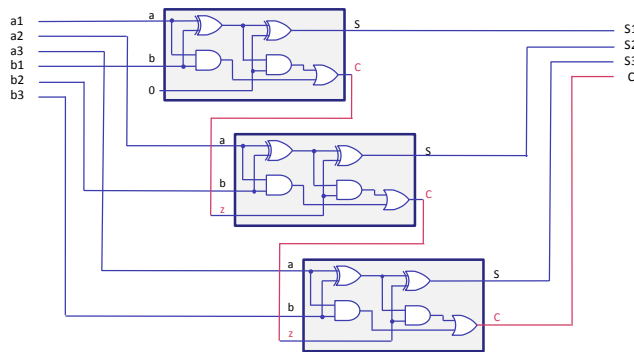
$$C = (\neg z \wedge (a \wedge b)) \vee (z \wedge (a \vee b))$$

Yksinkertainen toteutus!



Yhdistämällä näitä moduuleita voidaan periaatteessa laskea yhteen pitkiä lukuja (C yhdistetään seuraavan moduulin z:aan)

Moduulien yhdistäminen



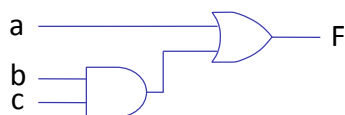
ELEC-C7110

47

Loogiset operaatiot

1. Miten looginen operaatio $F = (a \vee b) \wedge (a \vee c)$ voidaan toteuttaa loogisilla piireillä mahdollisimman yksinkertaisesti? Piirrä kuva!

a	b	c	$a \vee b$	$a \vee c$	$(a \vee b) \wedge (a \vee c)$
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	1	1	1
1	0	0	1	1	1
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	1	1	1



ELEC-C7110

48

Negatiiviset luvut



- » Negatiiviset luvut pitää esittää bitteinä
 - › Koska etumerkkiä (-) ei ole
- » Ratkaisu: Kahden komplementti
 - › Käännetään luvun kaikki bitit ($0 \rightarrow 1, 1 \rightarrow 0$) ja
 - › lisätään saatuun lukuun 1
- » Esimerkki -5 neljän bitin järjestelmässä
 - › $5 = 0101$
 - › Bittien käänös $\rightarrow 1010$
 - › Lisätään yksi $\rightarrow 1011$ (= -5)
- » Vähennyslasku samalla logiikalla kuin positiivisilla luvuilla

ELEC-C7110

49

Esimerkki $2 - 6 = -4$

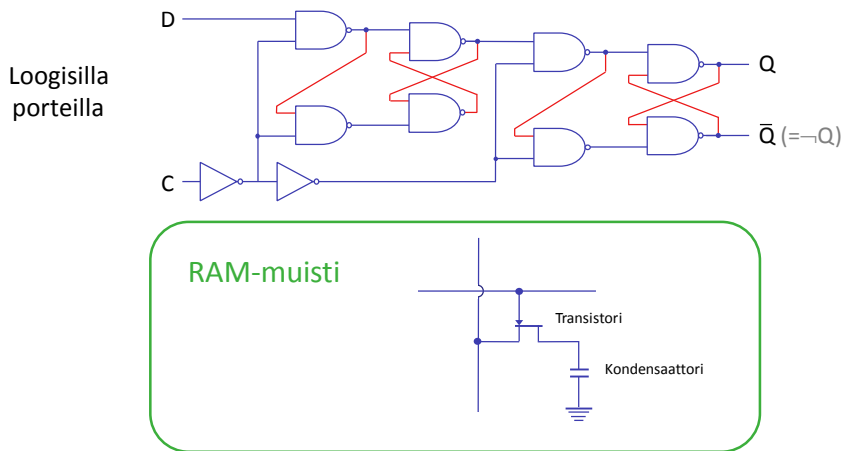


- » $2 = 0010$
- » $6 = 0110$
- » Muunnos negatiiviseksi ($6 \rightarrow -6$)
 - › $0110 \rightarrow 1001$
 - › $1001 + 1 = 1010$ (= -6)
- » Lasku
 - › $0010 + 1010 = 1100$ (= -4)
- » Tarkistus
 - › $1100 - 1 = 1011 \rightarrow 0100$ (= 4)

