

Forord



Dette er en lærebok for et begynnerkurs i programmering på høyskole- og universitetsnivå. Boken forutsetter ingen spesielle forkunnskaper i programmering ut over alminnelig bruk av en datamaskin, og den er plattform- og programmeringsverktøyuavhengig. Boken følges av et nettsted med forelesningsnotater, kildekode for alle programmeringseksempler, lenker til Java-ressurser m.m.

Hovedtema i boken er objektorientert programmering (OOP). Boken benytter UML (Unified Modelling Language) for å illustrere OOP-konsepser. Teknikker for problemløsning på datamaskinen vektlegges, og programmeringsspråket Java benyttes for utforming av løsninger.

Boken henvender seg i første rekke til nybegynnere i programmering, men den kan også brukes av programmerere som har bakgrunn fra andre språk enn Java.

Hva som er nytt i denne utgaven

Kvalitetsreformen i høyere utdanning har ført til omlegging av studietilbudet. Omleggingen har for eksempel ført til flere kurs med 10 studiepoeng i enkelte fag. Den nye utgaven av boken kan brukes til å kjøre ett eller flere kurs i programmering, med til sammen 20 studiepoeng.

Java versjon 5.0 introduserte en rekke nye språkkonstruksjoner og utvidede biblioteker. Mye av dette er tatt med i boken. Vi har innarbeidet ny funksjonalitet der det er hensiktsmessig å bruke den. Dette gjelder innlesing fra tastaturet med `Scanner`-klassen, den forenklede forløkken, formatering av verdier, håndtering av primitive verdier som objekter, oppramstyper, `StringBuilder`-klassen for håndtering av strenger, generiske typer og generiske metoder.

Boken inneholder også et avsnitt om pseudo-slumptallgeneratorer, som er nyttige for å lage interessante programmeringsoppgaver, for eksempel relatert til spill.

Test-drevet programutvikling og andre nyttige teknikker for å utvikle ikke-trivielle programmer blir tatt opp i et eget kapittel. En konkret problemstilling driver utviklingen fra idé til ferdig implementert løsning. I tillegg oppfordrer en del oppgaver til programtesting med JUnit, et ledende testrammeverk for Java-programmer.

Mange emner har blitt omstrukturert for bedre pedagogisk fremdrift, og denne omstruktureringen gir også større fleksibilitet når det gjelder kombinasjon av emner for pensumsammenstilling.

Hvert kapittel er grundig revidert. Nye figurer, eksempler og oppgaver er lagt til der det var nødvendig.

Temaer som er vektlagt

Vi har valgt å vektlegge følgende temaer i boken:

- **OOP med anvendelser:** Boken er strukturert rundt objektorientering og viser dens anvendelse i forskjellige sammenhenger. Den benytter klasser fra Javas standardbibliotek og legger vekt på utforming av egendefinerte klasser.
- **Konsepter fremfor syntaks:** Boken legger vekt på objektorienteringskonsepter, og viser deretter hvordan disse implementeres ved hjelp av språkkonstruksjoner i Java. Java-syntaksen er illustrert ved å gi et eksempel på typisk bruk av en språkkonstruksjon, der bestanddeler i syntaksen er klart identifisert.
- **Grunnleggende datamodellering:** Både grunnleggende *datamodellering* og *programmering* er nødvendige for å lære problemløsning på datamaskinen. Modellering av abstraksjoner og datastrukturer er grundig forklart og illustrert med diagrammer.
- **Algoritmeutvikling:** Boken oppfordrer til algoritmeutvikling, og bruker pseudokode for å vise progresjon fra problemstilling til implementering av løsning.

Presentasjon av konsepter

Boken bruker følgende fremgangsmåter for å presentere og forklare konsepter:

- **Eksempeldrevet fremstilling:** Boken bruker konkrete eksempler for å forklare og anvende konsepter. Hvert program er fullstendig og vises med utdata eller skjermbilde fra kjøring av programmet, slik at man lett kan sammenligne det med egne resultater.
- **Bruk av UML:** Alle viktige konsepter illustreres ved hjelp av UML, og vedlegg G gir en enkel innføring i UML-notasjon.
- **Fokus på problemløsningsteknikker:** Boken bruker noen få utvalgte problemstillinger for å illustrere programmeringskonsepter. Dette gjør at man blir godt kjent med disse problemstillingene, slik at boken kan fokusere på problemløsningsteknikker.

