

Terminsprøve

VEADM18D18X, Programmering 1, 14/03/2018

Varighed 09:30 - 12:00 (2½ time)

Hjælpemidler Alt er tilladt.

Aflevering Relevante afleveres på Fronter som zip mappe.

Hvis der i en opgave er uklarheder, eller du synes der mangler oplysninger, antager du blot de ting som du efterlyser, i nødvendigt omfang, så længe dine antagelser ikke ødelægger helheden i opgaven.

Hver opgave laves som en separat konsol applikation, med passende navn såsom "Opgave1", "Opgave2" (eller "Assignment1", "Assignment2") etc. Opgaverne 5 og 6 (der bygger på opgave 4) laves bare som ændringer i opgave 4. Opgave 7 kan afleveres som et simpelt tekst, Word, eller PDF dokument.

```
terminsprove/
├── Opgave1
│   ├── ...
│   └── Program.cs
├── Opgave2
│   ├── ...
│   └── Program.cs
├── Opgave3
│   ├── ...
│   └── Program.cs
├── Opgave4
│   ├── ...
│   └── Program.cs
└── Opgave7.txt
```

Løsninger bliver lagt op på [GitHub](#) (uden navn, medmindre andet aftales) og gennemgås fredag d. 16/03.

Opgave 1

Konstruer et lille program der kan danne og udskrive alle nedenstående 3 talrækker ved hjælp af loops:

- A. 7, 14, 21, 28, 35, 42, 49, 56, 63 (7-tabel)
- B. 1, 5, 9, 13, 17, 21, 25, 29, 33, 37 (læg 4 til)
- C. 8000000, 800000, 80000, 8000, 800, 80, 8 (divider med 10)

Opgave 2

Konstruer en statisk metode der kan undersøge om et tal opfylder følgende regler:

- Tallet skal være større end nul
- Tallet skal være mindre end 10.000

Det vil sige tallet 9875 er ok men tallene -2000 og 12.000 er ikke OK. Metoden skal returnere en passende værdi der signalerer om tallet er OK eller ikke OK.

Opgave 3

Konstruer en ny klasse *Employee* (medarbejder), der har de følgende tre attributter:

- *FirstName*
- *LastName*
- *MonthlySalary*

Disse attributter skal sættes ved hjælp af en *constructor* metode på *Employee*.

Lav nu to metoder på *Employee*:

- *GetFullName()* der returnerer medarbejderens fulde navn
- *GetSalary(int n)* der returnerer medarbejderens løn for *n* måneder

Dit program skal udskrive alle oplysningerne om en medarbejder, ved brug af metoderne på *Employee*.

Opgave 4

Et brev kan sendes som A-post eller som B-post. A-post er i denne opgave ALTID 1,50 kr dyrere end B-post uanset hvad brevet vejer. B-post er breve hvor man ikke kan garantere at brevet er fremme dagen efter men blot inden for 2-3 hverdage (derfor lidt billigere).

Hvis et brev vejer mellem 0 og 50 gram skal portoen beregnes som 8,00 kr, hvis brevet vejer mellem 50 og 100 gram er portoen 12,50 kr og endelig hvis brevet vejer mellem 100 gram og 1000 gram er portoen 35,00 kr.

Konstruer et program der beregner porto prisen for et brev. Hvis brevet vejer mere end 1000 gram skal programmet give en melding om at der ikke er tale om et brev men en pakke.

Opgave 5

En udvidelse af opgave 4. Beregn og udskriv nu portoprisen for 8 breve, samt den totale pris for alle 8 breve til sidst.

Opgave 6

En udvidelse af opgave 5. Omskriv dit program til, at benytte en ny *Delivery* klasse, som du selv definerer ud fra de egenskaber, som programmet skal bruge. *Delivery* klassen skal repræsentere hvad vi tidligere har kaldt "et brev".

Opgave 7

Skriv dine tanker omkring følgende:

- Hvad er en klasse?
- Hvad er et objekt?
- Hvad er formålet med en *constructor*?
- Hvorfor benytter vi klasser og objekter?