



TALLINNA
TEHNIKAÜLIKOOL



Infosüsteemide Arendamine I

Loeng 1 – Sissejuhatus

Liisa Jõgiste

Ajakava ja muu oluline

NB! T hakkame 45 min hiljem kui tunniplaanis: kell 9:00

Loeng - Esmaspäev: 14:00-15:30 (1 tund 30 min)

Harjutustunnid - Teisipäev: **09:00**-13:00

(ülesanded) Neljapäev: 11:00-15:30

Konsultatsioon - N 15:30; eelneval kokkuleppel

Loengu materjalid - <https://ained.ttu.ee/>

Kursuse eesmärgid

Kursuse edukalt läbinud üliõpilased

- Tunnevad programmeerimise põhimõisteid ning keelt
- Oskavad kasutada tarkvaraarendaja põhilisi tööriistu (riistvara ja tarkvara)
- On võimelised lahendama programmeerimise ülesandeid
- Oskavad programmeerida C# keeles
- Tunnevad tarkvaratehnika (*software engineering*) alustõdesid ning häid printsiipe (clean code, arhitektuur, mustrid)

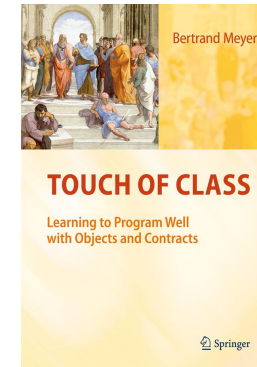
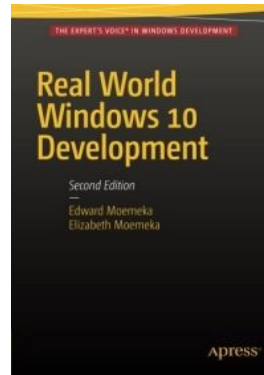
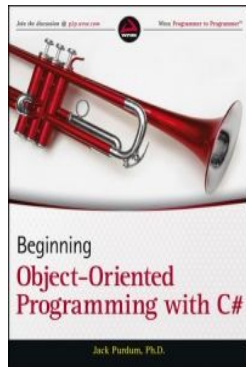
Omandatavad oskused:

- Koodi kirjutamine ja lugemine (olemasoleva taaskasutamine).
- Arvutuslik (*computational*) mõtlemine – kooditükkide kirjutamine
- Oma piiridest ning võimetest aru saamine
- Probleemidele arvutusliku lahenduse leidmine

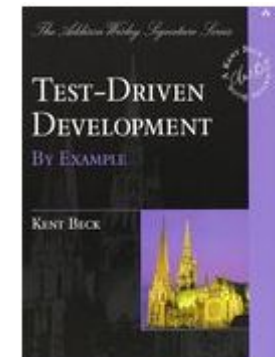
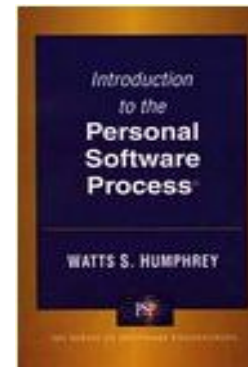
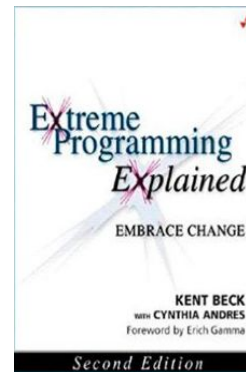
Kõik sõltub mõtlemisest.

Õpikud ja muud abivahendid

- Põhiõpikud



- Lisaõpikud



- Kaks videoloengute sarja

- Kõik lingid on aine kodulehel <https://ained.ttu.ee/>

Hinde kujunemine : 4 komponenti

- Praktiline kontrolltöö (1 tk) – praktikumi ajal
- Iganädalased kodutööd
- Teooriatestid (2 tk) – loengu ajal
- Suuline eksam (eelnevad kolm on eksamieelduseks)

Aine hinde moodustumine (kokku 100p):

- **Kodutööd – 28p**

Kodutööde esitamine toimub ained.ttu.ee keskkonnas iganädalaselt. Tähtajaks 24h enne järgmise nädala praktikumi algust. Näiteks esimese nädala kodutöö tuleb esitada 24h enne teise nädala praktikumi. Kodutööd esitada .zip formaadis ning faili nime sees peab olema kindlasti tudengi nimi ning mitmenda kodutööga on tegemist.

- **Teooriatestid – 22p**

Orienteeruvalt 5dal ning 11dal nädalal toimuvad loengus teooriatestid, mille eesmärgiks on koodilugemise oskusi ning teadmisi programmeerimise algpõhimõtetest.

- **Praktiline kontrolltöö – 20p**

Praktiline kontrolltöö koosneb programmeerimisülesandest, mida lahendatakse praktikumi ajal ning mis läheb hindamisele.

- **Eksam – 30p**

Eksam on programmeerimisülesanne, mida tudeng iseseisvalt lahendab (kodus) ning seejärel kaitseb õppejõule.

Eksamieeldus ning aine sooritatuks lugemine

Eksamieeldusteks on:

- Mõlemad teooriatestid sooritatud vähemalt 6le punktile (max: 11)
- Praktiline kontrolltöö on sooritatud vähemalt 10.5le punktile (max: 20)

Aine läbimiseks peab kokku tulema vähemalt 50.5 punkti koos eksamiga.

- Näiteks: kui kodutöödest on 5 punkti, eksamilt 18, kontrolltööst 13 ning teooriatestidest 12, siis kogusumma on 48 ning ei ole piisav aine läbimiseks.

Orienteeruv ajakava

Aeg	Tegevus	Max punktisumma	Teemad
1. Nädal			Visual studio, muutujad, tingimuslauseid
1. Nädal	1. kodutöö ülevaatus	1	Meetodid, tsüklid
1. Nädal	2. kodutöö ülevaatus	1	Klassid
1. Nädal	3. kodutöö ülevaatus	1	Failist lugemine, kirjutamine
1. Nädal	4. kodutöö ülevaatus + teooriatest	1+11	Kuidas lahendada probleeme; kordamine
1. Nädal	5. kodutöö ülevaatus	2	<i>Interface</i> , abstraktsed klassid
1. Nädal	6. kodutöö ülevaatus	2	Erandid
1. Nädal	Praktiline kontrolltöö + 7 kodutöö ülevaatus	22	Kontrolltöö
1. Nädal		0	Testimine
1. Nädal	9. kodutöö ülevaatus	2	Kordamine
1. Nädal	10. kodutöö ülevaatus	2	Vormirakendused
1. Nädal	11. kodutöö ülevaatus	2	Vormirakendused
1. Nädal	12. kodutöö ülevaatus + teooriatest	2+15	Vormirakendused: mäng
1. Nädal	13. kodutöö ülevaatus	2	Kordamine
1. Nädal	14. kodutöö ülevaatus	2	Eksamiks valmistumine
1. Nädal	15. kodutöö ülevaatus	2	
Eksam		30	

