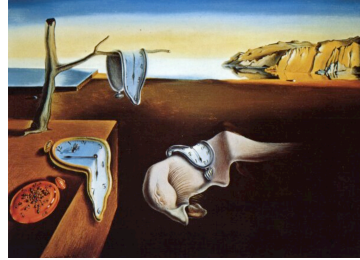




Välkomna till
Programmeringsmetodik DV1
Programkonstruktion 1
 http://www.it.uu.se/
 edu/course/homepage/pkpm/HT05

Föreläsare
Lars-Henrik Eriksson
 lhe@it.uu.se, http://user.it.uu.se/~lhe

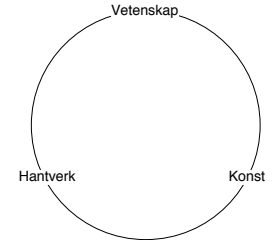


Salvador Dalí: Minnets varaktighet (1931)

Vad är ett "program"?

- Datorn
 - kan själv inte utföra något
 - har potential för att utföra allt.
- Program är *instruktioner* till datorn hur den skall utföra en uppgift.
- Program utför *beräkningar* där det undersöker och ändrar *data*.
- Data *representerar* någonting utanför datorn själv.
- Program = *Algoritmer + Datastrukturer*

(Min) syn på programmering



Biff Stroganoff

Matrecept kan ses som en sorts program. De innehåller instruktioner om hur man skall utföra en uppgift.

- Skär 500 g benfritt nötkött i cm-tjocka strimlor, ca 3 cm långa.
- Skala och hacka 2 gula lökar.
- Bryn kött och lök.
- Tillsätt 1 tsk salt, 1 krm peppar, 2 msk tomatpuré och 1,5 dl vatten.
- Efterstek i 5 min.
- Späd med 1,5 dl crème fraiche mot slutet.

Hur detaljerade skall instruktioner vara?

Det beror på läsaren!

Hur man skär kött i strimlor

1. Tag kött, en vass kniv och en skärbräda.
2. Tag en köttbit.
3. Skär loss en smal bit kött med kniven.
4. Finns det kött kvar på köttbiten? Gå i så fall till steg 3.
5. Finns det köttbitar kvar? Gå i så fall till steg 2. Annars är det klart!

"Hur man skär..." utgör ett *underprogram* till "Biff Stroganoff".
 Observera att instruktioner kan
 — läggas i följd — ges som alternativ — utprepas —
de grundläggande *programstrukturena*

Algoritm

"Effektiv procedur" för beräkning

- 1) Mekanisk (skall inte kräva "intelligens" för att förstå)
- 2) Ändliga resurser (måste bli färdig)
- 3) Deterministisk (det skall alltid vara entydigt vad som skall göras)
- 4) "Aritmetiserbart" problem (problemet skall kunna representeras i termer av data – egentligen heltal)

Vad som är "mekaniskt" beror förstås på vem som tolkar algoritmen.

Ett program är (oftast) en algoritm.
 (men med "algoritmer" i dagligt tal menar man normalt fundamentala konstruktionsmönster för programmering).

Euklides från Alexandria (325 – 265 f.v.t.)



Största gemensamma delare

(gcd – Greatest Common Divisor)

Det största heltal som jämnt delar två andra heltal.

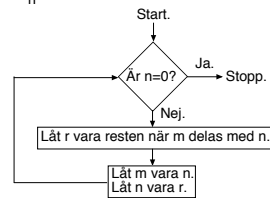
Exempel: Vad är $\frac{126}{168}$?

gcd till 126 och 168 är 42. $126 = 3 \cdot 42$, $168 = 4 \cdot 42$ så

$$\frac{126}{168} = \frac{3 \cdot 42}{4 \cdot 42} = \frac{3}{4}$$

Euklides algoritmen

Förvillkor: m och n är positiva heltal
 Eftervillkor: m är största gemensamma delare till de ursprungliga m och n.
 Variant: n



Euklides algoritmen: exempel

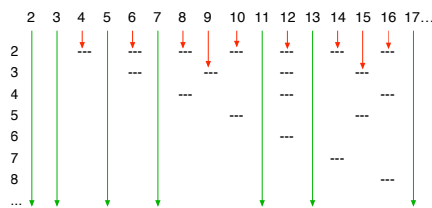
Största gemensamma delaren till 126 och 168

| m | n | r |
|-----|-----|-----|
| 126 | 168 | 126 |
| 168 | 126 | 42 |
| 126 | 42 | 0 |
| 42 | 0 | 0 |

Eratosthenes från Kyrene (276 – 194 f.v.t.)



Eratosthenes såll



De tal som faller igenom sållet är primtal.

Pseudoalgoritmer

Eratosthenes såll är en *pseudoalgoritm*. (En mekanisk procedur som inte uppfyller kraven på att vara en algoritm.)

