

OOP Eksamensopgave, Forår 2014

Opgaven er delt op i 2 dele. Læs hele opgaven igennem inden I begynder.

1. Struktur

I denne opgave skal der laves et auktions-system der administrerer salg af forskellige køretøjer (inspireret af lignende systemer som f.eks. autocom.dk).

Systemet skal administrere følgende typer køretøjer: **Personbiler**, **lastbiler**, **busser** og **autocampere**. I det følgende beskrives køretøjernes fællestræk samt deres forskelle.

Køretøj

Følgende egenskaber er fælles for køretøjer

- **Navn** (f.eks., "WV Polo" eller "Skoda Octavia")
 - Navn må ikke være *null*. Forsøg på at tildele en *null* værdi skal udløse en passende exception.
- **Km** (angiver hvor mange kilometer køretøjet har kørt i dets levetid)
 - Km må ikke være et negativt tal. Forsøg på at tildele en negativ værdi skal udløse en passende exception.
- **Registreringsnummer** (nummerplade)
 - Et registreringsnummer skal bestå af to bogstaver (i denne opgave accepteres alle bogstaver, ikke kun bogstaverne A-Z) efterfulgt af fem cifre. Forsøg på at tildele et ulovligt registreringsnummer skal udløse en brugerdefineret exception.
 - Når et registreringsnummer aflæses, skal de to første og de to sidste tegn skjules. Registreringsnummer XY12345 skal derfor vises som ****123****.
- **Årgang** (angiver hvilket år køretøjet er indregistreret)
 - Årgang skal være *read-only*, dvs. efter instantiering må værdien ikke kunne ændres.
- **NyPris** (angiver køretøjets oprindelige købspris)
 - NyPris må ikke være et negativt tal. Forsøg på at tildele en negativ værdi skal håndteres ved at NyPris tildeles værdien 0.
- **Trækkrog** (ja/nej)
 - I denne opgave skal personbiler til erhverv være udstyret med en trækkrog. For alle andre køretøjer er det frivilligt at have trækkrog.
- **KørekortType** (angiver hvilket kørekort der kræves for at føre køretøjet)
 - De mulige kategorier er A, B, C, D, BE, CE, DE. Se subclasseserne for detaljer.
- **Motorstørrelse** (som angivet ved slagvolumen i liter)

- I denne opgave vil forskellige typer køretøjer have forskellige tilladte motorstørrelser:
 - Den tilladte motorstørrelse for personbiler er 0.7 - 10 l.
 - Den tilladte motorstørrelse for lastbiler og busser er 4.2 - 15 l.
 - Den tilladte motorstørrelse for autocampere er 2.4 - 6.2 l.
- Forsøg på at tildele en værdi uden for den tilladte rækkevidde skal udløse en passende exception.
- **Km/l** (Angiver hvor mange kilometer køretøjet kan køre på en liter brændstof)
- **Brændstof** (Angiver den brændstoftype som køretøjet anvender)
 - Mulige værdier er *Diesel* og *Benzin*.
 - Lastbiler og busser kan kun køre på diesel. Alle andre køretøjer kan fås med begge brændstoftyper.
- **Energiklasse** (angiver hvor brændstoføkonomisk køretøjet er)
 - Energiklassen er en udregnet egenskab der afhænger af køretøjets brændstof, km/l og årgang. Energiklassen udregnes dog på en særlig måde for autocampere. Se længere nede.
 - For køretøjer før 2010 gælder følgende energimærker:
 - Hvis motorens brændstof-type er diesel, udregnes energiklassen som:
 - A klasse: $\text{km/l} \geq 23$
 - B klasse: $18 \leq \text{km/l} < 23$
 - C klasse: $13 \leq \text{km/l} < 18$
 - D klasse: $\text{km/l} < 13$
 - Hvis brændstof-typen er benzin, udregnes energiklassen som:
 - A klasse: $\text{km/l} \geq 18$
 - B klasse: $14 \leq \text{km/l} < 18$
 - C klasse: $10 \leq \text{km/l} < 14$
 - D klasse: $\text{km/l} < 10$
 - For køretøjer efter 2010 gælder følgende energimærker:
 - Hvis motorens brændstof-type er diesel, udregnes energiklassen som:
 - A klasse: $\text{km/l} \geq 25$
 - B klasse: $20 \leq \text{km/l} < 25$
 - C klasse: $15 \leq \text{km/l} < 20$
 - D klasse: $\text{km/l} < 15$
 - Hvis brændstof-typen er benzin, er energiklassen:
 - A klasse: $\text{km/l} \geq 20$
 - B klasse: $16 \leq \text{km/l} < 20$
 - C klasse: $12 \leq \text{km/l} < 16$
 - D klasse: $\text{km/l} < 12$
- **ToString()**
 - Alle køretøjer skal redefinere *object* klassens ToString() metode for at give en sigende beskrivelse af køretøjet.