Tööd on palju (ja peamiselt programmeerimist)

Aine maht on 12 EAP-d

Seega kokku 260 tundi õppimist

Kokku on semestris 16 nädalat

Nädalas vajalik koormus $260 / 16 = 16.25$ tundi

Loeng 1,5 tundi

Harjutustund 3.0 tundi

Kodus õppimine 11.75 tundi

KOKKU 16.25 tundi

Plagiaat ei ole aktsepteeritud

- Teiste pealt maha kirjutamine samuti mitte
- Kõik see kajastub parimal juhul teie hinded ja halvimal juhul eksmatrikuleerimises

NB! Jälgige õppekava kodulehte !!! (<https://ained.ttu.ee/>)

Plagiaat

Plagiaat e. loomevargus on teise isiku loodud teose või selle osa avaldamine oma nime all, ka võõraste seisukohtade esitamine nende allikale viitamata.

(Eesti keele seletav sõnaraamat, *sub* plagiaat. <http://eki.ee>)

Plagiaadi alla liigituvad ka:

programmeerimise koodile;

enda tehtud tööd;

kontrolltööde vastused;

...

Uuri lähemalt:

- Infotehnoloogia teaduskond: Allikate kasutamise juhend
- Infotehnoloogia teaduskonna õppuri akadeemiliste tavade rikkumise ja vääritud käitumise menetlemise kord

Suhtlemine

Meilidele vastamiseks peab arvestama paar päeva aega.

Kõik omad küsimused tuleb postitada kursuse foorumisse

Küsida võib (ja peab) loengutes, harjutustundides ja konsultatsioonides

Olge aktiivsed! Ärge häbenege küsida! Lolle küsimusi ei ole!

Arvuti on vajalik

Kooli arvutites on Windows 10

Kõik IT teaduskondade üliõpilased, õppejõud ja teadlased üle maailma võivad Microsofti tarkvara õppimiseks ja teaduse tegemiseks enda tarbeks DreamSpark keskkonnast alla laadida

The screenshot shows the Microsoft DreamSpark website. At the top, there is a purple banner for Visual Studio Community 2015 with the text "No cost. No catch. Download the new Visual Studio Community 2015 at no cost to students." Below this is the Microsoft DreamSpark logo and the text "Tallinn University of Technology - Informatics - DreamSpark Premium". Navigation links include Home, Help, Administration, and English. A user greeting "Hello, gunnar.pihho@ttu.ee" and links for "Your Account/Orders" and "Sign Out" are visible. A search bar and a shopping cart icon (0 items) are present. A green "Announcements" section lists two items: "Visual Studio Community 2015 - Update 2 is here! 2016-04-15" and "DreamSpark Premium Marketing Resources and Collateral Now Available 2015-03-05". A "Popular" sidebar on the left lists categories like New, Operating Systems, Developer Tools, Servers, Applications, and All. The main content area displays a grid of software products for download, including Windows 10, Access 2016, Project 2016, Visio 2016, Visual Studio 2015, Windows Server 2012, Expression Studio 4, XNA Game Studio 4.0, and Windows 8.1. A banner for Microsoft Imagine is also visible on the right.

Visual Studio Community 2015
No cost. No catch.
Download the new Visual Studio Community 2015 at no cost to students.

Microsoft
DreamSpark™
Tallinn University of Technology - Informatics - DreamSpark Premium

Home | Help | Administration | English
Hello, gunnar.pihho@ttu.ee | Your Account/Orders | Sign Out

Product Search [] [] Shopping Cart (0)

DreamSpark Premium Subscriptions

Announcements

- Visual Studio Community 2015 - Update 2 is here! 2016-04-15
- DreamSpark Premium Marketing Resources and Collateral Now Available 2015-03-05

Popular

- New
- Operating Systems
- Developer Tools
- Servers
- Applications
- All

Windows 10
Access 2016
Project 2016
Visio 2016
Visual Studio 2015
Windows Server 2012
Expression Studio 4
XNA Game Studio 4.0
Windows 8.1

Microsoft Imagine
Start exploring the cloud
GET AZURE NOW →

Millist arvutit kasutada ja mida meil vaja on?

- *Kas ma võin kasutada Mac-i?*

Jah; eeldusel, et sa oskad oma mac-i ise kasutada (navigeerud failisüsteemis). Mac-i kasutajad saavad kõik asjad tehtud; va nädalatel 11-13, kus on vaja Windowsiga arvutit. OS spetsiifilistele küsimustele vastamisel peate olema valimis kasutama ka google abi.

- *Kas ma võin kasutada oma arvutit?*

Jah, kodutööde tegemiseks on sul kindlasti vaja oma arvutit. Aga palun enne praktikumi installeeri sinna ka Visual Studio 2017/2019 ning .Net/C# komponendid.

Praktikumis võid kasutada ka kooliarvutit, kus on kõik vajalik tarkvara olemas.

Vajalik tarkvara

- Visual Studio 2017 või 2019.
 - Community litsentsiga saab tasuta alla laadida; dreamspark kaudu saab professionali ka tõmmata
 - Olemas ka Mac-ile, linuxiga läheb keeruliseks
- ReSharper Visual Studiole
 - Tudengitele tasuta

Tehnilised soovitused:

- Kui kasutad isiklikku arvutit, siis palun palun palun osta endale hiir. See teeb su elu efektiivsemaks. Õpi ka erinevaid klahvikombinatsioone oma elu lihtsustamiseks (ctrl+s; alt+tab).
- Kooliarvutil on hiir ja eraldi klaviatuur olemas, mis teeb töö mugavamaks ja kiiremaks.
- Vaata mõni lahendus failide jagamiseks, et saaksid kooliarvutis tehtud töid avada ka koduarvutis (kuna erinevate teenuste ligipääs on piiratud ja ajas muutuv, siis on USB pulk vast kõige kindlam).
- Puhasta oma isiklik arvuti ära. Visual Studio on suhteliselt nõudlik program ning ei ole midagi ärritavamast sellest, kui ta pidevalt kokku jookseb.
- Saa tuttavaks oma klaviatuuriga (vajadusel vaheta *layout*). Meil läheb vaja sümboleid: <>{|

Otsi nende asukoht üles ning oska neid unepaalt sisestada.

Kuidas hakkab asi välja nägema?

- Iga nädal on loeng, milles katame teoreetilised teemad ära.
- Iga nädal on praktikum, kus hakkame õpitut praktiliselt rakendama.
- Iga nädal on iseseisev kodutöö, mille esitamise tähtaeg on ENNE sinu rühma järgmist praktikumi.

