

INF1000 - Obligatorisk innlevering 3

Frist: 18. September kl 22:00

Temaer denne uka: *metoder, beslutninger, kodeflyt, arrayer og while-løkker.*

Oppgave 3.1)

Tema: *Beslutning, programflyt og test-caser*

Filnavn: `Timeplanlegger.txt`

Til denne oppgaven skal du besvare spørsmålene under i en tekstfil. I et timeplanleggingssystem ønsker vi å sjekke om to forelesninger overlapper. For enkelhets skyld starter alle forelesninger på hele klokkeslett fra 0 til 23. Følgende kodeutsnitt er ment å sjekke om forelesningen med starttidspunkt **start1** og sluttidspunkt **slutt1** overlapper med forelesningen med starttidspunkt **start2** og sluttidspunkt **slutt2**:

```
public static void sjekkOverlapp(int start1, int slutt1, int start2, int slutt2){
    if(slutt1 <= start2 || slutt2 <= start1){
        System.out.println("Forelesningene overlapper ikke.");
    } else {
        System.out.println("Forelesningene overlapper.");
    }
}
```

- Beskriv programflyten med ord når den første forelesningen er fra 10 til 12 og den andre er fra 11 til 13.
- Beskriv programflyten med ord når den første forelesningen er fra 10 til 11 og den andre er fra 12 til 13.
- Hvilke andre tidspunkter ville det være lurt å sjekke med denne metoden? Begrunn svaret ditt.

Synes du denne oppgaven var vanskelig? Se øvingsoppgaver R3.9, R3.10 og P3.18.

Oppgave 3.2)

Tema: *While-løkker og arrays*

Filnavn: Datastruktur

Lag en tegning som viser innholdet i arrayen som er definert i koden under, etter endt kjøring. Tegningen kan du levere i ett av følgende formater: JPEG, PNG eller PDF.

```
class Arrays {
    public static void main(String[] args) {
        int array[] = new int[5];
        int teller = 0;
        int verdi = 0;

        while(teller < 5) {
            array[teller] = verdi + 1;
            teller = teller + 1;
            verdi = verdi + 1;
        }
    }
}
```

Oppgave 3.3)

Tema: *Bruke løkker til å gå gjennom array*

Filnavn: NegativeTall.java

Lag et program som inneholder arrayen

```
int[] heltall = {1, 4, 5, -2, -4, 6, 10, 3, -2};
```

- Tell hvor mange negative tall det er i arrayen ved hjelp av en while-løkke og skriv resultatet i terminalen.
- Erstatt alle negative tall med deres posisjon i arrayen ved hjelp av en while-løkke (hvor posisjoner telles fra 0 og oppover).
- Print ut arrayen til terminal.

Synes du denne oppgaven var vanskelig? Se øvingsoppgaver 3.06.

Synes du denne oppgaven var lett? Se utfordringsoppgaver 3.07 og 3.08.

Oppgave 3.4)

Tema: Metoder med parametere, test-caser og løkker

Filnavn: Alternativ1.java ELLER Alternativ2.java

I denne oppgaven kan du velge mellom to alternative oppgaver, hvor du skal velge én av de. Alternativ 1 er antatt lettest, mens alternativ 2 er antatt vanskeligst. Om du ønsker å løse og få tilbakemelding på begge må du gjerne gjøre det, men du vil kun få ett poeng på denne oppgaven.

Alternativ 1)

a) Skriv en metode

```
static void skrivTekster(String tekst, int antall)
```

Metoden skal skrive ut teksten i parameteren `tekst` så mange ganger som angitt av variabelen "antall". Hver utskrift av `tekst` skal komme på en egen linje.

- b) Beskriv først med ord hvordan du vil endre/utvide metoden slik at det for hver tredje gang `tekst` skrives ut, også skal skrives ut et ekstra linjeskift/en tom linje. Skriv dette som en flerlinjekommentar ved metodesignaturen. Implementer deretter denne endringen/utvidelsen i metoden.
- c) Skriv en main-metode med ett eller flere kall på `skrivTekster` med passende testdata for å sjekke at metoden fungerer som den skal i alle tilfeller.