ELEC-C7110

50

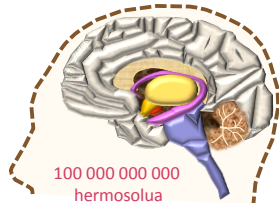
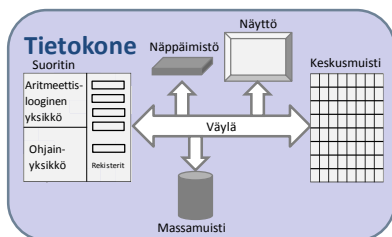
1 bitin muisti



ELEC-C7110

51

Tietokone-aivot-analogia



100 000 000 000
hermosolua
á 10 000 synapsia
10¹⁵ "liitosta"
kellotaajuus ~ 1kHz

Muistin "osat"

- Aistimuisti ~ 1 s
- Työmuisti (lyhytkestoinen)
 - ~ 7 numeroa tai 6 kirjainta tai 5 sanaa
- Pitkäkestoinen muisti (säilömuisti)

Aivot

Muisti on assosiativista (ei osoitteita)

Aritmeettis-looginen yksikkö hidas ja rajallinen

Aivot tehokas "hahmontunnistuksessa"
erityisesti sosiaalisten suhteiden havainnoinnissa

Ohjainyksikkö ("minä") itsenäinen
tai näin me ainakin kuvittelemme

Väylät paljon hajautetumpia

Silmien hermosolut tekevät esiprosessointia
Väylä silmistä aivoihin ~ 10 Mbit/s
Tietoinen havaintokyky ~ 40 bit/s

ELEC-C7110

52

Luentotehtävä / MyCourses

Mitä ohjelmointikieliä osaat nyt?

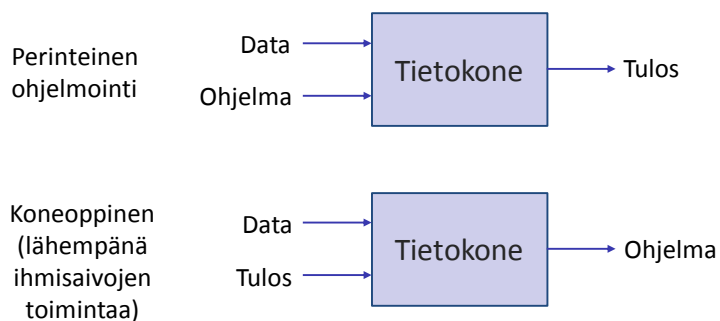
Arvioi osaamisesi asteikolla:

1. Vain perusteet
2. Vähintään kohtuullinen joka riittää ainakin suppeiden ohjelmointitehtävien itsenäiseen toteuttamiseen

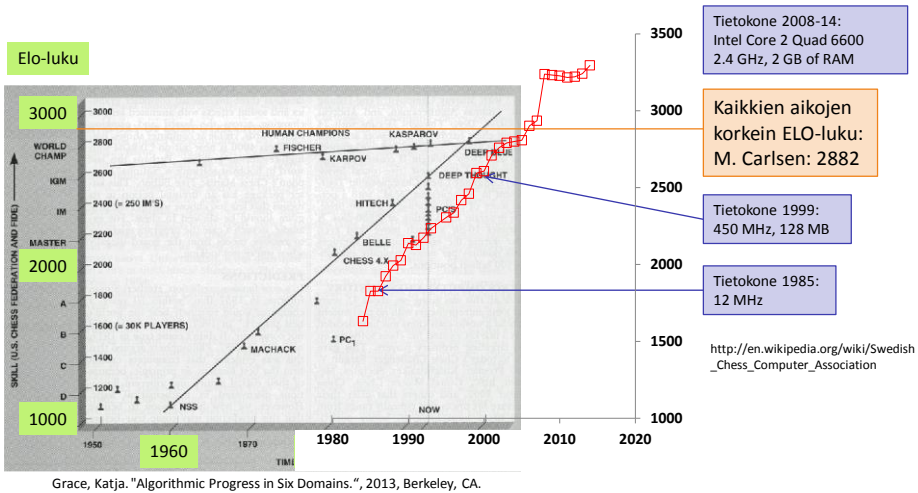
Entä tulevaisuudessa?