- Kõik aine materjalid on ained.ttu.ee lehel; kogu oluline info tuleb sinna; sh kodutööd, esitamised jne.

- Keel: materjalid eesti/inglise keeles. Inglise keelt võib pidada „IT keeleks“.

Soovitused

- Ära peta kodutöödega. See on sinu võimalus õppida ja tagasisidet saada.
- Ära jää maha ja panusta IGA nädal aega! Teemad on nagu tetris; kõik tükid omavahel seotud.
- Otsi endale õpisõber/sõbrad. Koos on probleeme palju kergem lahendada. Asjad, millega üksi võid tunde pusida, saad mitmepeale minutitega lahendatud.
- Ülikool on „uus algus“: keskkoolis kehtinud rollid, suhtumised jms ei kandu kaasa. Kõik on puhas leht.
- Minu lubadus:
 - Kõik kes käivad praktikumides kohal ning panustavad mõistliku koguse aega IGA nädal, läbivad kursuse edukalt.

Nippe ellujäämiseks

- Statistiliselt lõpetavad suure tõenäosusega need inimesed, kes leiavad endale „õpisõbrad“.
- Selleks:
 - Mõtle enda kohta **üks/kaks sõna, mis sind kirjeldavad** („adrenaliin, BMW, nohik, kass jms“) ning ütle need **koos oma nimega** enda kõrval ning ees ja taga istuvatele inimestele.



Soovitused (kursuseks ja üldse)

- Ole aktiivne
- Kasuta aega õppimiseks ja enese arendamiseks
 - Muuda enese arendamine enda elu peamiseks vajaduseks
 - **Õppimine on pidev ja järjekindel protsess** (mitte viimase minuti ponnistus)
- Võta kõik mis TTÜ pakub
 - Külalislektorid (siin käib palju maailma nimesid)
 - Konverentsid
 - Raamatukogu
 - **Mektory**
- Suhtle
 - Õppejõududega
 - Kaasüliõpilastega
 - ... kõikidega, kes ette satuvad
- Õpi rohkem kui nõutakse (niipalju kui suudad)
- Ole kursis ülikooli, teaduskonna ja instituudi tegemistega
- Osale tudengiprojektides
- Viivita tööle minemisega

Soovitused (kursuseks ja üldse):

- Osalege võimalikult palju loengutes ja praktikumides
- Ülikool on aeg „mõtlemiseks“ ning avastamiseks, tulevikus nii palju aega selleks ei anta
- Planeerige igal nädalal aeg kodutööks
- Kui soovite nõu küsida või hakkate maha jääma, tulge kohe konsultatsiooni!
- Kui olete „rongist maha jäänud“, siis panustage intensiivselt ning läbige kõik vahepealsed teemad; õppimine on nagu trepp, astmeid vahele jättes läheb keeruliseks
- Küsige kolleegidelt nõu, kuid ärge tehke copy-pastet

Aine eesmärgid

- Lihtsustatult: oskad programmeerida
- Keeruliselt:
 - Saad aimu, mis sind ümbritsevate IT süsteemide „köögipolel“ toimub
 - Oskad probleeme väiksemateks alamülesanneteks jaotada
 - Enne mõne ülesande lahendamist võtad hetke, et aru saada, kas mõistad probleemi
 - Oskad vähemal või rohkemal määral koodi kirjutada
 - Oskad ka enda arvutit paremini kasutada

Programmeerimine: müüdid ja tegelikkus

- Sa pead oskama väga hästi matemaatikat

In most cases you can see that the hard maths (the physical and geometry) is either done by a computer or has been done by someone else. While the calculations do happen and are essential to the successful running of the program, the programmer does not need to know how they are done.



Programmeerimine: müüdid ja tegelikkus

- Sa pead olema mees
- USA-s 78.6% programmeerijatest on mehed
- Keskmiselt ongi sooline jaotus 80/20 meeste kasuks
- Naisi tehnilistel kohtadel iga aastaga aina rohkem
- Maailma esimene programmeerija, Ada Lovelace, oli naine.



Programeerimine: müüdid ja tegelikkus

- Programmeerija oskab lahendada kõik arvutimured
- IT on lai valdkond, enamasti programmeerijad ei oska lahendada riistvaralisi jms probleeme



Aga kui ma ei taha saada programmeerijaks?

- 19 sajandil oli maailmas 2000 ametit, praegu on umbes pool miljonit. Nendest aina suurem osa on lähemal või kaugemal määral seotud IT-ga.
- Sa ei õpi ühte konkreetset ametit, vaid erinevaid oskusi mida elus rakendada
- Baasraamistik muutub maailmas; sa ei pea saama programmeerijaks kuid taustal toimuva mõistmine aitab pea igas ametis.
- Arvutioskus on vajalik pea iga ameti juures.

Miks programmeerida?

- Struktureeritum mõtlemine
- *When you begin to learn how to program, you also learn a new way of tackling problems, a more structured way and your brain begins to be programmed to deal with each problem as several smaller ones. Your thinking becomes more creative and logical and this doesn't use apply to your coding – it applies to your whole life.*



"IGOR, YOU FOOL! I TOLD YOU TO STEAL A HEALTHY BRAIN...NOT THE BRAIN OF A PROGRAMMER!"

