Frivillig øving - Versjonskontroll og testing

Denne øvingen er **ikke obligatorisk**, men anbefales løst.

Oppgave 1

Gjennomfør eksemplet under Enhetstesting med JUnit i leksjonen om enhetstesting. I eksemplet bruker vi IntelliJ som utviklingsmiljø, men prosessen vil være tilsvarende om du velger å bruke et annet IDE.

Oppgave 2

Sett prosjektet fra oppgave 1 under versjonskontroll:

1. Åpne et terminalvindu og naviger deg til rotkatalogen for prosjektet.
2. Initialiser prosjektet som et Git-repository med `git init`.
3. Legg til filene du mener du må ha kontroll på med `git add` slik at de er klare for commit (staging).
4. Sjekk inn filene med `git commit -m "Beskrivende kommentar"`.

Oppgave 3

Utvid klassen `DateUtils` med følgende kodesnutt:

```
1 public static long daysBetween(LocalDate startDate,
  LocalDate endDate, boolean includeEndDate) {
2   long days = ChronoUnit.DAYS.between(startDate,
  endDate);
3   if (includeEndDate) {
4     if (days >= 0) { return days+1; }
5     else { return days-1; }
6   } else {
7     return days;
8   }
9 }
```

Metoden `daysBetween` returnerer antall dager mellom to datoer. Hvis flagget `includeEndDate` er satt til `true` vil også sluttdato tas med i beregningen. Hvis sluttdato er før startdato vil et negativt antall dager returneres.

For at koden skal kompilere må du importere følgende klasser:

```
import java.time.LocalDate;
import java.time.temporal.ChronoUnit;
```

Verifiser at alt kompilerer og at eksisterende tester kjører uten feil. Sjekk deretter inn endringen med Git (add + commit).

Oppgave 4

Skriv tester for den nye metoden. Følg prinsippene fra leksjonen (Arrange-Act-Assert, positiv og negativ testing mm). Det er opp til deg hvordan du ønsker å strukturere testene. En mulighet er å refaktorere i to separate klasser:

- Eksisterende tester for `DateUtils.isLeapYear` legges i en ny klasse kalt `IsLeapYearTest`
- Tester for `DateUtils.daysBetween` legges i en ny klasse kalt `DaysBetweenTest`

Gjør commits underveis. Det er bedre å sjekke inn en gang for mye enn en gang for lite. Før du sjekker inn bør du alltid verifisere at koden kjører som den skal og at testene ikke feiler.