Bruk av boken

Kildekoden til alle programeksempelene i boken er tilgjengelig på nettet, og kan lastes ned og eksperimenteres med. All eksemplene er fullstendige og klare for kjøring, og programkoden har blitt grundig testet på flere plattformer.

Hvert kapittel avsluttes med følgende seksjoner:

- **Kontrollspørsmål:** Disse tester forståelse av temaer fra kapitlet. Fasit med forklaringer er inkludert i vedlegg A.

- **Programmeringsoppgaver:** Disse er av varierende størrelse og vanskelighetsgrad og gir praktisk øvelse i programmering. Fasit til utvalgte oppgaver blir lagt ut etter hvert på bokens nettsted.

I tillegg tilbyr boken:

- **Begrepsnavn på engelsk:** Begrepsnavn er gitt på engelsk i teksten, og en engelsk ordliste bakerst i boken kan brukes til å gjøre oppslag i boken. Denne listen gjør det lettere å foreta litteratursøk i andre bøker og på Internett.
- **Stikkordliste:** En fyldig stikkordliste i slutten av boken kan brukes til å gjøre oppslag i teksten.
- **Kryssreferanser:** Boken bruker kryssreferanser for å gjøre det lettere å knytte begreper til hverandre.

Praktisk bruk av Java

Vi har tatt spesielle hensyn i forbindelse med presentasjon og praktisk bruk av programmeringsspråket Java.

- **Plattformuavhengig programmeringsspråk:** Java oppfordrer til plattformuavhengig programmering, som også er tilfellet i denne boken. Spesifikke plattformavhengige detaljer er kun nevnt der det er nødvendig.
- **Programmeringsverktøyuavhengig fremstilling:** Boken bruker aller siste versjon av Java med standardverktøy (JDK 1.5), som kan lastes ned gratis. Detaljer om kompilering og kjøring av Java-program vha. kommandolinjeverktøyene *javac* og *java* er gitt i vedlegg F. Dersom det er ønskelig, kan andre verktøy eller integrerte utviklingsmiljøer (Integrated Development Environment, IDE) benyttes.
- **Bruk av Javas standardbibliotek:** Alle klasser fra Javas standardbibliotek som er benyttet i boken, er beskrevet med de viktigste metodene fra disse klassene. I tillegg anbefaler vi at man har tilgang til dokumentasjon for Javas standardbibliotek (Application Programming Interface, API) enten via nettet eller installert lokalt.
- **Utforming av dialog mellom programmet og brukeren:** I eksemplene bruker vi ingen skreddersydd klasse for innlesing av verdier fra tastaturet. Men boken tilbyr følgende egenutviklede klasser som kan tas i bruk dersom det er ønskelig, for å utforme dialog med brukeren: En klasse (*Tastatur*) som kan brukes til å lese verdier fra tastaturet og en klasse (*GUIDialog*) for utforming av enkle grafiske brukergrensesnitt for innlesing av data fra brukeren. Klassen *Tastatur* innkapsler bruken av `java.util.Scanner`-klassen, mens klassen *GUIDialog* benytter klassen `javax.swing.JOptionPane`. Utvikling av klassene *Tastatur* og *GUIDialog* blir grundig gjennomgått i boken, og begge klassene tilbyr statiske metoder for å lese inn heltall, flyttall og strenger.
- **Java-fortrinn:** Boken utnytter det Java 2-plattformen har å tilby for et begynnerkurs. Påstander (*assertions*) innføres tidlig i boken. Dynamiske datastrukturer (*Collections*) som mengder, lister og nøkkeltabeller innføres fra Javas standardbibliotek. Utformingen av grafiske brukergrensesnitt (Graphical User Interface, GUI) er utelukkende *Swing*-basert.

Hovedemner i begynnerkurs

Vi har strukturert stoffet slik: OBP (objektbasert programmering, dvs. ingen arv) og OOP ligger i kjernen, og andre emner og aspekter i Java betraktes som *anvendelser* av OBP/OOP innen ulike applikasjonsområder, f.eks. grafiske brukergrensesnitt. Denne fremgangsmåten sørger for at det dannes et grunnlag som benyttes for gjennomføring av øvrige emner.