Palk

Juuniorarendajal on palk väiksem

Kogenud arendajal on võimalik teenida omajagu

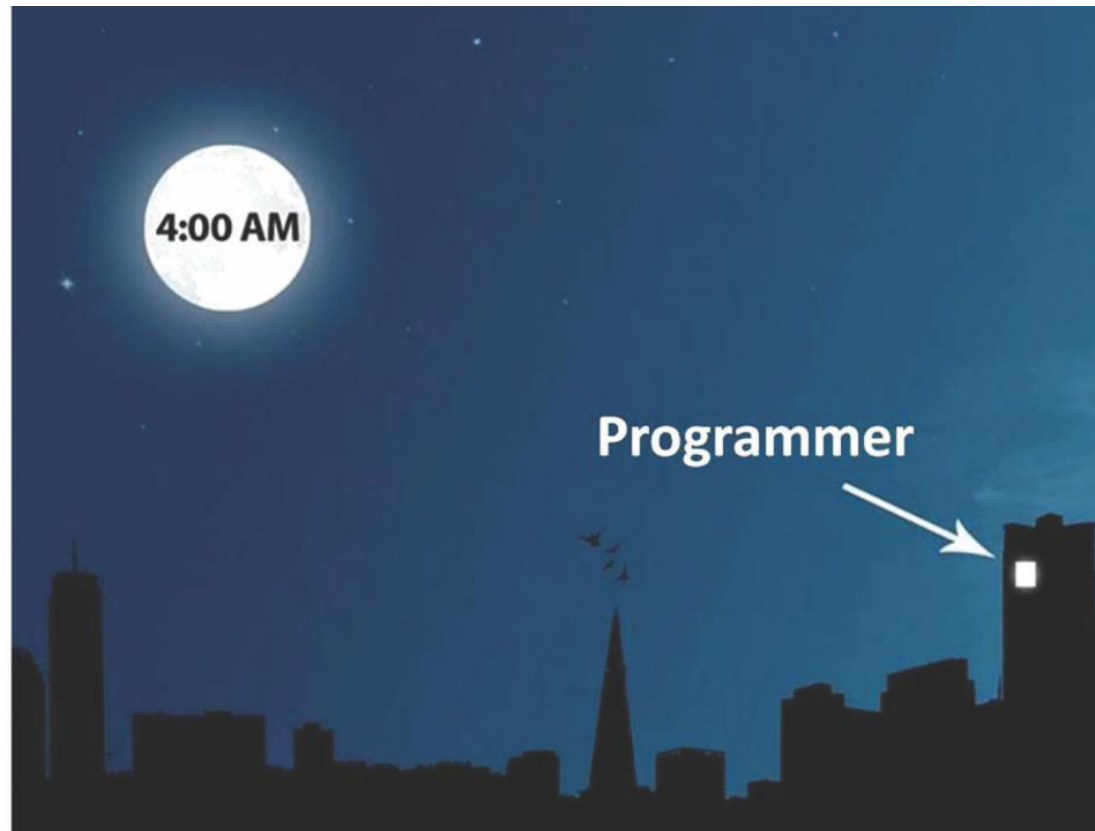
Me looking at US developer salaries



Vabadus

Suurem vabadus tööaja/koha suhtes kui mitmel teisel ametil

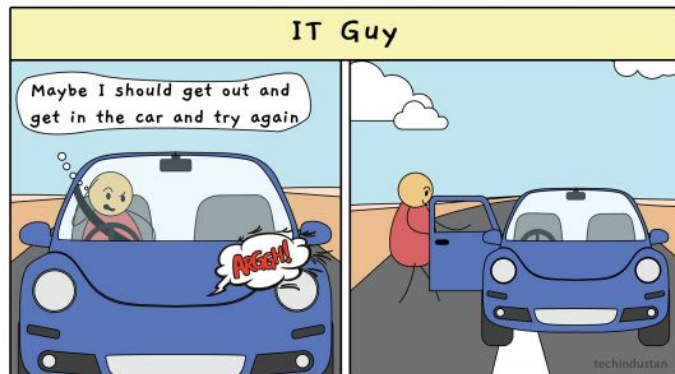
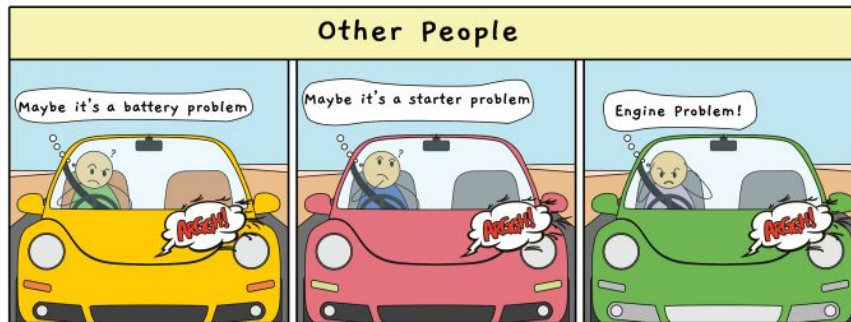
Ei tunne kultuuripiire



Kannatlikkus

- Sageli esimene lahendus probleemile ei tööta
- Mõnikord ka 50nes mitte
- (Aga lõpuks lahenduse leidmine on võidukas tunne)

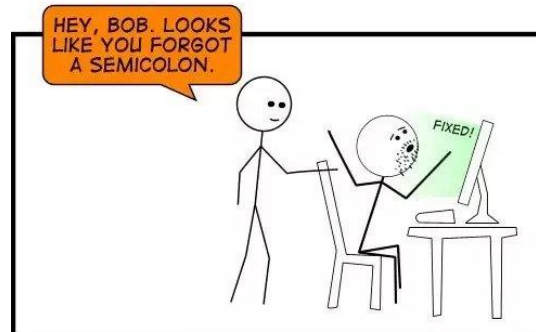
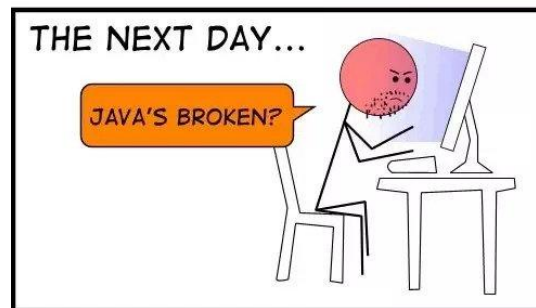
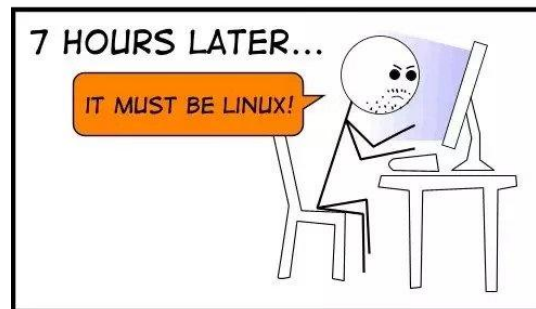
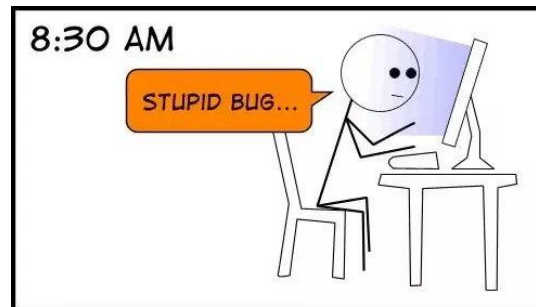
When the car won't start



I HATE PROGRAMMING
I HATE PROGRAMMING
I HATE PROGRAMMING
I HATE PROGRAMMING
IT WORKS
I LOVE PROGRAMMING

Tähelepanelikkus

- Üks puuduv koma või semikoolon 1000 realises koodis võib takistada selle käimaminemist