Alla program är inte algoritmer.
 Ofta vill man inte att program skall avslutas – t.ex. program som styr processer av olika slag:

- trafikljus
- telefonväxlar
- DVD-spelare
- kärnkraftverk...

Abu Ja'far Mohammad Ibn Musa al-Khwarizmi (c:a 780 – c:a 850)



Två böcker av al-Khwarizmi

Efter romarrikets fall skedde en vetenskaplig nedgång i Europa. Den islamska kulturen bevarade och utvecklade grekisk vetenskap. Matematikern al-Khwarizmi verkade i Bagdad som under 800-talet var ett kulturellt och vetenskapligt centrum.

- *Om beräkning genom "al-jabr" och "al-muqabala"*
Metoder för att förenkla ekvationer och lösa andragradsekvationer.
- *al-Khwarizmi om de indiska talen* ("Algoritmi de numero Indorum" – Endast en latisk översättning finns bevarad.)
Om siffrorna 0-9 och positionssystem med siffran 0.

Förvrängning av "al-jabr" och al-Khwarizmis namn har gett oss orden "algebra" och "algoritm".

Gottfried Wilhelm von Leibniz (1646 – 1716)



".....calculemus" (låt oss räkna)

Leibniz kalkylator



Leibniz kalkylator förverkligar (implementerar) algoritmer för de fyra räknesätten.

Gammal matematisk tabell

| sin 0° = 45° | |
|--------------|---|
| Grad | 1 2 3 4 5 6 7 8 9 |
| 0 | 0000 0001 0002 0003 0004 0005 0006 0007 0008 0009 |
| 1 | 0019 0038 0057 0076 0095 0114 0133 0152 0171 0190 |
| 2 | 0209 0228 0247 0266 0285 0304 0323 0342 0361 0380 |
| 3 | 0399 0418 0437 0456 0475 0494 0513 0532 0551 0570 |
| 4 | 0589 0608 0627 0646 0665 0684 0703 0722 0741 0760 |
| 5 | 0779 0798 0817 0836 0855 0874 0893 0912 0931 0950 |
| 6 | 0969 0988 1007 1026 1045 1064 1083 1102 1121 1140 |
| 7 | 1159 1178 1197 1216 1235 1254 1273 1292 1311 1330 |
| 8 | 1349 1368 1387 1406 1425 1444 1463 1482 1501 1520 |
| 9 | 1539 1558 1577 1596 1615 1634 1653 1672 1691 1710 |
| 10 | 1729 1748 1767 1786 1805 1824 1843 1862 1881 1900 |
| 11 | 1919 1938 1957 1976 1995 2014 2033 2052 2071 2090 |
| 12 | 2109 2128 2147 2166 2185 2204 2223 2242 2261 2280 |
| 13 | 2299 2318 2337 2356 2375 2394 2413 2432 2451 2470 |
| 14 | 2489 2508 2527 2546 2565 2584 2603 2622 2641 2660 |
| 15 | 2679 2698 2717 2736 2755 2774 2793 2812 2831 2850 |
| 16 | 2869 2888 2907 2926 2945 2964 2983 3002 3021 3040 |
| 17 | 3059 3078 3097 3116 3135 3154 3173 3192 3211 3230 |
| 18 | 3249 3268 3287 3306 3325 3344 3363 3382 3401 3420 |
| 19 | 3439 3458 3477 3496 3515 3534 3553 3572 3591 3610 |
| 20 | 3629 3648 3667 3686 3705 3724 3743 3762 3781 3800 |
| 21 | 3819 3838 3857 3876 3895 3914 3933 3952 3971 3990 |
| 22 | 4009 4028 4047 4066 4085 4104 4123 4142 4161 4180 |
| 23 | 4199 4218 4237 4256 4275 4294 4313 4332 4351 4370 |
| 24 | 4389 4408 4427 4446 4465 4484 4503 4522 4541 4560 |

Tabellen ger värden på sinus för vinklar i steg av 0,1°

Charles Babbage (1791 – 1871)

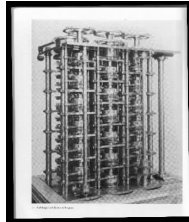


Differenskalkylen

Polynom kan approximera matematiska funktioner som sinus, logaritm etc. (med taylorutvecklingar – se kurs i analys)
Differenskalkylen beräknar polynom med enbart additioner.

| x | 2x ² -3x+4 | 1:a diff. | 2:a diff. | 3:e diff. |
|---|-----------------------|-------------|-----------|-----------|
| 0 | 4 | | | |
| 1 | 3 | -1 | | |
| 2 | 6 | 3 | 4 | |
| 3 | 13 | 7 | 4 | 0 |
| 4 | 24 | 11 | 4 | 0 |
| 5 | ?? (24+11+4+0) | ?? (11+4+0) | ?? (4+0) | ?? (0) |

Babbages differensmaskin (del)



Differensmaskinen implementerar en algoritm för differenskalkyl.