3. Minkä ohjelmointikielten opiskeluun uskot panostavasi seuraavan 5 vuoden aikana (mainitse 1 - 3 kieltä)

Perinteinen vs. koneoppiminen



Tietokoneiden shakkiohjelmät



ELEC-C7110

55

Go



Googlen DeepMind ohjelma voitti Go-pelin Euroopan mestarin lokakuussa 2015
Ja maailmanmestari Lee Sedolin maaliskuussa 2016

Ohjelma perustuu täysin hahmontunnistukseen!

ELEC-C7110

56

Tulevaisuus?



ELEC-C7110

57

Ohjelmoinnin alku

- » Ada Lovelace
 - › 1842: Idea ohjelmitavasta tietokoneesta ja algoritmeista
- » ENIAC
 - › 1949: Betty Holberton & John Mauchly C-10: "ohjelmointikielten prototyyppi"
- » Univac
 - › 1952: Grace Murray Hopperin ryhmä kehitti A-0:n, jossa konekielen joukkoon voitiin sijoittaa "makroja"
 - › 1955: A-2: koneen omaa kieltä ei tarvinnut enää lainkaan



Enemmän Hopperista (video n. 50 min - suosittelen!):
<http://videosift.com/video/Mother-of-Computer-Science-Grace-Hopper-and-Kurt-Beyer>

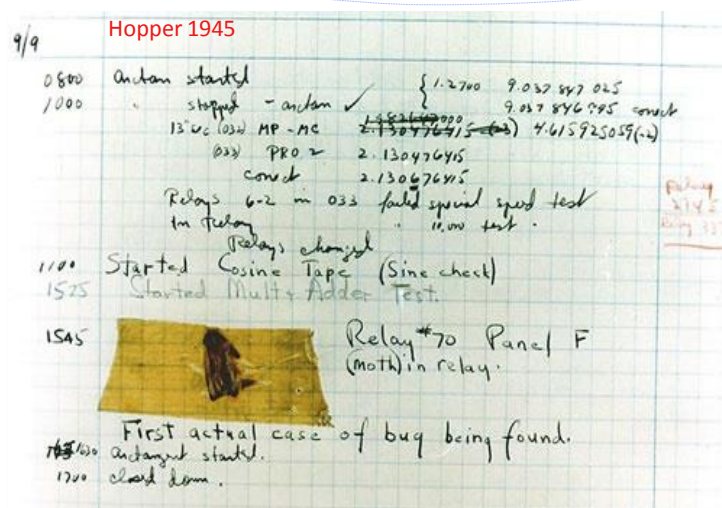
ELEC-C7110

58

Fortran

- » 1953 ehdotus uudesta ohjelmointikielestä (John Backus, IBM)
- » Kehityksen aikataulu venähti
 - › 1955: piti valmistua muutamassa kuukaudessa
 - › 1957: ensimmäinen buginen versio
- » Ja kova usko ohjelmoinnin helppouteen [IBM raportti, 1954]
 - › "After an hour course in FORTRAN notation, the average programmer **can fully understand** the steps of a procedure stated in FORTRAN language without any additional comments."
 - › "Since FORTRAN **should virtually eliminate coding and debugging**, it should be possible to solve problems for less than half the cost that it would be required without such a system."
 - › Saavutti (ongelmista huolimatta) suuren suosion (samoin Cobol)
- » Fortran 66 (1966) ensimmäinen standardoitu ohjelmointikieli

