

Programmeerimine

1.loeng

Varmo VENE
Arvutiteaduse Instituut
Tartu Ülikool
EMAIL: varmo.vene@ut.ee

Täna loengus

- Administratiivinfo
- Definiitsioonid: programmeerimine, programm, arvuti
- Ajalugu
- Programmeerimiskeeled
- Translaatorid
- Python

Administratiivinfo

Üldinfo

- MTAT.03.100 6 ainepunkti.
 - Loengud (32 tundi)
 - Arvutipraktikumid (32 tundi)
 - Iseseisev töö (92 tundi)
- Kursuse veebileht:
<https://courses.cs.ut.ee/2014/programmeerimine/>
- Mitteavalikud materjalid *Moodle*'i keskkonnas:
<http://moodle.ut.ee/>

Administratiivinfo

Eesmärgid

Kursuse eesmärgiks on anda alusteadmised programmeerimise põhikonstruktsioonidest ning esmased oskused algoritmide ja programmide koostamiseks.

Õpiväljundid

Kursuse läbinud üliõpilane:

- oskab kasutada põhilisi programmeerimiskonstruktsioone: muutuja, avaldis, omistuslause, tingimuslause, tsükel, alamprogramm, rekursioon, andmevahetus kasutaja ja failidega;
- tunneb põhilisi andmetüüpe ja -struktuure (täis- ja ujukomarvud, tõeväärtused, sõned, järjendid) ning oskab kasutada vastavaid standardoperatsioone;
- oskab analüüsida ja üksikasjalikult selgitada programmi töö käiku ning programmi modifitseerida ja laiendada;
- oskab luua lihtsamale ülesandele vastava algoritmi, koostada ja korrektselt vormistada lahendusprogrammi ning seda siluda ja testida.

Administratiivinfo

Hinde kujunemine

- Hinde kujuneb praktikumides ja eksamil kogutud punktide summana.
- Praktikumid (kuni **30 punkti**):
 - kontrolltöö (arvestus)
 - praktikumides osalus ja koduülesanded (10 punkti)
 - tärnülesanded (10 punkti)
 - projekt (10 punkti)
- Eksam (kuni **80 punkti**)
 - 2 vaheeksamit (20 + 20 = 40 punkti)
 - lõputest (40 punkti)
- **NB!** Enne vaheeksameid peab **kontrolltöö** olema **arvestatud**.
- **NB!** Igast eksamiosast tuleb saada **vähemalt 50%**.

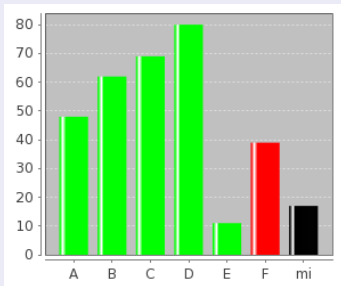
Administratiivinfo

Hindeskaala

	Hinne	Punkte
A	suurepärase	≥ 91
B	väga hea	81 – 90
C	hea	71 – 80
D	rahuldav	61 – 70
E	kasin	51 – 60
F	puudulik	0 – 50

Administratiivinfo

Eelmise aasta hinnete jaotus



Administratiivinfo

Eritingimused

- Vabariikliku informaatikaolümpiaadi lõppvoorust *edasi- jõudnute rühmas* osavõtjad saavad soovi korral aine arvestatud:
 - täpsem info varmo.vene@ut.ee.

Administratiivinfo

Ajakava

4	25.09 / 26.09	kontrolltöö
9	30.10 / 31.10	1. vaheksam
15	11.12 / 12.12	projekt esitus
	05.01 / 13.01	2. vaheksam
	08.01 / 16.01	lõputest
	21.01 / 22.01	järelvaheksamid
	26.01	järellõputest

Kirjandus

Põhiõpik

- Veebiõpik kursuse kodulehel

Pythoni raamatuid

- J. Elkner, P. Wentworth, A. Downey, C. Meyers. *How to Think Like a Computer Scientist: Learning with Python*. 3rd edition.
- J. Zelle. *Python Programming: An Introduction to Computer Science*. 2nd edition. 2010.

Mis on programmeerimine?

e-teatmik (www.vallaste.ee)

- **programmeerimine** Programmide kirjutamine.
- **programm** Organiseeritud käsujada, mis täitmisel põhjustab arvuti käitumist etteantud viisil. Ilma programmideta on arvutid kasutud.
- **arvuti, raal, kompuuter** Programmeeritav masin.

Mis on programmeerimine?

e-teatmik (www.vallaste.ee)

- **programmeerimine** Programmide kirjutamine.
- **programm** Organiseeritud käsujada, mis täitmisel põhjustab arvuti käitumist etteantud viisil. Ilma programmideta on arvutid kasutud.
- **arvuti, raal, kompuuter** Programmeeritav masin.

