

Endringer fra 3. til 4. utgave av "Programmering i Java"

Dette notatet beskriver endringer i begrepsbruk samt måten å programmere på. Endringer i bokas struktur og innhold er ikke tatt med her. Gi gjerne beskjed om du savner noe.

Kodestil

Tidligere utgaver har fulgt Java kodestil (<http://java.sun.com/docs/codeconv/>) med de unntak som er nevnt for eksempel i vedlegg F, side 851, i 3.utgaven. I 4.utgave følger vi Java kodestil i større grad. De mest merkbare endringene er følgende:

- Konstantnavn skrives med STORE bokstaver og understrekingstegn mellom ordene.
- Klammeparenteser og linjeskift brukes alltid i if/else/while/for/do-setninger. Eksempel:
 - Ikke
if (a > b) b = 20;
 - Men
if (a > b) {
 b = 20;
}
- Hele registeret av ulike kommentartyper er tatt i bruk. Se side 53 i 4.utgave. javadoc-kommentarer er imidlertid ikke tatt i bruk, men som tidligere, så vidt vist i et vedlegg.

I praksis har 2.kulepunkt foran medført at operatoren ?: til en viss grad er tatt i bruk. Hovedgrunnen er nok å spare plass, men det er en operator som er en del i bruk, og som en kanskje bør venne seg til å bruke mer enn det vi har gjort hittil.

Navngiving og terminologi

Forstavelserne get- og set- er tatt i bruk (i stedet for finn- og sett-) for metoder som henter ut og gir verdi til attributter.

I konstruktører brukes nå samme navn på parameterne som på objektvariablene. Forstavelsen start- på parameterne er dermed ikke i bruk lenger. (Tilordning skjer med this, eksempel this.navn = navn;)

Noen faglige termer er skiftet ut:

<i>3.utgave</i>	<i>4.utgave</i>
definisjonsområde	skop
tabell-liste	ArrayList
forkortet beregning	avbrutt beregning
katalog	mappe
konsoll, MS-DOS-vindu	kommandovindu/-linje

final som modifikator

final er tatt i bruk som modifikator for objektvariabler som ikke endrer verdi (readonly) i klassediagrammet). Jeg ser i ettertid at jeg dessverre ikke har klart å gjennomføre dette alle steder,

spesielt ikke i de siste kapitlene. Etter hvert kommer det info. om dette i trykkfeil-listen, og eksempelkode på hjemmesiden blir oppdatert.

Testing og feilhåndtering

Enhetstesting er vektlagt sterkere enn tidligere. Egne testklienter er lagt inn i slutten av flere av klassene. Dette er klienter som kjøres uten at brukeren trenger å skrive inn testdata. Sånn sett er det en manuell utgave av testverktøy som f. eks. JUnit, som gjennomgås så vidt i kapittel 22.

Feil i parameterne til konstruktører har i tidligere utgaver av boka i praksis vært ignorert. Det er nå valgt å innføre kasting av unntaksobjekter i situasjoner som denne – og det allerede i kapittel 6.5. Det er gjort for tidlig å bevisstgjøre på testing og feilkontroll. Imidlertid blir det "forstyrrelser" i koden, og det brukes derfor i hovedsak klasser som lages for å brukes i flere og større eksempler. Det må også nevnes at det kun er kasting (*throw new IllegalArgumentException()*) som innføres så tidlig. Fullstendig unntakshåndtering behandles ikke før i kapittel 15.

Modellering

For å vise sammenhenger mellom objekter av forskjellige klasser brukes nå kun aggregering og komposisjon. Begrepet assosiasjon er utelatt. Årsaken er at det programmeres på samme måte som aggregering.

Utvidet for

Utvidet for ("enhanced for") innføres nå i tilknytning til ArrayList. Det har vist seg at konstruksjonen sjelden brukes for tabeller.

Trondheim, august 2009

Else Lervik