Loovus

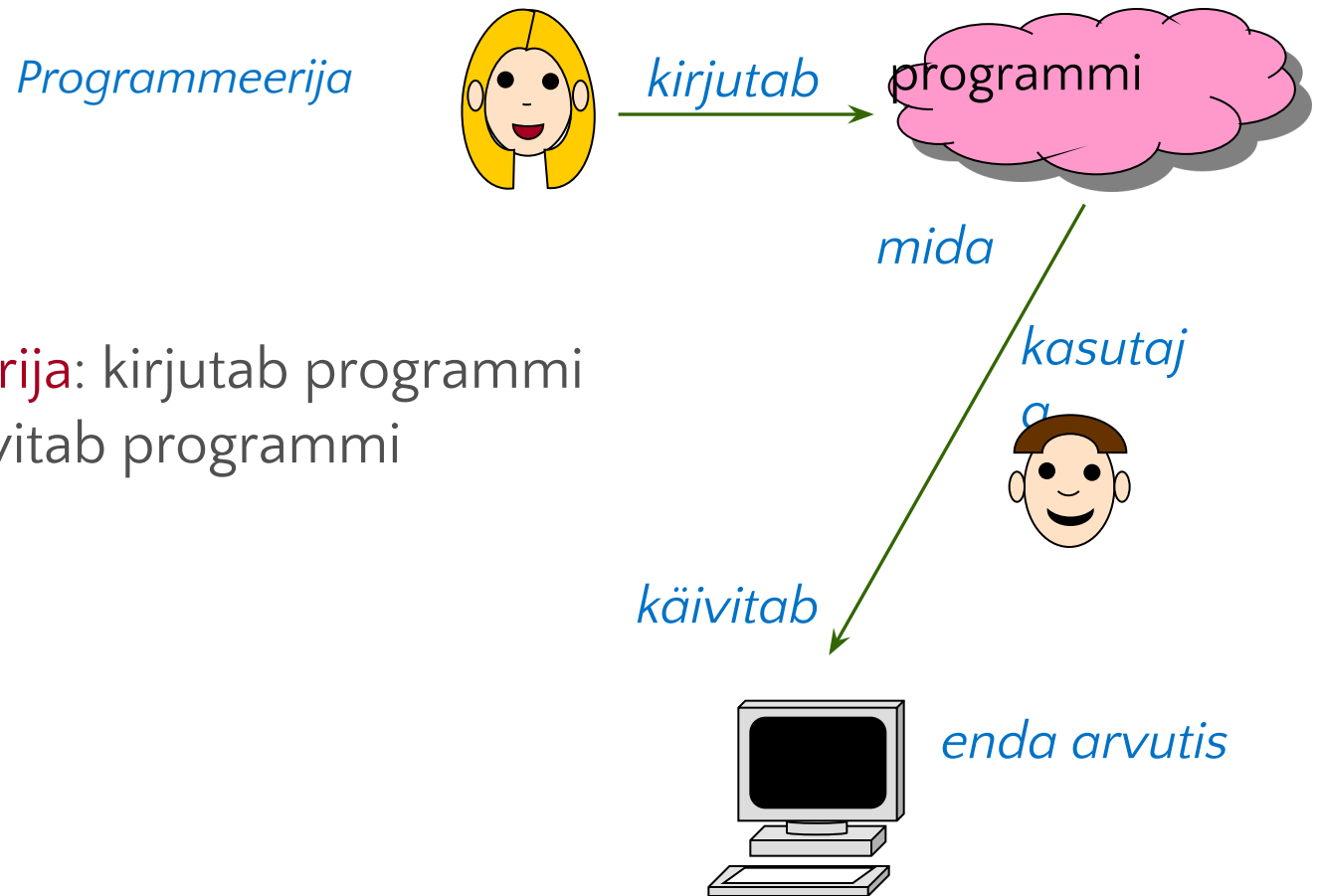
- Enamusele programmeerimisülesannetele on rohkem kui üks lahendus; igal omad head ja vead
- Mõnikord peab katsetama/otsima väga erinevaid lahendusi ning „kastist välja mõtlema“
- Probleemile tuleb leida uusi vaatenurki



Arvutid

- Universaalsed masinad, mis täidavad programmides antud korraldusi
- Piirideks on kujutlusvõime
- Õnneks teevad arvutid
 - teevad täpselt (ja ainult) seda, mida kästakse

Programmide loomine ja jooksumine *)



- **Programmeerija**: kirjutab programmi
- **Kasutaja**: käivitab programmi

Arvutid

- Universaalsed masinad, mis täidavad programmides antud korraldusi
- Õnneks arvutid
 - teevad täpselt (ja ainult) seda, mida kästakse
 - teevad seda väga kiiresti
- **Õnnetuseks arvutid**
 - **teevad täpselt (ja ainult) seda, mida kästakse**
 - **teevad seda väga kiiresti**

To Err is Human; To Really Foul Things Up Requires a Computer *)

*) <http://quoteinvestigator.com/2010/12/07/foul-computer/>

Levinud müüdid ja vabandused

„Arvutid on intelligentsed”

Tegelikkuses ei ole ei lollid, ega ka intelligentsed

On masinad, mis teevad ainult seda, mida programmeerija on programmeerinud

Arvuti oskused on väga piiratud (liida kaks arvu, salvesta arv, ...)

„Arvuti läks lolliks”

„Arvuti ei luba seda teha”

„Arvuti kaotas faili ära”

„Arvuti rikkus faili ära”

Arvutid ei tee vigu *

- Programmid ei tee kah vigu
- Vigu teevad programmeerijad

* Inimeste tehtud vead riistvaras on tunduvalt haruldasemad, kui inimeste tehtud vead tarkvaras

The “Blue Screen Of Death”

WARNING!

The system is either busy or has become unstable. You can wait and see if it becomes available again, or you can restart your computer.

- * Press any key to return to Windows and wait.
- * Press CTRL+ALT+DEL again to restart your computer. You will lose unsaved information in any programs that are running.

Press any key to continue _

Programmeerimine on karm

- Programmid kukuvad kokku (pole veel väga halb)
- See, et programm kokku ei kukku ei tähenda, et ta korrektset töötab
- Vead programmis on tapnud inimesi (meditsiiniseadmed)
- Vead programmis on „põletanud“ raha
(Ariane 5, 1996: \$10 miljardit, pisike programmi viga)
- Programmeerijad vastutavad koodi eest
- Meie eesmärk: **usaldusväärsed, koostalitlusvõimelised ja evolutsioneeruvad ettevõtte infosüsteemid**

*) <http://www.jaist.ac.jp/~bjorner/>

Most costly failures

- In 1962 a \$80 million racket exploded after being air for 5 minutes.
- The reason? a tiny typo in mathematical code

- In 1996: A space error: \$370 million for an integer overflow
- Start. 37 seconds of flight. KaBOOM! 10 years and 7 billion dollars are turning into dust.
- The reason? Wrong variable type used: integer overflow.