Mis on programmeerimine?

e-teatmik (www.vallaste.ee)

- **programmeerimine** Programmide kirjutamine.
- **programm** Organiseeritud käsujada, mis täitmisel põhjustab arvuti käitumist etteantud viisil. Ilma programmideta on arvutid kasutud.
- **arvuti, raal, kompuuter** Programmeeritav masin.

Mis on programmeerimine?

e-teatmik (www.vallaste.ee)

- **programmeerimine** Programmide kirjutamine.
- **programm** Organiseeritud käsujada, mis täitmisel põhjustab arvuti käitumist etteantud viisil. Ilma programmideta on arvutid kasutud.
- **arvuti, raal, kompuuter** Programmeeritav masin.

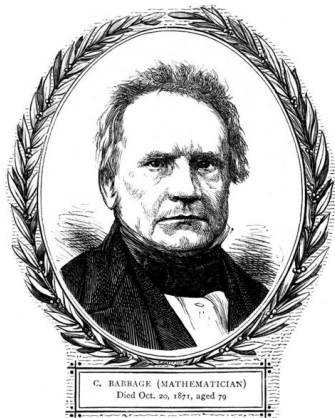
Mis on programmeerimine?

Vikipeedia (*et.wikipedia.org*)

- **Programmeerimine** on arvutiprogrammide lähtekoodi kirjutamise, testimise, silumise ja haldamise tegevuste jada.
- **Arvutiprogramm** (tavaliselt lihtsalt "programm") on arvutile arusaadav käskluste kogum.
- **Arvuti** on mehhaaniline või elektrooniline seade, mis suudab hõivata, töödelda ja väljastada andmeid vastavalt etteantud reeglitele; masin, mille otstarbeks on tehete sooritamine.

Babbage'i analüütiline masin

- 1837. a. esitas **Charles Babbage** (1791–1871) **analüütilise masina** kirjelduse üldiste matemaatiliste võrrandite lahendamiseks.
- Sisuliselt oli esimene universaalne arvuti, millest füüsiliselt valmis siiski ainult osaline prototüüp (1910).
- Masin kasutas jõuallikana aurumootorit ning koosnes "**veskist**" (*mill*) ja "**laost**" (*store*).
- Laos sai hoida kuni 1000 arvu (igauks kuni 50 kümnendkohta).
- Veski oli võimeline teostama aritmeetlisi ja võrdlustehteid ning arvutama ruutjuurt.



Babbage'i analüütiline masin

- Analüütilist masinat juhiti perfokaartide abil.
- Kasutati kolme liiki perfokaarte: aritmeetiliste operatsioonide, arvkonstantide ning veski-lao vahelise suhtluse tarvis.
- "Programmeerimiskeel" võimaldas kasutada nii hargnemis- kui tsükli-konstruktsioone.
- 1843. a. esitas **Ada Lovelace** (1815-1852) kirjelduse, kuidas arvutatada analüütilise masinaga Bernoulli arve.
- Antud kirjeldust peetakse esimeseks arvutiprogrammiks.



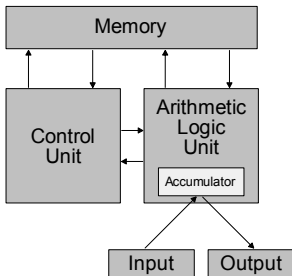
Turingi masin

- 1936. a. esitas **Alan Turing** (1912-1954) idealiseeritud universaalse arvutusmasina kirjelduse arvutatavusega seotud teoreetiliste küsimuste uurimiseks.
- Tänapäeval kasutatakse Turingi masinaid algoritmiteoorias, keerukusteoorias ja mitmetes teistes teoreetilise arvutiteaduse harudes.



Von Neumanni arhitektuur

1945. a. esitas **John von Neumann** (1903-1957) arvutiarhitektuuri, millel baseeruvad praktiliselt kõik tänapäeva arvutid. Oluliseimaks uudseks ideeks oli, et mälus ei paikne mitte ainult andmed, vaid ka programm.



Programmeerimiskeeled

- **Programmeerimiskeel**: programmide kirjutamiseks mõeldud tehiskeel.
- On defineeritud **süntaksi-** ja **semantikareeglite** abil, mis määravad vastavalt programmide struktuuri ja tähenduse.
- Programmeerimiskeelte põlvkonnad:
 - 1. põlvkond – **masinkood**;
 - 2. põlvkond – **assemblerkeeled**;
 - 3. põlvkond – **kõrgtasemekeeled**;
 - ...