Augusta Ada Byron grevinnan av Lovelace (1815 – 1852)



Babbages analytiska maskin

Den analytiska maskinen skulle programmeras att utföra *godtycklig* algoritm.

Beståndsdelarna i princip samma som i en modern dator, men allt byggt mekaniskt.

Ada Lovelace skrev det första *programmet* i modern mening – för beräkning av Bernoullital.

Den analytiska maskinen kunde aldrig byggas – dätidens finmekanik räckte inte.

Alonzo Church (1903 – 1995)



Alan Mathison Turing (1912 – 1954)



Vad är "mekaniskt"?

Matematiska modeller av beräkningar.

Church: λ-kalkylen ("lambdakalkylen")

Turing: turingmaskiner

Church-Turings tes: Dessa modeller kan beskriva *allt* som går att beräkna.

I praktiken: uttryckbart som program i en dator.

Alla beräkningar kan inte mekaniseras enligt Church/Turing. Kan de då utföras alls?

λ-kalkylen har blivit ett viktigt verktyg inom datavetenskapen.

Turing award

Datavetenskapens "nobelpris" är uppkallat efter A.M. Turing och delas sedan 1966 ut av Association for Computing Machinery (ACM)

Några pristagare:

- 1983: Ken Thompson, Dennis M. Ritchie – operativsystemet UNIX.
- 1988: Ivan Sutherland – grundläggande principer för datorgrafik
- 1991: Robin Milner – programspråket ML (bl.a.)
- 2001: Ole-Johan Dahl och Kristen Nygaard – objektorienterad programmering (Simula)
- 2002: Ronald L. Rivest, Adi Shamir, Leonard M. Adleman – kryptografi med publika nycklar ("RSA-krypto")
- 2003: Alan Kay – objektorienterad programmering (Smalltalk) och ideerna bakom persondatorer.
- 2004: Vinton G. Cerf och Robert E. Kahn – Internet (TCP/IP)

Datarepresentation

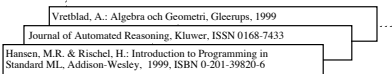
Datorer är byggda för att arbeta med *heltal*. Med hjälp av dessa kan annan information *representeras*. (Minns ni kravet på algoritmer om *aritmetiserbarhet*?)

T.ex. kan text representeras genom att *koda* bokstäver: genom kodningen A=1, B=2 etc. kan ordet "UPPSALA" *representeras* som talföljden 20 15 15 18 1 11 1.

En bild kan byggas upp av *bildpunkter* (pixels) av bestämd färg. Färger kan kodas som tre tal som anger inblandning av *rött*, *grönt* och *blått* i procent. T.ex. kan *mörkgult* kodas som 100 75 0. En bild representeras då som en (stor) mängd sådana taltriplar.

Datastrukturer

En bibliotekskatalog kan representeras som en *upprepning av poster*



Varje post *sätts samman* av olika *fält* – här texter (*strängar*) och tal. Poster kan ha *olika utformning* beroende på vilka data som behövs.

Sammansättning, upprepning och alternativ är de grundläggande metoderna för strukturering av data. (Jämför programstrukturer!)

Ada Lovelace om datarepresentation

"Again, [the *Analytical Engine*] might act upon other things besides number, were objects found whose mutual fundamental relations could be expressed by those of the abstract science of operations, and which should be also susceptible of adaptations to the action of the operating notation and mechanism of the engine . . . Supposing, for instance, that the fundamental relations of pitched sounds in the science of harmony and of musical composition were susceptible of such expression and adaptations, the engine might compose elaborate and scientific pieces of music of any degree of complexity or extent."

Notes on the description of the Analytical Engine, 1843

Grace Murray Hopper (1906 – 1992)



"If it's a good idea . . . go ahead and do it. It is much easier to apologize than it is to get permission"

Högnivåspråk

Datorn styrs med numeriska koder (*maskinspråk*) vars funktion definieras i termer av datorns inre uppbyggnad.

Data representeras ytterst som numeriska koder.

Människor vill ha mer lättanvänd notation. Därför finns *användarorienterade programspråk* (*högnivåspråk*) för att konstruera program.

En *kompilator* är ett program översätter från ett högnivåspråk till maskinspråk.

Grace Hopper skrev den första kompilatorn (≈ 1952).

Programmeringsmetodik/konstruktion är mer än att bara skriva program...