Alternativ 2)

a) Skriv en metode

```
static void skrivKarakterprotokoll(int[] poeng)
```

som får som parameter en array med poengsummene til et antall studenter med kandidatnummer fra 0 og oppover så lang arrayen er. Hver student kan ha fått fra 0 til maksimalt 40 poeng. Metoden skal skrive en liste over alle studentene med kandidatnummer og karakter basert på følgende tabell (fra nasjonal eksamen i matematikk for grunnskolelærerutdanningene):

- A: 36-40 poeng
- B: 31-35 poeng
- C: 23-30 poeng
- D: 18-22 poeng
- E: 16-17 poeng
- F: 0-15 poeng

Om én av poengene er under 0 eller over 40, skriv ut en passende feilmelding til terminalen.

For eksempel, om arrayen som blir sendt som argument til metoden inneholder tallene 10, 20, 30 og 40 skal programmet gi følgende utskrift:

```
> Kandidatnummer 0 : F
> Kandidatnummer 1 : D
> Kandidatnummer 2 : C
> Kandidatnummer 3 : A
```

Og om arrayen inneholder tallene 37, 25, 20, 31, 33, 2, 38, 40, 25, 24, 20, 23 og 40 skal programmet gi følgende utskrift:

```
> Kandidatnummer 0 : A
> Kandidatnummer 1 : C
> Kandidatnummer 2 : D
> Kandidatnummer 3 : B
> Kandidatnummer 4 : B
> Kandidatnummer 5 : F
> Kandidatnummer 6 : A
> Kandidatnummer 7 : A
> Kandidatnummer 8 : C
> Kandidatnummer 9 : C
> Kandidatnummer 10 : D
> Kandidatnummer 11 : C
> Kandidatnummer 12 : A
```

- b) Skriv en main-metode med ett eller flere kall på skrivKarakterprotokoll med passende testdata for å sjekke at metoden fungerer som den skal i alle tilfeller.

Oppgave 3.5)

Tema: *Egen oppgave*

Filnavn: MinOppgave3.java

- a) Skriv en egen oppgavetekst som handler om arrayer og/eller løkker. Ta gjerne utgangspunkt i array-algortimene i boka (kapittel 6.3)
- b) Løs oppgaven du skrev i a).

NB! Husk at du skal levere både løsningen og oppgaveteksten du skrev i a).

Krav til innleveringen

1. Klassenavnet og filnavnet skal være identisk.
2. Klassenavn skal skrives med stor forbokstav.
3. Variabelnavn skal ha liten forbokstav.
4. Oppgaven må kunne kompilere og kjøre på IFI sine maskiner.
5. Kun .java-filen skal innleveres.
6. Ikke bruk æ, ø eller å i .java-filene(heller ikke som kommentarer eller utskrift).
7. Filene skal inneholde gode kommentarer som forklarer hva programmet gjør.
8. Programmet skal inneholde gode utskriftssetninger som gjør det enkelt for bruker å forstå.

Fremgangsmåte for innleveringer i INF1000

1. Lag en fil som heter README.txt. Følgende spørsmål skal være besvart i filen:
 - (a) Hvordan synes du innleveringen var? Hva var enkelt og hva var vanskelig?
 - (b) Hvor lang tid (ca) brukte du på innleveringen?
 - (c) Var det noen oppgaver du ikke fikk til? Hvis ja:
 - Hvilke(n) oppgave er det som ikke fungerer i innleveringen?
 - Hvorfor tror du at oppgaven ikke fungerer?
 - Hva ville du gjort for å få oppgaven til å fungere hvis du hadde mer tid?
2. Logg inn på [Devilry](#).
3. Lever de 3 .java-filene (**ikke .class**), TXT til opg. 3.1, PDF/JPG/PNG til opg. 3.2, samt README.txt i *samme innlevering*.
4. Husk å trykke lever og sjekk deretter at innleveringen din er komplett.

Den obligatoriske innleveringen er minimum av hva du bør ha programmert i løpet av en uke. Du finner flere oppgaver for denne uken [her](#) og flere utfordringsoppgaver [her](#).