

LITT OM PROGRAMMERING

Noen råd om programmering knyttet til oblig. 1

For forelesning 13/9 2004

MAT-INF1100

Hvorfor Programmering i MAT-INF1100

- Programmering er viktig i realfag. Vi vil integrere det i kurs fra starten av.
 - Mange får bruk for programmering i masteroppgaver o.l.
 - Kunnskaper om et språk som Java nyttig bakgrunn for all databehandling
- Koding av algoritmer gir bedre følelse med hva de er og hva som kan gjøres med dem. Noen ganger økes også forståelsen av matematikken bak

Programmering ingen heksekunst, men krever trening

Programmer ukesoppgaver i INF1000 og MAT-INF1100

Siktemål med oblig. 1

- Få dere til å skrive noen programmer i Java *og få dem til å virke*
- La dere utforske fenomener som “overflow”, udefinerte størrelser og avrundingsfeil.
Det blir mye fokus på det nå fordi det er tidlig i kurset.
- Dere gjør dere kjent med de ulike typene (**float**, **double** etc.)
- Koding av de første løkkestrukturene – utregning av ting man ikke får til greit for “hånd” (primitiv kalkulator)

Samarbeid gjerne, men kod deres egne ting.

Javating

Numeriske ting ikke spesielt vektlagt i “Rett på Java”, men du finner det du trenger

- De innledende “besvergelses” i tallrike eksempler
- Deklarasjoner typer og konstanter i 2.4
- Matematiske funksjoner i tabell 2.4 – klassen Math
- Lokker og forgreninger i 4.2-4.3 (**if**, **while**, **for**)

Se også kompendium og eget skriv med tips på hjemmeside

Typer og konvertering

Hierarki

| | |
|---------------|------------------|
| int | 32 bits heltall |
| long | 64 bits heltall |
| float | 32 bits flyttall |
| double | 64 bits flyttall |

1. Automatisk konvertering går nedover i listen
2. Konvertering oppover må gjøres eksplisitt
3. Standard type for konstanter er **int** og **double**
2134234356L angir **long**, mens **4.56F** angir **float**
4. Funksjoner **Math.*** returnerer **double** selv om argumentet er **float**

Typetukling i Java 1

```
float a=1.5;          /* feil */
float a=1.5F;        /* ok */
a=2.37;              /* feil */
a=2.37F;             /* ok */
a=(float)2.37;       /* ok */
a=(float)2.37*3.1;   /* feil */
a=(float)(2.37*3.1); /* ok */
a=Math.cos(a);       /* feil */
a=(float)Math.cos(a); /* ok */
```

Typetukling i Java 2

```
float a=1.5F;           /* ok */
double b=4.76;         /* ok */
a=b;                   /* feil */
b=a;                   /* ok */
a=(float)b;           /* ok */
b=1/2;                 /* gir b=0.0 */
b=1.0/2.0;            /* gir b=0.5 */
long ii=101;          /* ok */
long ii=123123243434; /* feil */
long ii=123123243434L; /* ok */
a=ii;                 /* ok ???? */
```

Regneoperasjoner og typer

- Addisjon, subtraksjon og multiplikasjon virker ganske likt for alle typer
- Divisjon “/” er ulik for heltall og flytende tall
NB! Heltallsdivisjon ⇒ **resten sløyfes**
Eksempler: $13/8$ gir 1, $-8/3$ gir -2

Når en operasjon mellom to tall utføres gjøres begge om til den mest generelle typen. feks. vil

3.0/8

utføres som divisjon mellom to **double**.

PerogKari

Eksempel 4.1.18 i boka. Formel for Kari's vinningsjans

$$s = \frac{1 - \left(\frac{2}{3}\right)^n}{1 - \left(\frac{2}{3}\right)^{20}}$$

der Kari starter med n av 20 kr.

- Vi starter med et dårlig forsøk på $n = 5$.
- Vi ender med et program som regner for alle n .