- In 1999, the \$125 million (\$172 million in 2014 dollars) Mars Climate Orbiter flew off course and disintegrated after spacecraft engineers forgot to convert from English to metric measurements.

- Heathrow terminal in UK built an automatic baggage handling system. It was tested with over 12000 pieces of luggage but still failed on the opening day. Why? Probably “real life” scenarios such as removing a bag from the system manually had caused the entire system to become confused and shut down. Over the following 10 days some 42,000 bags failed to travel with their owners, and over 500 flights were cancelled.

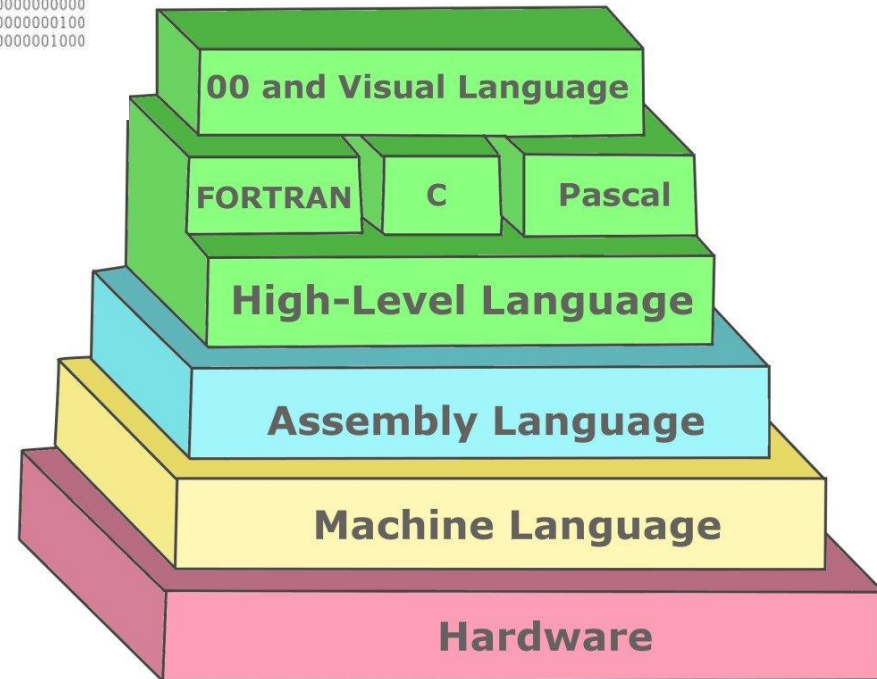
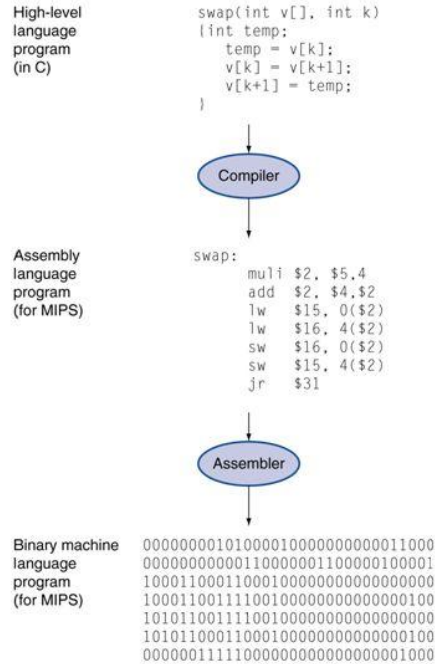
Kus programmid asuvad?

- **Salvestatud programmiga arvuti:** programmid asuvad arvuti mälus
- Programm saab arvuti mälus olla erinevates kujudes:
 - **lähtekood:** programmeerimiskeeles kirjutatud ja inimesele mõistetav tekst
 - **masinkood (täidetav kood):** arvutile mõistetav programm
- **Kompilaator** on masin, mis muudab programmi teksti (lähtekood) arvutile mõistetavateks korraldusteks (täidetav kood, exe ja dll lõpuga failid)



Levels of Program Code

- High-level language
 - Level of abstraction closer to problem domain
 - Provides for productivity and portability
- Assembly language
 - Textual representation of instructions
- Hardware representation
 - Binary digits (bits)
 - Encoded instructions and data



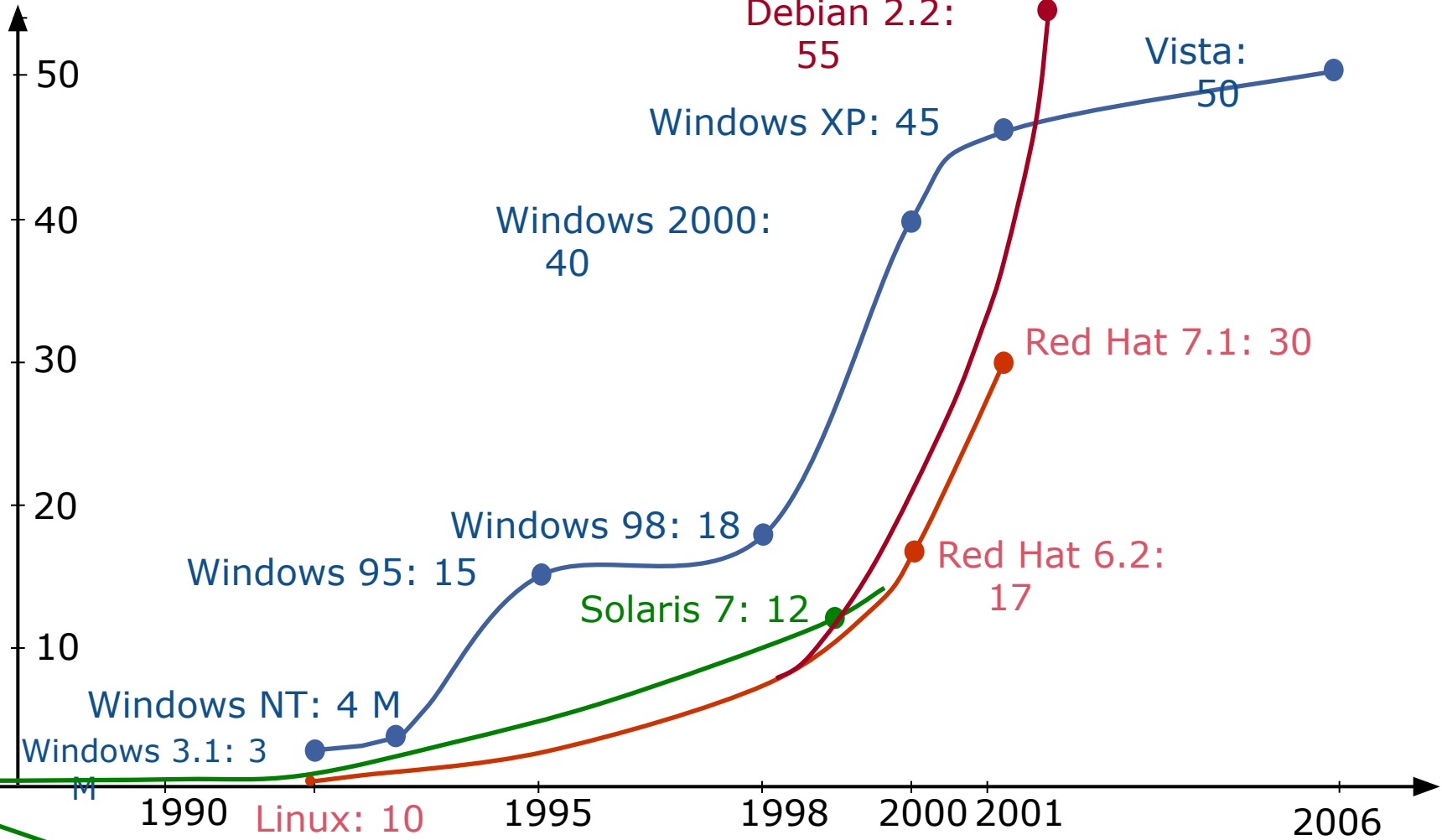
Tarkvaratehnika (*software engineering*)

