

Afleveringsopgave 1

**Afleveres i CampusNet senest:
*Søndag d. 17. februar kl. 23.59***

Jørgen Villadsen

Husk at afleveringsopgaverne er obligatoriske og indgår i kursets samlede bedømmelse.

Husk også at der ofte kan være flere måder at løse opgaverne på.

Herunder er først lidt generel information om afleveringsopgaverne. Det er dog ikke det hele, der er lige relevant for denne lille afleveringsopgave. Specielt er det begrænset, hvor mange test-kørsler, som I kan foretage.

Generel information om afleveringerne

Grupperne

Afleveringsopgaverne skal laves i grupper af 3 personer (undtagelsesvis 2 personer). I danner selv grupperne. Hvis nogen af jer ikke selv kan finde nogen at være i gruppe med, kan I spørge hjælpelærerne eller den kursusansvarlige lærer om hjælp.

Hver gruppe skal vælge en person, som er ansvarlig for at få uploadet gruppens løsning til CampusNet. Det er kun denne person, som skal uploade en løsning. Denne person inviterer så de resterende personer i gruppen, og hver af disse personer skal huske at acceptere invitationen.

Da universitetsloven kræver at alle studerende skal bedømmes individuelt, er det nødvendigt, at hver gruppe skriver lidt om "hvem der har lavet hvad" i afleveringen. Det behøver ikke være meget detaljeret. Det er nok at angive, hvem der har haft hovedansvaret for hvilke dele af afleveringen.

I opfordres under alle omstændigheder til at samarbejde om alle opgaverne frem for at dele dem mellem jer. I kan stadig godt dele hovedansvaret mellem jer, men I får mest ud af at arbejde sammen om tingene, og det vil også give det bedste og mest homogene slutresultat. Undgå at nogen i gruppen kun skriver kode, mens andre kun skriver rapport. Det er vigtigt, at alle lærer begge dele.

Der må ikke samarbejdes på tværs af grupperne, og naturligvis må grupperne ikke få andre til at hjælpe bortset fra hjælpelærerne og den kursusansvarlige lærer.

Aflevering i CampusNet

Opgavebesvarelsen skal afleveres i CampusNet. Det foregår på den måde, at man går ind under kursets CampusNet-side og dernæst vælger ‘Opgaver’ i menuen til venstre. Man får mulighed for at uploade filerne ved at vælge ‘Besvar’ for den pågældende afleveringsopgave.

Afleveringen skal bestå af to dele:

1. En **pdf-fil** med rapport-delen af opgavebesvarelsen. På forsiden af rapporten skal der stå følgende:
 - Kursusnummer, kursusnavn og nummeret på afleveringsopgaven.
 - Studienumre og navne på alle gruppens medlemmer.
 - En kort beskrivelse af arbejdsdelingen i gruppen (“hvem der har lavet hvad”) — se afsnittet om grupperne ovenfor.

Af kompatibilitetshensyn er det vigtigt, at rapporten ligger i en pdf-fil og ikke en doc-fil.

2. En **zip-fil** indeholdende kildekoden til alle de programmer, som I har skrevet. I skal kun uploade de `.java`-filer, som I selv har skrevet, altså ingen af de udleverede filer, og I skal heller ikke uploade jeres `.class`-filer. Af kompatibilitetshensyn er det vigtigt, at formatet er `.zip` og ikke `.rar` eller `.tar` eller andre formater, tak! I skal også sørge for, at zip-filen ikke genererer nogen underkataloger, når den udpakkes.

Bemærk at rapporten skal uploades som en separat pdf-fil, altså ikke i zip-filen.

Udformning af besvarelsen

Herunder er nogle hints til, hvad vi forventer af jeres opgavebesvarelse:

Korrekthed af programmerne. Det er naturligvis vigtigt at jeres programmer virker korrekt og i alle situationer gør det, som opgaven foreskriver.

Strukturering af programmerne. Det er lige så vigtigt at jeres programmer er struktureret godt, både med henblik på at reducere redundans, med hensyn til at gøre koden overskuelig, og med hensyn til at gøre den let at modificere. Brug statiske metoder hvor det kan hjælpe til at give programmerne en fornuftig struktur.