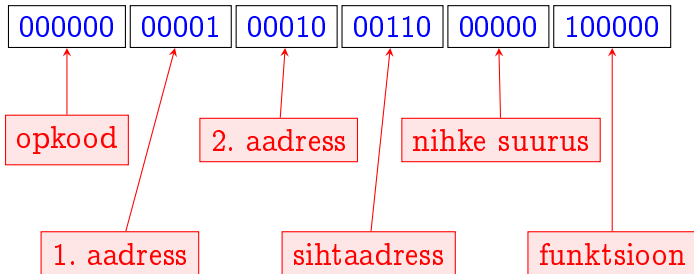
Masinkood

- Arvutis on kõik andmed binaarkujul ...
- ... see kehtib ka masinkoodis programmide kohta!
- Näide: MIPS arhitektuuri masinkoodi käsk "liita aadressitel 1 ja 2 olevad arvud ning salvestada resultaat aadressile 6"

0000000001000100011000000100000

Masinkood

- Arvutis on kõik andmed binaarkujul ...
- ... see kehtib ka masinkoodis programmide kohta!
- Näide: MIPS arhitektuuri masinkoodi käsk "liita aadressitel 1 ja 2 olevad arvud ning salvestada resultaat aadressile 6"

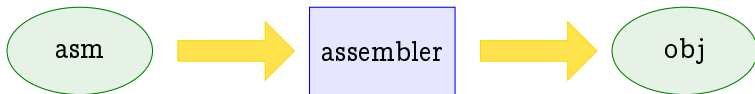


Assemblerkeeled

- Assemblerkeeled kasutavad käskude jaoks mnemoonilisi koode.
- Samuti lubavad kasutada aadressite jaoks nimesid.
- MIPS näide uuesti:

`add $1, $2, $6`

- Kuna arvutid töötavad siiski ainult binaarkoodis programmidega, siis tuleb assemblerkeeles kirjutatud programm transleerida masinkoodi.
- Vastavat translaatorit nimetatakse **assembleriks**.



Kõrgtasemekeeled

- Masinkoodis ja assemblerkeeltes kirjutatud programmid on madalatasemelised: nad on tugevalt platvormist sõltuvad ning neid on suhteliselt keeruline kirjutatada, neist aru saada ja hallata.
- Tavaliselt kasutatakse programmeerimisel nn. **kõrgtasemekeeli**, millede käsustik ja struktuur on abstraktsemad ning lähedasemad inimkeeltele.
- Sõltuvalt arvutusmudelist, millel keel baseerub, liigitatakse kõrgtasemekeeli erinevatesse **paradigmadesse**.
- Käesolevas kursuses vaatleme põhiliselt **imperatiivsesse** paradigmasse kuuluvaid keeli, mis baseeruvad von Neumanni arvutusmudelil.
- Näiteid imperatiivsetest keeltest: Fortran, Cobol, Pascal, C, (Java), (**Python**)

Translaatorid

- Kõrgtasemekeeles kirjutatud programmide täitmiseks tuleb nad **transleerida**.
- Traditsiooniliselt eristatakse kaht liiki transleerimist: **interpreteerimine** ja **kompileerimine**.

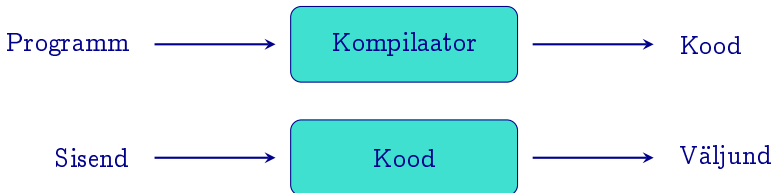
Translaatorid

- Kõrgtasemekeeles kirjutatud programmide täitmiseks tuleb nad **transleerida**.
- Traditsiooniliselt eristatakse kaht liiki transleerimist: **interpreteerimine** ja **kompileerimine**.
- Interpreteerimisel täidetakse programmi iga käsk koheselt pärast ta transleerimist.



Translaatorid

- Kõrgtasemekeeles kirjutatud programmide täitmiseks tuleb nad **transleerida**.
- Traditsiooniliselt eristatakse kaht liiki transleerimist: **interpreteerimine** ja **kompileerimine**.
- Kompileerimisel transleeritakse programm kõigepealt masinkoodi, mida on seejärel võimalik täita.



Python

- Loodi 1991. a. Guido van Rossumi poolt.
- Põhiomadused:
 - multi-paradigmiline;
 - dünaamiliselt rangelt tüübitud;
 - automaatse mäluhaldusega;
 - taandetundlik süntaks;
 - interpreteeritakse läbi baitkoodi.
- Kursuses kasutame Pythoni versiooni 3.



```
# HelloWorld.py
def helloWorld():
    print ("Hello , world!")
```

Järgmiseks korraks

- Registreeruda ainele ÕIS-is (kes veel ei ole) ja võtta osa arvutipraktikumist.
- Lugeda läbi õpiku peatükid:
 - Ptk. 1 "*Sissejuhatus*"
 - Ptk. 2 "*Avaldised ja lihtlaused*"
- Soovi korral osaleda **lisapraktikumis kolmapäeval kell 18.15 ruumis 405**.
- Kui lugemisel tekkis küsimusi, mille kohta soovite järgmises loengus vastust, siis need võib saata hiljemalt **esmaspäeva lõunaks** mailiga **varmo.vene@ut.ee** ja/või **helle.hein@ut.ee**.

Suur tänu osalemast

ja

kohtumiseni!