Tarkvara inseneeria

- Oskus kirjutada tarkvara mis on
 - Korrektnel – teeb seda (ja ainult seda), mida peab tegema
 - Muudetav (ka evolutsioneeruv) – ettevõtete ärinõuded muutuvad
 - Loetav (iga loll suudab kirjutada tarkvara, millest saab arvuti aru)
 - Korduskasutatav – jalgratast ei ole vaja leiutada
 - Vastupidav – saab hakkama ka inimese tehtud vigadega
 - Turvaline – suudab taluda ründeid
- Testitav – kood, mis ei ole testitud ei ole puhas kood
- Meie eesmärk: Usaldusväärsed, koostalitlusvõimelised ja evolutsioneeruvad ettevõtte infosüsteemid

Koodiridade arv operatsioonisüsteemides

Lines of code (millions)



Programmeerimine on keeruline

- Korrektse programmi kirjutamine on aeganõudev *)

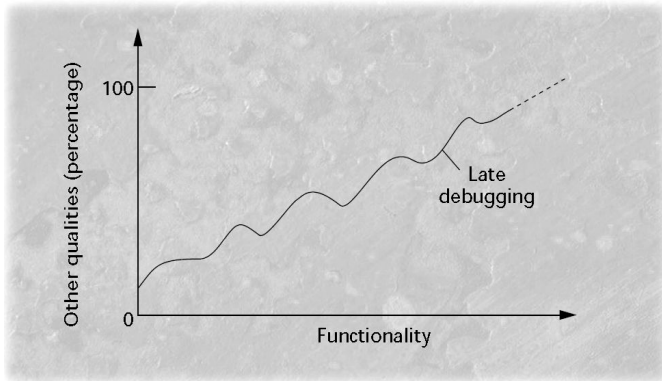


Figure 1. The most common development scheme addresses functionality and quality simultaneously.

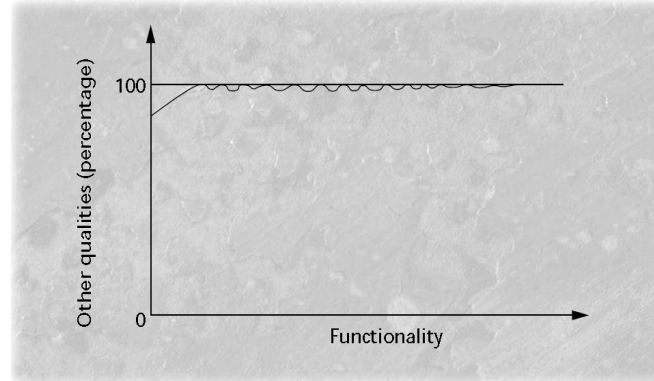


Figure 2. The Quality First development scheme does not add functionality until the quality is perfect.

- Katsetamine ja vigadest õppimine on väga ebaefektiivne
- Vead on väga kallid
(raha ja inimelud; 2005, USA, haigla infosüsteemid, 100000 inimelu)
- Õpi kvaliteetse ja puhta koodi kirjutamine peensusteni selgeks

Mis on programm

- A **computer program** is a collection of instructions that performs a specific task when executed by a computer.
- **Computation** is any type of calculation that includes both arithmetical and non-arithmetical steps and follows a well-defined model understood and described as, for example, an algorithm.
 - Näiteks: tōsta sõnas keskmine täht viimaseks.
- „Retsept“

Programm: näide: auto tankimine manuaaljaamas

- Millistest sammudest võiks koosneda? Võimalikult detailselt
- Vaata, kus pool asub tankimisava
- Vali õige kütusega tankur
- Sõida tankurile õigele poole
- Veendu, et kütuse eest tasumiseks on raha (või jalas head jooksutossud)
- Ava kütusepaak
- Aseta püstol paaki
- Vali õige kütus ning alusta tankimist
- Tangi sobiv kogus
- Aseta püstol tagasi
- Sulge kütusepaak
- Tasu kütuse eest

Programmeerimiskeeled

Õppides ühte programmeerimiskeelt, siis õpite põhimõtteliselt kõiki

Eri keeled on eri probleemide lahendamiseks paremad kuid baasosad on neil sarnased

Erinevad: süntaks, teegid jne

Meie kasutame C#

Miks C# ja .NET?

Suurem osa maailma äritarkvarast on kas Java või .NET rakendused

C# ja Java on nii sarnased, et kui oskad ühte, oskad ja teist


C# keeles ja .NET vahenditega saad kirjutada nii Windows, iOS, Linux kui ka Android masinatele ... nii töölaua, veebi kui ka mobiilirakendusi

C# keel on väga hea objektorienteeritud keel

Visual Studio on väga võimas integreeritud arendusvahend

Esmased ehitusklotsid:

- Main meetod – automaatselt genereeritud meetod, millest toimub koodi käivitamine. Esialgu kirjutame kogu oma koodi selle



```
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace MyFirstApp
8 {
9     class Program
10    {
11        static void Main(string[] args)
12        {
13            //We write our commands here
14        }
15    }
16 }
17
```

- Käsud – laused, millega ütleme programmile, mida ta tegema peab. Nt: liida kaks arvu, prindi konsooli väärtus jne
- Muutujad (*Variables*). Muutujad hoiavad endas väärtust ning võivad olla eri tüüpi.

