

# **MockEksamen/ Bunden opgave**

**i fagene**

**Programmering & Teknik**

**2018 September**

**Timeforbrug 3-6 timer**

**Start dato 27. september kl 19.00**

**Afleveringsdato 5. oktober kl. 19.00**

Alle skriftlige materialer, pc'er, bærbare computere og internet ressourcer er tilladt til eksamen.

Ved periodens afslutning skal du lægge din løsning op i Wiseflow.

Opgaven kan typisk løses på 4 timer. Måske ikke 100% korrekt, men rigeligt til at bestå.

Du må ikke benytte/gemme din løsning på eksterne netværksdrev/værter som Github, Facebook o.lign.

Normalt skal nedenstående regler (*med kursiv*) følges, MEN i denne opgave –pga de særlige omstændigheder omkring flytningen- fraviges disse regler.

*Mobiltelefoner og kommunikation med andre personer (incl. chatbots), bortset fra kommunikation med eksaminator og censor, er forbudt.*

*Ved eksamens afslutning skal du lægge din løsning op i Wiseflow.*

*Eksamen varer 4 timer og efterfølges af 1 times evaluering. I disse tidsrum kan den studerende kun forlade eksamenslokalet med henblik på toilet besøg. Rygning er ikke tilladt.*

*Ud over disse opgaver, kan du kan blive bedt at besvare nogle ekstra spørgsmål omkring dine svar og eventuelle andre valg.*

Eksamenssættet består af 5 opgaver.

**Kontroller at dette sæt indeholder 5 sider inklusiv forsiden.**

## **Bil Afgift**

I denne eksamen skal du arbejde med Library, Test, TCP og Typescript. Denne eksamen handler om et system til beregning og administration af bilregistreringsafgift i Danmark.

Udover at programmere dette system bliver du undervejs bedt om at redegøre for overvejelser og nedskrive forklaringer. Dette samler du i et pdf-dokument, som uploades til Wiseflow sammen med dine programmer i en zip fil. Zip filen bør have navnet "Mock 1 <dit navn fødselsår>".

### **Domæne beskrivelse**

Skat ønsker et nyt system til beregning af registreringsafgiften for personbiler, og det er din opgave at udvikle dele af dette. Mere specifikt skal du programmere et system, som i første omgang kun kan udregne registreringsafgiften for almindelige personbiler og elbiler jf. taksterne i nedenstående tabel:

<b>Køretøj</b>	<b>Registreringsafgift</b>
Personbil	Afgiften er 85 % af prisen hvis prisen er mindre end eller lig med 200000 kr. Ellers anvendes en større procentsats 150%, se formelen nedenfor
Elbil	Afgiften beregnes først som for en personbil. Men af denne beregnede afgift skal kun betales 20 %.

Registreringsafgiften for en personbil kan derfor beregnes efter flg. formler:

$\text{bilAfgift} = \text{pris} \times 0.85$ ; hvis pris er mindre end eller lig med 200000.

$\text{bilAfgift} = (\text{pris} \times 1.50) - 130000$ ; hvis pris er større end 200000.

Registreringsafgiften for en elbil kan beregnes efter flg. formel:

$\text{elBilAfgift} = \text{bilAfgift} \times 0.20$ ; hvor bilafgift er beregnet ud fra en af ovenstående formler.

Her er vist to beregninger:

Personbil til pris 300000:  $\text{bilAfgift} = 300000 \times 1.50 - 130000 = 320000$

Elbil til pris 300000:  $\text{elBilAfgift} = \text{bilAfgift} \times 0.20 = 64000$ , hvor bilafgift er beregnet som ovenfor.

### **Opgave 1: Class Library**

Opret et nyt projekt, Skat, af typen "Class Library(.Net Framework)" med en klasse Afgift.

- a) Opret flg. metoder i Afgift:

Du må gerne bruge hjælpemetoder til beregningerne.

- `int BilAfgift(int pris)`  
Returnerer personbilafgift ud fra parameteren pris.
- `int ElBilAfgift(int pris)`  
Returnerer elbilafgift ud fra parameteren pris.