Personbil

- **Antal sæder**
 - Der findes to slags personbiler: Personbiler til privat brug og personbiler til erhverv. Personbiler til erhverv er kendetegnet ved at de har akkurat to sæder, mens personbiler til privat brug kan have mellem 2 og 7 sæder.
- **Bagagerums dimensioner** (dette skal være en type med længde, bredde og højde)
- **Kørekorttype**
 - Kørekorttypen for personbiler er som udgangspunkt B. Personbiler til erhverv kan dog også kræve et BE kørekort.

Personbil til privat brug

- **Isofix beslag** (ja/nej)
 - Isofix beslag giver en forbedret forankring af autostole ift. selemontering.

Personbil til erhverv

- **Sikkerhedsbøjle** (ja/nej)
 - Angiver hvorvidt der findes en sikkerhedsbøjle bag førersædet.
- **Lasteevne**
 - Angiver hvor meget påhængsvægt bilen kan køre med. Hvis vægten overstiger 750 kg. kræver det et BE kørekort

Lastbil

- **Lasteevne** (Angiver maksimal lasteevne i kg)
- **Højde** (Angiver køretøjets højde)
- **Vægt** (Angiver køretøjets vægt)
- **Længde** (Angiver køretøjets længde)
- **Kørekorttype**
 - Er som udgangspunkt C. Hvis lastbilen har trækkrog kræver det imidlertid et CE kørekort.

Bus

- **Antal siddepladser**
- **Antal sovepladser**
- **Toilet** (Indikerer om bussen indeholder toilet-faciliteter)
- **Højde** (Angiver køretøjets højde)
- **Vægt** (Angiver køretøjets vægt)

- **Længde** (Angiver køretøjets længde)
- **Kørekorttype**
 - Er som udgangspunkt D. Hvis bussen har trækkrog kræver det imidlertid et DE kørekort.

Autocamper

- **Antal siddepladser**
- **Antal sovepladser**
- **Toilet** (indikerer om autocamperen indeholder toilet-faciliteter)
- **Varmesystem** (Angiver autocamperens primære varmekilde)
 - Mulige værdier er: *gas*, *strøm* eller *oliefyr*.
- **Energiklasse**
 - Autocamperens varmesystem spiller ind på hvilken energiklasse den bliver placeret i. Hvis autocamperen bruger oliefor, så skal km/l ganges med 0.7 inden energiklassen beregnes.
 - Hvis autocamperen bruger strøm, så skal km/l ganges med 0.8 inden energiklassen beregnes.
 - Hvis autocamperen bruger gas, så skal km/l ganges med 0.9 inden energiklassen beregnes.
 - Eksempel: En benzin-drevet autocamper fra 2007 der kører 18.5 km/l (klasse A) vil med oliefor havne i klasse C ($18.5 * 0.7 = 12.95$), mens en strøm-udgave vil havne i klasse B ($18.5 * 0.8 = 14.8$)
- **Kørekorttype**
 - Er altid B.