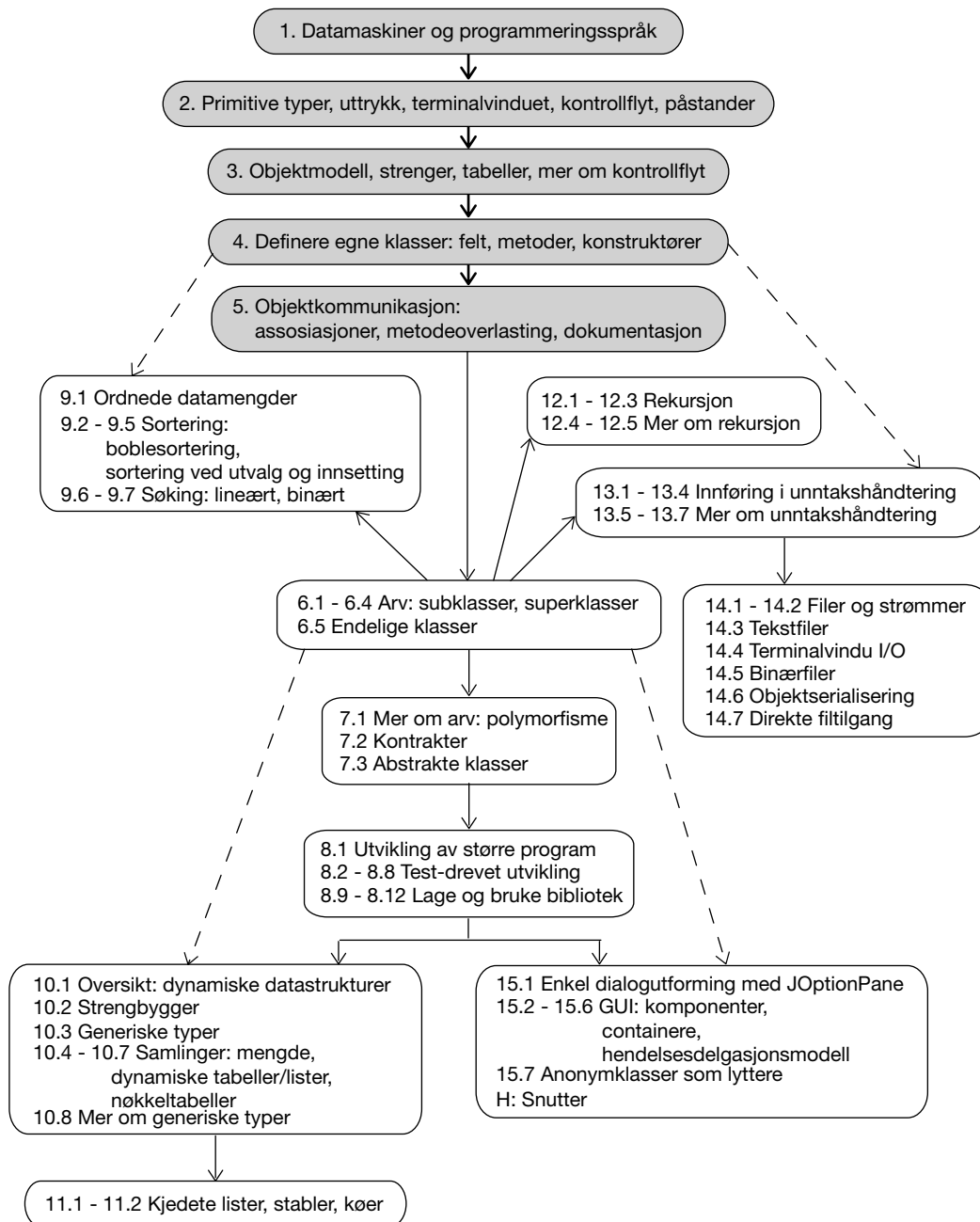
Utvalgte emner fra boken kan brukes til et begynnerkurs (10 studiepoeng), og de øvrige emnene kan danne et videregående kurs (10 studiepoeng). Forslag om hvordan boken kan benyttes for forskjellige typer kurs er vist i avhengighetsgrafene på neste side.

Emner frem til OBP bør dekkes i alle kurs. (Slike emner vises i skyggete bokser i figuren.) Omfanget på kurset kan så varieres ved å velge i hvilken grad de øvrige emnene skal tas med. Vanlige piler viser optimal dekning av stoffet, og stiplede piler viser *snarveier* som kan benyttes for å lage forskjellige kursvarianter.

