

# Afleveringsopgaven

Som grundlag for eksamen på kurset *Objektorienteret programmering* skal der udarbejdes en rapport, der beskriver løsningen af en programmeringsopgave. Nedenfor beskrives kravene til opgaven, og der gives retningslinjer for rapportens udformning.

## 1. Mål

Der skal konstrueres et Java-program, og programmet skal beskrives i en rapport.

## 2. Gruppearbejde

Opgaven løses i grupper på 2 studerende. Det er dog også muligt at løse opgaven individuelt. Grupperne dannes efter de studerendes egne ønsker.

## 3. Valg af opgave

Opgaven vælges af de enkelte grupper ud fra egne interesser. Der er relativt frit spillerum, men der stilles det krav, at opgavens løsning skal være baseret på objektorienterede designprincipper (såsom dekomposition, abstraktion og indkapsling).

To eller flere grupper kan vælge at gå sammen om at løse en større opgave. Men i så fald skal arbejdsdelingen imellem de enkelte grupper være meget skarp.

## 4. Arbejdsomfang

Det forventes at hver studerende bruger cirka 50 arbejdstimer på løsning af opgaven (rapportskrivning iberegnet).

Opgaven forventes løst i løbet af kursusperioden. Der afsættes tid til at arbejde med opgaven ved øvelserne.

## 5. Opgaveformulering

Hver gruppe udarbejder en kort opgaveformulering (maksimalt 2 sider), som kursusrereren skal godkende, før arbejdet påbegyndes.

## **6. Udformning af rapport**

Nedenfor er givet et forslag til rapportens opbygning.

### ***Rapportens form***

Husk en forside med titel, forfatter(e), tid og kursusnavn. Herefter følger følgende hovedafsnit:

- Forord
- Indledning
- Problemstilling
- Brugervejledning
- Programdokumentation
- Afprøvning
- Konklusion

En eventuel litteraturliste skal være sidste afsnit. Rapporten må ikke fylde mere end 20 sider. Programtekst og eventuelle tests vedlægges som bilag.

### ***Forord***

Forordet angiver hvor, hvornår og i hvilken sammenhæng, arbejdet er lavet.

### ***Indledning***

Her motiveres opgaven, og der gives overblik over problemstillingen uden detaljer.

### ***Problemstilling***

Der gives en kort omtale, af den problemstilling, som programmet omhandler, og der opstilles krav til programmet. Der skal være forklaring nok til, at man kan forstå, hvad det er, programmet skal gøre, og hvordan det kan gøre det. Dette afsnit kan eventuelt opdeles i to: *Problemformulering* (Hvad skal programmet kunne?), *Problemanalyse* (Hvordan kan man få programmet til det?).

### ***Brugervejledning***

Her forklares, hvordan programmet fungerer ud fra et brugersynspunkt, altså uden henvisning til tekniske detaljer.

Hvis programmet er vinduesorienteret, bør der være en tegning af brugergrænsefladen, og en forklaring af, hvad de forskellige grafiske komponenter bruges til. Hvis programmet benytter tekstfiler, bør deres format beskrives. Hvis der er begrænsninger i programmet, eller særlige krav til inddata, skal de anføres. Forklar betydningen af fejlmeddelelser fra programmet.

Giv et eksempel på programudførelse, detaljeret nok til at læseren kan køre det selv.

### ***Programdokumentation***

Her beskrives de vigtigste moduler (klasser), datastrukturer (objekter) og algoritmer (metoder). Beskrivelsen kan typisk inddeles i 3 underafsnit:

- Interne datastrukturer: hvordan repræsenteres data i programmet?
- Interne algoritmer: hvordan behandles data i programmet?
- Brugergænsefladen: hvilke klasser består brugergænsefladen af?

Diagrammer kan med fordel benyttes.

### ***Afprøvning***

Her godtgøres, at programmet løser den stillede opgave. Det kan gøres med en brugerprøvning. En systematisk intern eller ekstern afprøvning forlanges ikke.

### ***Konklusion***

Konklusionen skal resumere arbejdet og angive, hvor godt det er lykkedes.

## **7. Aflevering af rapport**

Rapporten skal afleveres i 3 eksemplarer til kursislæreren senest mandag den 13. november 2006 klokken 15<sup>00</sup>. Ved hvert eksemplar skal vedlægges en CD med det udviklede program samt en kørselsvejledning.