Praktiline osa: sinu esimene programm; klassikaline „Hello world“.

- Programm täidab kõik main meetodis olevad käsud ning lõpetab seejärel töö.

```
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace MyFirstApp
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
            Console.ReadLine();
        }
    }
}
```

Mida meil esialgu vaja teada:

- Programm käivitub ALATI Main meetodist. Programmil saab olla (ja peab olema) ainult üks Main meetod. (Mõttele nt jooksuraja stardikohale).
- Meie esimesed käsud:
- `Console.WriteLine(„HelloWorld“)` – prindib sulgudes ja jutumärkides oleva teksti konsooli
- `Console.ReadLine()` – loeb seda, mida kasutaja konsooli kirjutas; teksti sisestamise lõpetamiseks tuleb vajutada Enter

Mida võiks teha antud käsud?

```
static void Main(string[] args)
{
    Console.WriteLine("Hello student!");
    Console.Write("Hello!!");
    Console.WriteLine("Please enter your name:");
    Console.ReadLine();
    Console.WriteLine("Thanks");
}
```

```
Hello student!
Hello!!Please enter your name:
```

```
Hello student!
Hello!!Please enter your name:
Liisa
Thanks
```

Mida võiks teha antud käsud?

```
static void Main(string[] args)
{
    Console.WriteLine(„Täna on ilus päev!");
    Console.WriteLine(„Homme on veel ilusam.");
}
```


Muutujad

- **Muutuja** on koht (aadress) mälus, mis hoiab väärtusi.
- Sageli kasutame samu andmeid (näiteks nimi, valuutakurss vms) koodis korduvalt, sellisel juhul pole otstarbekas infot dubleerida: mitmes kohas muutuste tegemine on ajamahukas ja keeruline, suureneb vigade oht. Seetõttu kasutatakse **muutujaid**.

- *Näide:*

Inimene Tõnu ja koer Kati elavad koos. Tõnu vaatab telekat ja Kati haugub.

Esimeses lauses ütleme, et kes on Tõnu ja Kati. Edaspidi piisab ainult nende nimest, pole vaja enam tüüpi öelda.

Inimene Tõnu ja koer Kati elavad koos. Inimene Tõnu vaatab telekat ja koer Kati haugub.

Muutujad

- Muutuja tüüp määrab selle, milliseid väärtusi saab selles muutujas hoida ning kui palju mälu talle eraldatakse. Lisaks panbe paika võimalikud operatsioonid.
- Muutuja väärtust saab igal hetkel muuta.
- Iga muutuja nime saab kasutada ainult ühe korra.

```
string name = "Dr. Jones";  
Console.WriteLine("Good morning " + name);
```

Muutujad

Muutuja tüüpe on palju; lihtsaimad:

- **string**: sõne; sisaldab tekstijada. Nt „Tere maailm!“
- **int**: täisarvud, nt 4, 100
- **bool**: tõeväärtus; variandid „yes“, „no“

Miks on vaja eri tüüpi muutujaid?

- $1+1= ?$
- „1“ + „1“ = ?

- $1+1=2$
- „1“ + “1“ = „11“

Erinevat tüüpi andmetega saab teha erinevaid operatsioone.
Arvutile peame ütlema, mida me täpselt teha tahame.
Muutuja tüübiga saame selle paika panna.

Numbrid: kirjutatakse ilma jutumärkideta

```
int a = 5; //muutuja a väärtus on 5
int b = 6; //muutuja b väärtus on 6
int c = a + b; //muutuja c väärtus on 5+6=11
b = 10; //muutuja b uus väärtus on 10
c = a + b; //muutuja c uus väärtus on 15
```

Tekst: kirjutatakse jutumärkidega

```
string a = "5"; //muutuja a väärtus on tekstiline 5
string b = "6"; //muutuja b väärtus on tekstiline 6
string c = a + b; //muutuja c väärtus on 5+6=56
b = "10"; //muutuja b uus väärtus on tekstiline 10
c = a + b; //muutuja c uus väärtus on tekstiline 510
```

Muutujad: vead

- Iga muutuja nime saab kasutada AINULT ühe korra
- Muutuja tüüpi ei saa muuta
- Muutuja tüüpi tohib defineerida AINULT ühe korra
- Iga tüüp eeldab enda tüüpi väärtust; nt string saab olla ainult tekst

```
string a = "5"; //muutuja a väärtus on  
tekstiline 5
```

```
string b = "6"; //muutuja b väärtus on  
tekstiline 6
```

```
string b = "10";
```

```
//muutuja b on juba defineeritud!
```

```
int c = a + b;
```

```
//muutuja c on int tüüpi, aga a ja b string
```

Muutujad: kokkuvõte

Defineerimine:

```
<data_type> <variable_name>
```

```
int i, j, k;  
char c, ch;  
float f, salary;  
double d;
```

```
int: int i; string firstName;
```

Initsialiseerimine:

```
variable_name = value; <data_type> <variable_name> = value;
```

```
int x = 1; // x holds the value 1  
x = 2;    // now x holds the value 2  
string greeting = "Hello, World!";
```

Initsialiseerida saab ka defineerimise ajal:

```
int a;  
a = 5;    =    int a = 5;
```

Tingimuslause; if/else

- „If“ lause määrab tingimuslause alusel, milline lause järgmiseks käivitatakse.

```
bool condition = true;

if (condition)
{
    Console.WriteLine("The variable is set to true.");
}
else
{
    Console.WriteLine("The variable is set to false.");
}
```


Programmi struktuur:

```
1  using System;
2
3  namespace HelloWorld
4  {
5      class Program
6      {
7          static void Main(string[] args)
8          {
9              //your code here
10             Console.WriteLine("Hello World");
11         }
12     }
13 }
14
```

The diagram shows three nested rectangular boxes on the left side of the code, representing the structure of the program:

- A large yellow box labeled "Nimeruum" (Namespace) covers lines 3 through 13.
- A medium green box labeled "Klass" (Class) covers lines 5 through 12.
- A small light blue box labeled "Meetod" (Method) covers lines 7 through 11.

Each box has a small square icon with a minus sign on its left side, indicating it is a collapsed view of the code block it represents.

Ülesanded:

- Logi sisse ained.ttu.ee keskkonda ning lisa end ainesse „ITB1701 – Infosüsteemide arendamine I: baasoskused “
- Installeeri Visual Studio oma arvutisse ENNE järgmist praktikumi. (Lisainfot selle kohta mida meil vaja on, leiad s Summary failist)
- Õpi leidma klaviatuurilt `{<>}` märke
- Tee kursuse lehelt esimese nädala alt ülesanne **Loenguharjutus** ning too tulemus oma järgmisesse praktikumi.

Tänu