Strukturert programmering/algorithmutvikling behandles i objektorienteringssammenheng, f.eks. under utforming av metoder. Objektorientering deles i to konseptuelle blokker. Først kommer OBP, som omhandler bruk av ferdiglagde klasser og objekter, og deretter egendefinerte klasser og kommunikasjon mellom objekter av disse klassene. Etter OBP innføres det arv, dvs. *ekte* OOP, som omhandler programmeringsparadigmer som arv har å tilby. Boken viser hvordan Java-språkkonstruksjoner støtter programmeringskonsepter. Den konsentrerer seg om grunnleggende emner, som gir en grundig innføring i OOP, og overlater avanserte emner til en videregående bok.

Etter OOP er det naturlig å diskutere programutviklingsprosessen, og da med stor vekt på testing. Søking- og sorteringsalgoritmer bør være med i enhver programmerers verktøykasse. Et program under kjøring må kunne lagre datasamlinger i minnet. Til det formålet brukes datastrukturer som Java-standardbiblioteket tilbyr. Rekursjon som problemløsingsteknikk hører med i et programmeringskurs. Et program må også kunne utveksle data med omverdenen. For dette formålet blir lagring av data på filer grundig gjennomgått.

Vi legger vekt på applikasjoner (dvs. selvstendige programmer). Vi introduserer GUI som en anvendelse etter at nødvendig bakgrunn er dekket. Vedlegg H gir en enkel innføring i snutter (*applets*).



Forkunnskaper

Boken forutsetter kjennskap til:

- Vanlig datautstyr, dvs. datamaskin med tastatur, mus og skjerm.
- Hvordan man utfører kommandoer i operativsystemet, f.eks. utføring av program fra et kommandolinjevindu.
- Vanlige grafiske brukergrensesnitt med vinduer, knapper og menyer.
- Enkel filhåndtering for å opprette, slette og finne filer i filsystemet.
- En tekstbehandler for å skrive tekstfiler, f.eks. *emacs* eller *vi* på Unix, eller *Notepad* på Windows.
- En nettleser for å navigere på nettet for å hente informasjon.

Konvensjoner brukt i boken

Navn i Java-kildekode

Alle klasse- og kontraktnavn begynner med stor bokstav. I tillegg begynner de fleste kontraktnavn med bokstaven I (for *interface*). Navn på pakker, variabler og metoder begynner med liten bokstav. Konstanter er alltid angitt med store bokstaver. I tillegg slutter alle metodenaavn i teksten med () for å skille dem i fra andre navn.

Kodelinjereferanser i teksten

Ofte vil kodelinjer som forekommer i eksempler eller som kodebiter, ha et tall etter tegnene // på linjen. Dette tallet brukes i teksten til å referere til tilsvarende linje i koden.

Bokens nettsted

Vi har opprettet et nettsted med nyttig tilleggsmateriale. Adressen til bokens nettsted er:

<http://www.ii.uib.no/~khalid/jfps3u/>

Der vil man finne, blant annet:

- Kildekode for alle eksempler fra boken
- Rettelser
- Lenker til andre nyttige ressurser: støttelitteratur, nettlesere, Java-verktøy m.m.

I tillegg blir det lagt ut ressurser som er myntet på forelesere:

- Forelesningsnotater for alle kapitler
- Kildekode for eksempler i forelesningsnotatene
- Lenker til kurs der man kan finne forslag til prosjekter og ukentlige oppgaver

Tilbakemelding

Vi setter stor pris på tilbakemelding. Spørsmål, kommentarer, forslag og rettelser kan sendes til: jfps@ii.uib.no.

Forfattere

Institutt for informatikk, Universitetet i Bergen (UiB) gikk i 1997 over til å bruke Java i sitt innføringskurs i databehandling, I110. Mughal og Rasmussen var ansvarlige for oppbygging av et opplegg for dette kurset. Kvalitetsreformen i høyere utdanning kom høsten 2004, og krevde omlegging av studietilbudet. Mughal var en hovedaktør i planleggingen av programmeringsemnene INF100 og INF101 ved Institutt for informatikk, UiB.

Alle forfatterne har vært med på å utvikle nettbaserte varianter av kursene INF100 og INF101, som nå tilbys regelmessig over Internett (<http://nettkurs.ii.uib.no/jafu/>). Mughal og Hamre har ledet en serie med seminarer om objektorientering ved Institutt for informatikk, UiB. Mughal og Rasmussen er også forfattere av en sertifiseringsbok i Java: *A Programmer's Guide to Java Certification: A Comprehensive Primer (Second edition)* (<http://www.ii.uib.no/~khalid/pgjc2e/>). Forfatterne samarbeider også om forskning innenfor anvendelse av objektorientering.