Ensimmäinen tunnettu bugi



Margaret Hamilton, lead software engineer of the Apollo Project, stands next to the code she wrote by hand and that was used to take humanity to the moon. [1969]



ELEC-C7110

61

Koodien vertailu



A-2

1.	GMM000	000001	
	000180	020216	10.0 = i
	1RG000	001000	
2.	AM0034	034040	$a_i^2 = t$
3.	RNA040	010040	$\sqrt{t} = t'$
4.	APN034	012038	$a_i^2 = y$
5.	AM0002	038038	$5y = y'$
6.	AA0040	038038	$t' + y' = y''$
7.	AS0004	038040	$400 - y'' = t''$

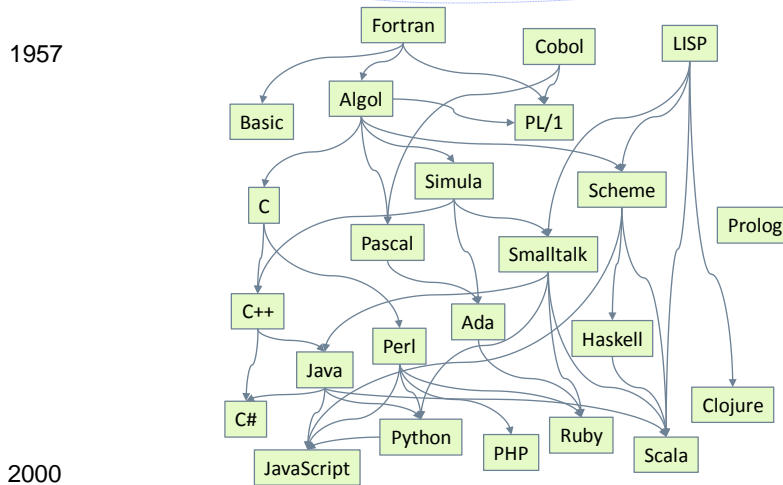
Fortran

```
1      DIMENSION A(11)
2      READ A
3      DO 3,8,11 J=1,11
4      3 I=11-J
5      Y=SQRT(ABS(A(I+1)))+5*A(I+1)**3
6      IF (400 >= Y) 8,4
7      4 PRINT I,999.
8      GO TO 2
9      8 PRINT I,Y
10     11 STOP
```

ELEC-C7110

62

Ohjelmointikielten ”pensas”



ELEC-C7110

63

Ohjelmointikielten tasot

1. Miten ohjelmointikielten eri tasot eroavat toisistaan ja miksi ylipäätään on kehitetty eri tason kieliä?

Taso	Esimerkkejä
Ohjelmistokehykset (software framework)	Ajax framework
Täsmäkielet (domain-specific languages)	Mathematica, SQL
Korkean tason kielet (high level languages)	Python, Perl, Ruby
Keskitaso kielet (middle-level languages)	C, C++
Symbolinen konekieli (assembly language)	Assembler
Konekieli (machine language, machine code)	

Mutta miksi on kehitetty niin erilaisia kieliä?

kun luonnolliset kielet (suomi, englanti, ...) ovat kaikki samalla tasolla!
mutta onhan meillä matematiikan kieli erikseen – onko muita!

