

Opgaver tirsdag den 21. februar

Opgave 1

I denne opgave skal du lave en JUnit test til klassen BilForsikring. Klassen er givet ved nedenstående specifikation og koden kan findes i filen bilforsikring.zip på Canvas: Læg mærke til at metoden beregnPræmie er afhængig af, at variablen grundPræmie indeholder en gyldig værdi.

```
public class BilForsikring {
    // angiver den til enhver tid gældende
    // grundpræmie for en bilforsikring
    private double grundPræmie;

    /**
     * Registrerer bilforsikringens grundPræmie
     */
    public void setGrundPræmie(double grundPræmie) {
    }

    /**
     * Returnerer bilforsikringens grundPræmie
     */
    public double getGrundPræmie() {
    }

    /**
     * Beregner og returnerer en præmie udregnet ud fra følgende regler:
     * grundPræmie danner udgangspunkt for præmien
     * hvis der er tale om unge under 25 tillægges grundPræmien 25%
     * hvis der er tale om en kvinde reduceres præmien med 5%
     * hvis man har kørt skadefrit i:
     *     0 til 2 år reduceres præmien med 0%
     *     3 til 5 år reduceres præmien med 15%
     *     6 til 8 år reduceres præmien med 25%
     *     over 8 år reduceres præmien med 35%
     * ovenstående skal udregnes i den angivne rækkefølge
     * Hvis parametrene ikke er indenfor det gyldige område
     * kastes en exception med en passende tekst
     *
     * Krav: grundPræmie er tildelt værdi.
     */
    public double beregnPræmie(int alder,
                               boolean isKvinde,
                               int skadeFrieÅr) {
    }
}
```

Lav testcases i JUnit så du får sikret at beregnPræmie regner rigtigt og får kastet de rette exceptions.

Opgave 2

Følgende er taget fra Århus Kommunes Bibliotekers hjemmeside og viser gebyrer ved for sen aflevering af materialer (i en forsimplet udgave – der er faktisk flere dagsintervaller). Bøden er den samme, hvis flere materialer afleveres for sent. Det kan antages, at alle afleverede materialer har samme tilbageleveringsdato, og der skal ikke tages hensyn til søn- og helligdage.

Overskridelse	Børn og unge 0 til 17 år	Voksne fra 18 år
1 – 7 dage	10 kr.	20 kr.
8 – 14 dage	30 kr.	60 kr.
15+ dage	45 kr.	90 kr.

Til beregning heraf laves følgende metode.

```
/*  
 * Returnerer størrelsen af bøden beregnet i henhold til skemaet  
 * ovenfor  
 * krav: beregnetDato og faktiskDato indeholder lovlige datoer og  
 *       beregnetDato < faktiskDato  
 *       (beregnetDato er forventet afleveringsdato og  
 *       faktiskDato er den dag bogen blev afleveret; voksen er  
 *       sand, hvis det er en voksen og falsk ellers)  
 */  
public int beregnBøde(LocalDate beregnetDato,  
                     LocalDate faktiskDato, boolean voksen)
```

Programmer metoden og test den under anvendelse af JUnit.

Opgave 3

I denne opgave skal du lave en test case af klasserne Studerende og Skole fra den 10. februar opgave 2.1. Husk at såvel associeringen mellem klasserne som alle metoder på klasserne skal testes.

Opgave 4

Programmer unit tests til "beregnet samlet pris"-metoden fra dit KAS-projekt fra 1. semester. Benyt gerne testdata fra noten "Test-data til KAS-projekt.pdf" som blev udgivet under KASprojektet.

Opgave 5

I denne opgave skal du implementere en Bag. En bag minder om et set, men med den forskel, at det er tilladt at have dubletter. For hvert element der tilføjes bag'en, skal der derfor holdes styr på hvor mange gange elementet forekommer i bag'en.

Implementationen skal være baseret på et HashMap, og have følgende interface:

```
public interface Bag {  
    /** Add a String to the bag */  
    public void add(String s);  
    /** Remove a String from the bag */  
    public void remove(String s);  
    /** Get the count of Strings equal to s in the bag */  
    public int getCount(String s);  
}
```

Lav det der skal til, så din nye Bag også kan anvendes direkte i et "for each" loop i Java. Elementer der er tilføjet flere gange til Bag, skal kun optræde én gang i gennemløbet.