Hovedforfatter: Khalid Azim Mughal

Khalid A. Mughal er førsteamanuensis ved Institutt for informatikk, UiB. Han har utviklet og gitt kurs for studenter og IT-bransjen om programmering i Java og Java-relaterte teknologier. I 1999 ble han, på grunnlag av begynnerkurset i Java-programmering, tildelt pris som beste foreleser ved Det matematisk-naturvitenskapelige fakultet, UiB. Hans undervisningserfaring omfatter programmeringsspråk, objektorientert systemutvikling, e-læring, databaser og kompilatorkonstruksjon.

Medforfatter: Torill Hamre

Torill Hamre er forskningsleder ved Nansen senter for miljø og fjernmåling i Bergen, med hovedansvar for marin informasjonsteknologi. Hun utvikler objektorienterte løsninger for marine informasjonssystemer. Hun har bistilling som førsteamanuensis ved Institutt for informatikk, UiB.

Medforfatter: Rolf W. Rasmussen

Rolf W. Rasmussen er systemutvikler for firmaet *vizrt* i Bergen, som leverer systemløsninger for fjernsynsbransjen. Han arbeider med kontrollsystemer, videoprosessering, typografi og grafisk sanntidsvisualisering. Han har bidratt til utvikling av GCJ (GNU sin Java-implementering) som er en del av GNU Compiler Collection. Han har også jobbet med *cleanroom*-implementering av Javas grafiske biblioteker.

Vi vil gjerne takke ... (tidligere utgaver)

Først vil vi takke Rasa Ziburkute i Cappelen Akademisk Forlag, som har bidratt sterkt til at denne boken ser dagens lys.

Tilbakemeldingen fra den anonyme konsulenten ved Universitetet i Oslo var meget verdifull. Vi takker hjerteligst for mange konstruktive forslag.

En stor takk til grafikkdesigner Alf E. Andresen, som tok på seg å trylle frem omslaget til boken.

Vi vil gjerne takke følgende for at de tok seg tid til å lese (deler av) manuset og ga oss verdifull tilbakemelding: Harald Soleim, Geir Gundersen, Roland Kaufmann, Sigmund Nysæter, Jan Storenes og Eskil Saatvedt.

En stor takk til Marit S. Mughal for å ha fungert som vår personlige språkrådgiver. At hun hadde tålmodighet til å lese utallige utkast av det vi skrev, er imponerende. Vi er fremdeles ikke sikre på om vi har taket på alle kommareglene, men er takknemlige for de råd vi har fått underveis, både fra Marit og fra forlagets språkkonsulenter.

Vi vil minnes med takknemlighet pionerene Kristen Nygaard og Ole-Johan Dahl, som ga oss Simula og la fundamentet for OOP.

Til slutt vil vi takke våre familier, som har støttet oss under dette arbeidet. Nå kan det bli tid til litt mer familieliv!

Vi vil gjerne takke ... (denne utgaven)

Først vil vi takke forlagssjef Ester Moen i Cappelen Akademisk Forlag, som har vært veldig tålmodig med oss mens vi har jobbet med den nye utgaven.

Vi vil gjerne takke de to tekniske reviewerne Kjetil Jørgensen-Dahl og Steinar Line for mange verdifulle kommentarer, som bidro til å forbedre boken.

Marit Seljeflot Mughal har tatt oss i ørene med rettskrivingen, slik hun har gjort med tidligere utgaver. Vi aner at hun holder på ta steget fra å være en tidligere lærer til å bli en «code reviewer»: Hun oppdaget uoverensstemmelser i programkoden, slik at vi måtte skjerpe oss. Takk, Marit!

Vi vil gjerne rette en takk til alle lesere som har gitt oss tilbakemelding på tidligere utgaver. Denne tilbakemeldingen har bidratt til utformingen av denne utgaven. Det er vanskelig å skille ut enkeltindivider, men vi vil likevel spesielt takke professor Marc Bezem for mange konstruktive forslag, både av faglig og pedagogisk art. Til syvende og sist, er det vi som er ansvarlige for eventuelle feil eller mangler ved denne utgaven.

Vi takker også Institutt for Informatikk, UiB, for et miljø der vi kunne klekke ut denne boken, og anvende den i kurssammenheng.

Det er ikke til å komme utenom at uten støtte fra familiene våre ville denne utgaven tatt enda lengre tid. Takk, folkens! Vi kan dessverre ikke love at det blir siste gang vi går i gang med et bokprosjekt.

Bergen, 17. mai 2006.

Khalid Azim Mughal
Torill Hamre
Rolf W. Rasmussen