ELEC-C7110

64

Kysely 2016-18

Kieli	Opettelu (%)		
	2016	2017	2018
C	34	30	27
C++	28	25	20
Python	23	17	19
Java	5	17	21
Javascript	3	2	3
Scala	2	4	6
C#	2	2	2
PHP	1	0	1
Matlab	1	4	0

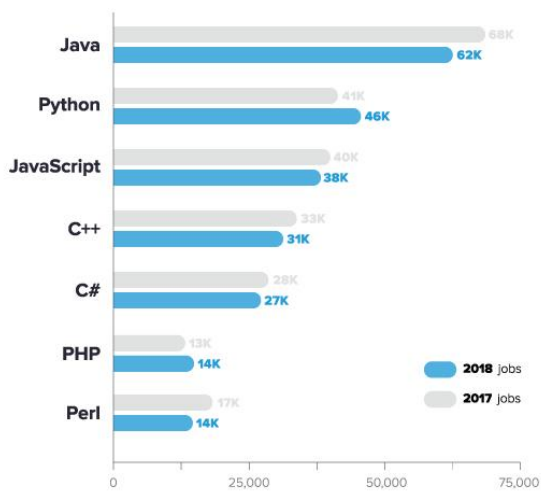
ELEC-C7110

65

Tarve?