Ni skall även:

- lära er ett högnivåprogramspråk (Standard ML)
- förstå betydelsen av
 - specifikationer (vad?)
 - motivering (varför?)
 - dokumentation i övrigt
 - elegans, precision, klarhet...
- få en första introduktion till
 - algoritmer och datastrukturer
 - algoritmkomplexitet

PK16/PM1 HT-05 introduktionsföreläsning

Sida 30

Uppdaterat 2005-09-21

Små- och storskalig programmering

På denna kurs skall ni lära er *småskalig* programmering – att själv skriva små program, oftast för eget bruk. ("Små" betyder här några tusen rader programkod.)

En helt annan sak är *storskalig* programmering – att många programmerare tillsammans utvecklar stora program för någon uppdragsgivare. Stora program kan bestå av miljontals rader programkod. För detta krävs inte bara erfarenhet utan även andra kunskaper som behandlas på kurserna Programmeringsmetodik 2 och Programvaruteknik.

Jämför med att laga mat hemma till sig själv eller familjen mot att vara kock i ett restaurantkök eller vid en festmiddag för hundratals gäster!

PK16/PM1 HT-05 introduktionsföreläsning

Sida 34

Uppdaterat 2005-09-21

Thomas Thorild (1759 - 1808)



"Tänka fritt är stort men tänka rätt är större"
Devis över ingången till universitetsaulan

PK16/PM1 HT-05 introduktionsföreläsning

Sida 35

Uppdaterat 2005-09-21

En kurs på universitetet...

En universitetsutbildning skiljer sig från andra utbildningar genom att

- vila på *vetenskaplig grund*. Det som lärs ut skall inte vara fritt tyckande utan vila på "vetenskaplig ... grund samt på beprövad erfarenhet." (högskolelagen)
 - skall ge *förståelse* för inte bara *hur* utan även *varför*.
 - skall leda till *självständigt och kritiskt tänkande*. (högskolelagen)
- Vad har ni för anledning att tro eller inte tro på det som jag säger?

Man får spekulera i att homeopati fungerar eller att programutvecklingsmetod X ger billiga program av hög kvalitet, men i verkligheten är det underbyggda påståenden som gäller. Det kostar liv och egendom att "tänka fritt" vid brobygge, behandling av sjukdomar, konstruktion av programvara...

PK16/PM1 HT-05 introduktionsföreläsning

Sida 36

Uppdaterat 2005-09-21

Kursupplägg

Föreläsningar: A-föreläsningar (grundläggande).
B-föreläsningar (fullständiga – inkl. A-materialet)
C-föreläsningar (överkurs)

Lektioner: Redovisning/genomgång av laborationer, lärarledda övningar (första 29/9)

Laborationer: Självständiga övningar på kursmaterialet ensam eller i *grupper om två*, med schemalagda handledningstider (första 30/9)

Inlämningsuppgifter: Större uppgifter som löses självständigt och individuellt. Inlämning i december resp. januari.

Tentamen: Torsdagen den 22/12.

PK16/PM1 HT-05 introduktionsföreläsning

Sida 37

Uppdaterat 2005-09-21

Hur man blir godkänd på laborationerna

- 1) lämna in en (nästan) riktig lösning i tid – normalt kl. 08:00 andra arbetsdagen efter den schemalagda labben (första gången den 4/10) – *eller*
- 2) lämna in ett *seriöst försök* till lösning *i tid* – i detta fall får du betyg "K" (komplettering) på labben – *och*
 - Deltag på följande lektion (i detta fall alltså *obligatoriskt!*)
 - lämna in en (nästan) riktig lösning senast vid nästa inlämningstid.

I annat fall blir laborationen *underkänd*.

För att få godkänt på kursen måste *7 av de 10* laborationerna vara godkända.

PK16/PM1 HT-05 introduktionsföreläsning

Sida 38

Uppdaterat 2005-09-21

Presentation och opposition

För att träna på *munlig framställning* och *kritiskt tänkande* skall alla vid något tillfälle under kursen

- Presentera sin lösning på en labbuppgift på lektionen.
- Opponera på någon annans presentation.

Opposition – opponenter läser lösningen i förväg och funderar: Är detta rätt? Varför är det gjort som det är gjort? Kan det göras på bättre sätt? Efter presentationen ställer opponenter frågor om detta och den som presenterar (respondenten) skall kunna svara på frågorna.

Assistenterna väljer före varje lektionstillfälle ut vilka som skall presentera och opponera – ni får alltså reda på det med rätt kort varsel.

PK16/PM1 HT-05 introduktionsföreläsning

Sida 39

Uppdaterat 2005-09-21

Vilken var webbadressen nu igen?

<http://www.it.uu.se/edu/course/homepage/pkpm/HT05>

Läs informationen om kursen!
Kontrollera vilken lab/lektionsgrupp du tillhört!
Läs nyheter på webben! Minst två gånger i veckan.

PK16/PM1 HT-05 introduktionsföreläsning

Sida 40

Uppdaterat 2005-09-21