

Opgave i Java-programmering

Denne opgave omhandler objektorienteret programmering ved hjælp af klasser og strukturer mellem klasser (objekter) - i denne opgave aggregeringsstrukturen.

På det lokale fjernvarmeværk ønsker man at få lavet et lille it-system, som kan anvendes til at afregne fjernvarmeforbruget for de enkelte forbrugere i et distrikt. Afregningen sker én gang om året, idet forbrugeren aflæser måleren og indsender den aflæste værdi til fjernvarmeværket.

I objektmodellen indgår to klasser: *Distrikt* og *Forbruger*, således at 0 til 200 forbrugere indgår i et distrikt.

1.

Første del af opgaven går ud på at programmere klassen *Forbruger*. Klassen har følgende attributter og metoder:

Attributter/Datamembers

<i>navn</i>	Navnet på en forbruger. Tekst.
<i>målerNr</i>	Målnummer for en fjernvarmemåler. Heltal. Målnummer identificerer et forbruger-objekt.
<i>nyAflæsning</i>	Målværdien (i kubikmeter varme) ved seneste aflæsning. Heltal.
<i>forrigeAflæsning</i>	Målværdien (i kubikmeter varme) ved forrige aflæsning. Heltal.
<i>målerMax</i>	Maksimal mulig værdi på måler. Når måleren når denne værdi, begynder den forfra fra nul (overløb). Heltal.

Metoder

<i>Forbruger</i>	Default constructor.
<i>Forbruger</i>	Constructor, som medtager forbrugers navn, et målnummer og målerens maksimalværdi som parametre.
<i>getMålerNr</i>	Access metode til datamember <i>målerNr</i> .
<i>setNavn</i>	Mutator metode til datamember <i>navn</i> .
<i>getNavn</i>	Access metode til datamember <i>navn</i> .
<i>aflæsMåler</i>	Som input-parameter har metoden forbrugers aflæsning af måler-værdien. Metoden skal opdatere datamembers <i>nyAflæsning</i> og <i>forrigeAflæsning</i> .
<i>beregnForbrug</i>	Metoden skal returnere differencen mellem <i>nyAflæsning</i> og <i>forrigeAflæsning</i> , idet der tages højde for eventuelt overløb af måleren.

Skriv klassesdefinitionen for klassen *Forbruger*, som indeholder ovenfor angivne attributter og metoder.

2.

Anden del af opgaven går ud på at programmere klassen *Distrikt*. Klassen har følgende attributter og metoder:

Attributter/Datamembers

<i>navn</i>	Distriktets navn. Tekst.
<i>prisPrM3</i>	Prisen pr kubikmeter (m3) varme. Decimaltal.
<i>forbrugere</i>	Her er forbrugerne aggregeret op i distrikt-objektet ved indlejring, det vil sige en tabel med plads til 200 objekter af klassen <i>Forbruger</i> .
<i>antalForbrugere</i>	Angiver hvor mange forbrugere, der er oprettet i distriktet. Heltal.

Metoder

<i>Distrikt</i>	Default constructor.
<i>Distrikt</i>	Constructor. Navn og pris pr. kubikmeter varme som parametre.
<i>setForbruger</i>	Opretter en forbruger i distriktet. Har forbrugeren med som parameter.
<i>afregnForbruger</i>	Metoden skal beregne prisen for et årsforbrug for én forbruger. Metoden har forbrugerens målnummer med som inputparameter og skal returnere prisen. Hvis målnummeret ikke kan findes, returneres et negativt tal (-1).

Skriv klassesdefinitionen for klassen *Distrikt*, som indeholder ovenfor angivne attributter og metoder.

3.

Tredje del af opgaven går ud på at skrive et main-program, som aftester klasserne *Forbruger* og *Distrikt*.