- b) Giv beskrivende kommentarer til dine metoder.

- c) Sammen med din DLL skal du have XML kommentarer genereret.

### **Opgave 2: Test dit Class Library**

- a. Tilføj et testprojekt, som testerne metoderne i dit Class Library.  
Du skal teste begge metoder.
- b. Kør testen og overvej om din test har god "Code Coverage".  
Angiv %tallet og redegør kort for dine overvejelser.

### **Opgave 3: TCP Server**

Du skal nu gøre brug af en TCP Server, som kan tilbyde de samme to udregninger af registreringsafgiften. Dertil skal der designes en brugerdefineret protokol, der kan skelne mellem de to udregninger. Protokollen skal derfor kunne:

1. Modtage hvilken udregning der ønskes: ” Personbil” eller ”Elbil”  
*Derefter*
2. Modtage prisen

*Det kan anbefales først at lave en meget simpel TCP Server (denne opgave – Opgave3), som kan testes med en meget simpel TCP Client (Opgave 4). Herefter kan programmet (programmerne) udvides sideløbende. Din TCP Server må meget gerne fra start af udnytte/udstille dine metoder i Library*

- a. På denne baggrund overvej og design en meget simpel protokol, dvs. hvordan du adskiller om udregningen er ”Personbil” eller ”Elbil”, samt hvordan værdierne overføres.
- b. Lav et simpel Console Application projekt, der gør brug af en TcpListener og TcpClient sockets, og lytter (listening) på en port f.eks. 7000.
- c. Udvid programmet til:
  - Læs beskederne (defineret ved din protokol) fra klienten og udskriv beskeden.
  - Fortolk beskederne og udregn det ønskede resultat.
  - Send et svar med det udregnede resultat tilbage til klienten.
- d. Hvis du ikke har programmeret en multi-trådet (concurrent) TCP-server, skal du nu redegøre for hvordan dette kunne gøres. (5 linier burde være nok)  
Udvid derefter programmet, så TCP-serveren kan håndtere hver klient i en separat task/tråd.
- e. Din TCP-server skal nu kunne tilgå dit Library og derfor skal du skal nu udstille disse metoder i dette projekt. Du skal bruge din DDL og XML fil fra opgave 1.  
Udnyt derefter metoderne i dit Library til beregningerne af bilafgiften.

### **Opgave 4: TCP Client**

Lav et simpelt Console Application projekt, som kan forbinde til din TCP Server og som kan sende beskeder til Serveren.

- a. Send besked(er) til serveren, modtag og udskriv serverens svar.
- b. Brug Wireshark eller lign. værktøj til at opfange og vise de sendte beskeder og svar.  
Gem et par billeder i et dokument, så du let kan besvare næste opgave c).
- c. Forklar formatet og indholdet af et TCP-segment?  
Husk at skrive din forklaring i dit dokument.

### **Opgave 5: Typescript**

Programmer en webapplikation som giver mulighed for at beregne registreringsafgiften som beskrevet i starten af opgaven.

Du skal bruge webpack templaten når du løser opgaven. Husk at fjerne alt irrelevant typescript kode inden du starter programmeringen. Link til webpack templaten: <https://github.com/dimselab/Webpack-template>

- a) Webinterfacet (HTML'en) kan f.eks se ud som på nedenstående illustration. Det skal være muligt at angive en købspris uden afgift, samt om bilen er af typen bil eller elbil. Den beregnede afgift vises når der trykkes på knappen "Beregn Afgift". Det er ikke et krav at der bruges en dropdown liste til at vælge bil/elbil, andre løsninger godtages også.

Bil Pris:

  

bil ▼

Beregn Afgift

Bil afgift :234789

Du skal ikke bruge dit Library eller TCP server til løsning af denne opgave.

- b. Kan du udvide siden med en tæller, der viser, hvor mange gange brugeren har lavet en beregning. (Ekstraigave for de rigtig hurtige).