

Opgave i objektorienteret programmering i C++¹

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

```
private string nummeret;
```

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

Beskrivelse af memberfunktionerne (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 *int*.

```
public int getMaaned()
```

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

```
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 blev født i 1920.

```
public int getDato()
```

returnerer CPR-nummerets dato på formen YYYYMMDD som et element af datatypen *int*.

```
public int getAlder()
```

returnerer personens alder i år.

¹ Opgaveformuleringen er af ældre dato og reglerne for opbygning af CPR-nummeret er siden blevet ændret.

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*.

public boolean isValid()

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

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

Herefter skal man bruge funktioner fra dato-klassen til at validere dato-delen af CPR-nummeret

med. Funktionen skal desuden udføre et såkaldt *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.

Endelig skal der skrives en main-metode (fx i Source.cpp), som tester klassen.