

## Opgave i Objektorienteret Programmering

### CPR

Denne opgave går ud på at skrive en række metoder til en klasse og at af teste disse. Klassen skal kunne håndtere CPR-numre og eneste data member eller attribut i klassen skal defineres som følger:

```
private String nummeret;
```

Det er en del af opgaven at transformere tekstdata til numerisk data<sup>1</sup>, så det er **forbudt** at tilføje ekstra (numeriske) attributter eller at ændre attributtens datatype.

Klassen skal hedde CprNr.

#### Beskrivelse af metoderne

```
public CprNr(String etCprNr)
```

Constructor, som sætter *nummeret* = cifrene i *etCprNr*, f.eks. "0502201972" (Farmors CPR-nr).

```
public String getCprNr()
```

returnerer CPR-nummeret.

```
public void setCprNr(String etCprNr)
```

sætter *nummeret* til *etCprNr*.

```
public int getDag()
```

returnerer dagen i CPR-nummerets dato som et element af datatypen integer.<sup>2</sup>

```
public int getMaaned()
```

returnerer måneden i CPR-nummerets dato som et element af datatypen integer.

```
public int getAar()
```

returnerer året i CPR-nummerets dato som et element af datatypen integer på formen YYYY.

Med hensyn til datoens århundrede er dette indlagt i CPR-nummerets 7. ciffer på følgende måde:

0 – 3	1900-tallet
4 – 5	1800-tallet
6 – 9	2000-tallet

og eftersom 7. ciffer i Farmors CPR-nummer er 1, passer det fint med, at hun naturligvis er født i 1920.

```
public Dato getDato()
```

returnerer CPR-nummerets dato på formen YYYYMMDD som et objekt af klassen Dato.

```
public int getAlder()
```

returnerer personens alder i år.

```
public boolean erMand()
```

returnerer *true*, hvis CPR-nummerets indehaver er en mand (sidste ciffer ulige) ellers *false*.

```
public boolean erKvinde()
```

returnerer *true*, hvis CPR-nummerets indehaver er en kvinde (sidste ciffer lige) ellers *false*.

---

<sup>1</sup> I Javas String-klasse findes en metode, som hedder *charAt(int)*. Kaldt på et String-objekt returnerer den karakteren (char) på parameterens plads. Husk indeksering starter altid med indeks 0.

<sup>2</sup> Følgende statement: `Integer.parseInt(nummeret.substring(2, 4))`; returnerer karaktererne på plads 2 og 3 i String-variablen *nummeret* som int.

**public boolean erValid()**

returnerer *true*, hvis CPR-nummeret er korrekt ellers *false*.

Først skal der tjekkes på, at CPR-nummerets længde er ti tegn og at alle tegn er numeriske.

Dernæst skal der naturligvis være tale om en valid fødselsdato (de første 6 cifre).

Endelig skal der udføres et *modulus-11 check*. Alle CPR-numre er opbygget således, at det 10. og sidste ciffer er et såkaldt checkciffer, hvis værdi man altid kan beregne, hvis man har de foregående ni cifre. Da vi imidlertid forudsætter, at vi altid har alle CPR-nummerets 10 cifre til rådighed, behøver vi ikke beregne checkcifferet, men blot kontrollere, at det er korrekt. Det foregår ved at gange hvert enkelt ciffer med en forudbestemt vægt og dernæst finde summen af de herved fremkomne tal. Hvis denne sum er delelig med 11, er checkcifferet korrekt. Et eksempel på beregning er vist i nedenstående tabel:

Ciffer	0	5	0	2	2	0	1	9	7	2
Vægt	4	3	2	7	6	5	4	3	2	1
Udregnet	0	15	0	14	12	0	4	27	14	2

Summen af tallene i rækken "Udregnet" er 88, som er deleligt med 11.