God kodestil og passende kommentering af programmerne. Det er vigtigt at give jeres variable, konstanter og metoder meningsfulde navne, som hjælper til at forstå programmerne. Sørg desuden for at overholde de gængse konventioner for navngivning og indrykning m.m. Undlad venligst at bruge de danske bogstaver “æøå” i navne i selve koden. Det er vigtigt at jeres programmer er kommenteret fornuftigt. Man skal passe på ikke hverken at underdrive eller overdrive brugen af kommentarer. For at finde det rigtige niveau kan I tænke på, hvilke kommentarer I tror, der vil være nødvendige, for at en af jeres med-studerende umiddelbart vil kunne forstå jeres program ved blot at læse kildekoden med de inkluderede kommentarer.

Overordnet beskrivelse af programmerne i rapporten. I rapporten bør I give en overordnet beskrivelse af jeres programmer. Det skal ikke være en meget detaljeret gennemgang af programmerne — en sådan vil fremgå af selve koden med de inkluderede kodekommentarer. Rapporten skal blot indeholde en overordnet og nogenlunde kortfattet beskrivelse af, hvordan I har valgt at designe jeres programmer. Hvis I har overvejet flere forskellige løsningsmetoder, kan I eventuelt angive dem, og fortælle hvorfor I har valgt de løsninger, som I har. I bør naturligvis også angive, hvis jeres programmer har funktionaliteter udover det, som opgaveteksten foreskriver. Omvendt skal I også skrive, hvis jeres programmer har nogle begrænsninger i forhold til opgaveteksten, f.eks. ikke virker på alle input.

Dokumentation af test-kørsler i rapporten. Alle programmer, som I skriver, skal afprøves på nogle velvalgte inputs for at checke, om de virker efter hensigten. Resultaterne af disse test-kørsler skal inkluderes i rapporten. Det letteste er blot at lave copy-paste fra Eclipses konsol ind i rapporten. Knyt et par kommentarer til hver af jeres test-kørsler. Sørg for at jeres test-kørsler får afprøvet så mange af programmets “features” som muligt.

Det er vigtigt, at I er opmærksomme på, at alle ovennævnte faktorer indgår med afgørende betydning i vurderingen af jeres besvarelse. Dokumentation og strukturering er lige så vigtigt, som at selve programmerne virker.

Der er ikke nogen egentlig hverken nedre eller øvre grænse for rapportens længde.

Feedback på jeres løsninger

Når jeres opgaver er blevet rettet, vil I modtage skriftlig feedback i CampusNet under selve afleveringen, så I kan se, hvad I har gjort godt og skidt, hvad der eventuelt kunne forberedes osv. Det er vigtigt, at I tænker over den feedback, som I får, så I kan forbedre jer i næste aflevering.

Opgave 1

Skriv et program `Opgave1`, der udskriver følgende linjer:

```
Use "\\" to obtain a 'backslash' character.  
Remember:  
Use "\\" to obtain a 'backslash' character.
```

Husk at eliminere eventuel redundans.

Opgave 2

Betragt følgende ukomplette program.

```
public class Opgave2 {  
    public static void main(String[] args) {  
        // ???  
        System.out.println(1 + 3 + x + x + x + x + x + 5 / 4 * 2);  
    }  
}
```

Erstat kommentaren med en variabel-deklaration, så programmet udskriver følgende linje:

42

Tip: Der skal ikke deklareres en heltals-variabel.

Opgave 3

Betragt følgende program.

```
public class Opgave3 {  
    public static final int LINES = 3;  
  
    public static void main(String[] args) {  
        for (int i = 1; i <= LINES ; i++) {  
            for (int j = 1; j <= LINES; j++) {  
                System.out.print ("+-");  
            }  
            System.out.println("+");  
        }  
    }  
}
```

Foretag en lille ændring i programmet, så det udskriver følgende linjer:

```
+--+--+  
+--+--+  
+--+--+
```

Kør også det ændrede program med forskellige værdier af klassekonstanten, specielt 0, 1 og 2.