Job postings containing top languages

Indeed.com - November, 17th 2017



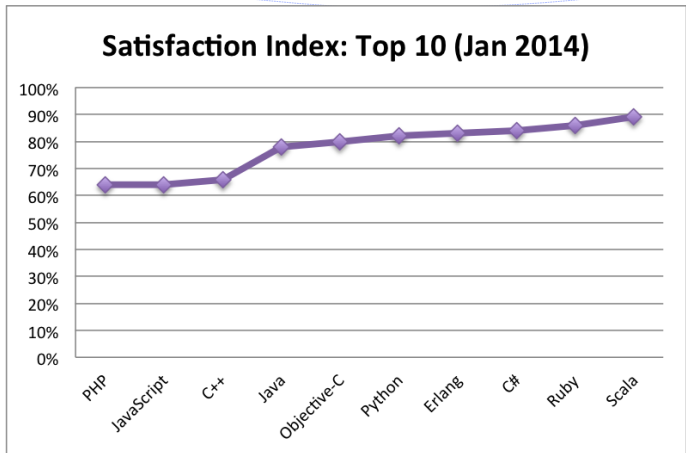
ELEC-C7110

66

Best Programming Languages to Learn

1. JavaScript
2. Python
3. C#
4. Java
5. PHP
6. Go
7. Swift
8. Rust
9. Kotlin
10. C & C++

Tyytyväisyys



<http://intersog.com/blog/programming-languages-rating-as-of-january-2014>

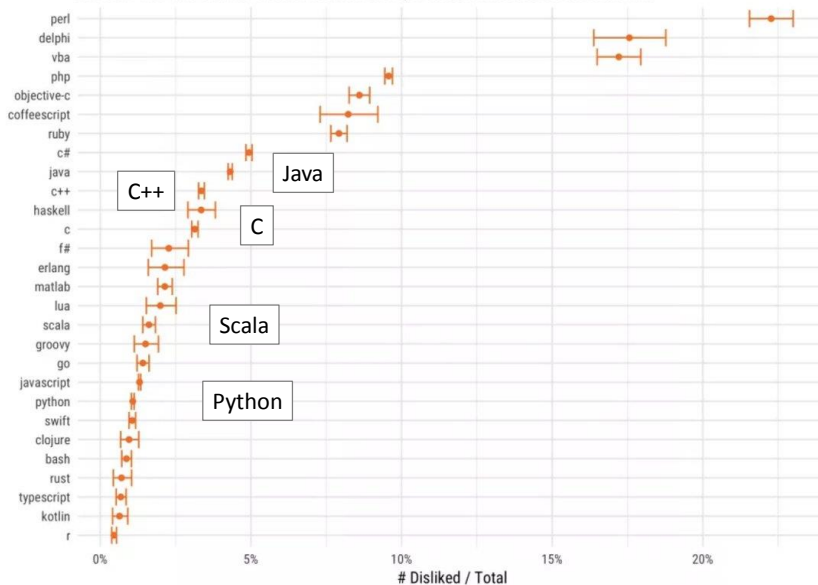
ELEC-C7110

67

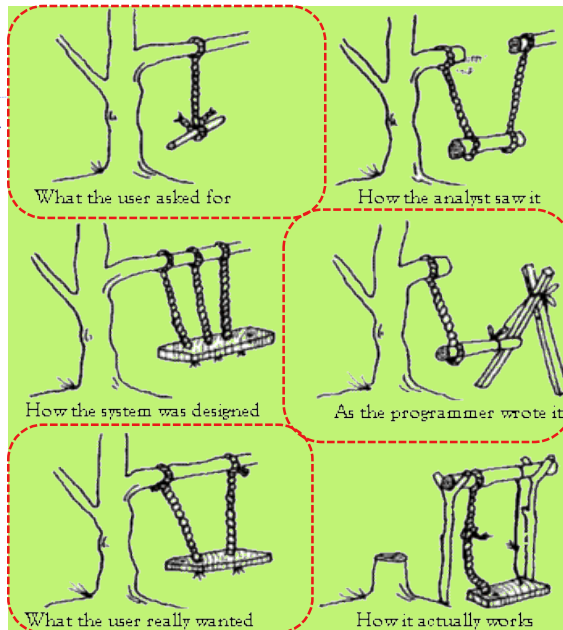
Perl is the most hated programming language

How disliked is each programming language?

Based on "likes" and "dislikes" on Stack Overflow Developer Stories. Includes 95% credible intervals



Oma
kokemus
1988-
1990
Pascal



ELEC-C7110

69

A?
Aalto-yliopisto

Hamming-koodaus

A?
Aalto-yliopisto

» Miksi?

- › Numerot ovat tärkeitä, koska informaatio esitetään ja käsitellään numeroina
- › Yksikin virhe voi olla kriittinen – tietokonelaskelmien pitää olla täsmälleen oikein (toisin on aivoissa, joissa sallitaan epätarkkuutta)
- › Virheitä aiheuttavat mm. suurenergiset kosmiset hiukkaset

ELEC-C7110

70

Hamming

- » [7/4] –koodaus (periaate yleistyy loputtomiin...)
 - › 4 databittiä (d1 – d4) + 3 pariteettibittiä (p1 – p3)

Bitti	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
	p1	p2	d1	p3	d2	d3	d4	p4	d5	d6	d7	d8	d9	d10	d11	p5	d12	d13
p1	x		x		x		x		x		x		x		x		x	
p2		x	x			x	x			x	x			x	x		x	x
p3				x	x	x	x					x	x	x	x			
p4								x	x	x	x	x	x	x	x			
p5																x	x	x

- » Jos vain yksi bittimuutos, virhe voidaan korjata

Esimerkki 1

- » 7/4 koodin bitit ovat 0101010
 - › Onko koodissa virhe?
- » Pariteetilaskenta (punaiset)
 - › **0101010** : parillinen ok
 - › **0101010** : parillinen ok
 - › **0101010** : parillinen ok

Esimerkki 2

- » 7/4 koodin bitit ovat 1011101
 - › Onko koodissa virhe?
- » Pariteettilaskenta (punaiset)
 - › **1011101** : **parillinen ok**
 - › **1011101** : **parillinen ok**
 - › **1011101** : **pariton virhe**

↑
Jos vain yksi virhe, niin sen täytyy olla tässä (p3)
- koska muiden bittien täytyy olla oikein

- › **1010101** = **oikea**

Esimerkki 3

- » 7/4 koodin bitit ovat 0101011
 - › Onko koodissa virhe?
- » Pariteettilaskenta (punaiset)
 - › **0101011** : **pariton virhe**
 - › **0101011** : **pariton virhe**
 - › **0101011** : **pariton virhe**

↑
Jos vain yksi virhe, niin sen täytyy olla tässä (d4)
- koska tämä on ainoa, jossa yksi virhe johtaa virheeseen jokaisella rivillä

- › **0101010** = **oikea**