2. Administration af køretøjer (Auktionshus med købere og sælgere)

I skal lave et auktionshus hvor køretøjer kan sælges igennem. Auktionshuset formidler kontakten mellem en sælger og potentielle købere, og auktionshuset modtager et salær for hvert salg.

Sælger

En sælger er karakteriseret ved følgende:

- **Saldo**
 - Når en sælgers køretøj bliver solgt, så tilføjes køretøjets salgspris til sælgerens saldo.
- **PostNummer** (udtrykt som et heltal)
- **ModtagNotifikationOmBud(...)**

- Dette er en metode der kan bruges til at notificere sælgeren om at der er afgivet et interessant bud på et køretøj, som sælgeren har til salg. Metoden kan nøjes med at registrere (f.eks. udskrive) information om det pågældende køretøj samt buddets størrelse.

Køber

- **Saldo**
 - Når et køretøj købes, trækkes køretøjets købspris fra saldoens eksisterende beløb.

Sælgere og købere fordeler sig på to typer: **privatpersoner** og **firmaer**. Begge typer kan optræde i begge roller; dvs. en privatperson/ et firma kan være både sælger og køber.

En privatperson er kendetegnet ved at have et unikt *CPR nummer*, mens et firma er identificeret ved at have et unikt *CVR nummer*. Et firma har desuden *kredit*, der tillader firmaets saldo at gå i minus. Som eksempel, så vil et firma med en saldo på 40.000 kr., og en kredit på 30.000 kr. være i stand til at købe køretøjer der koster op til 70.000 kr. En privatpersons saldo må derimod ikke gå i minus, og en tilsvarende person kan derfor højst købe køretøjer til en værdi af 40.000 kr.

Auktionshus

I skal lave et auktionshus hvor køretøjer kan sælges igennem. Auktionshuset formidler kontakten mellem en sælger og potentielle købere, og tager følgende salærer for salg af køretøjer:

Salgspris	Salær
0 < 10.000	DKK 1.900
10.000 < 50.000	DKK 2.250
50.000 < 100.000	DKK 2.550
100.000 < 150.000	DKK 2.850
150.000 < 200.000	DKK 3.400
200.000 < 300.000	DKK 3.700
>= 300.000	DKK 4.400

Tabel 1: Auktionshusets salærer (kopieret fra <http://www.autocom.dk/consumer/conditions2009.do>)

Auktionshus-klassen skal tilbyde følgende metoder til at administrere salg af køretøjer:

`public int SætTilSalg(Køretøj k, Sælger s, decimal minPris)`

- Et køretøj sættes til salg via ovenstående metode, hvor *k* er køretøjet der sættes til salg, *s* er sælgeren, og *minPris* angiver mindstebeløbet før sælger vil overveje at sælge. Når der kommer et bud der overstiger den angivne minimumpris, skal sælgeren notificeres herom via sælgers `ModtagNotifikationOmBud(..)` metode. Endeligt skal der returneres et unikt auktionsnummer der kan bruges til senere at referere til det givne køretøj der er sat til salg.

`public int SætTilSalg(Køretøj k, Sælger s, decimal minPris, notifikationsMetode)`

- Som ovenfor, men her skal notifikationer i stedet sendes til den angivne *notifikationsMetode*.

`public bool ModtagBud(Køber køber, int auktionsNummer, decimal bud)`

- Potentielle købere skal kunne afgive bud på et bestemt køretøj som identificeret vha. det angivne auktionsnummer. Metoden returnerer en `bool` der angiver om budet accepteres. Et bud accepteres kun hvis køberen har penge nok til at kunne købe køretøjet, og hvis budet er større end det hidtil største bud. I tilfælde af, at budet accepteres, og samtidig overstiger sælgers angivne minimumpris, så skal sælger notificeres om at der er afgivet et interessant bud.

`public bool AccepterBud(Sælger sælger, int auktionsNummer)`

- Ovenstående metode skal implementeres for at afslutte en handel med et bestemt auktionsnummer. Der skal først laves et tjek på at den angivne sælger er identisk med den sælger der har sat køretøjet til salg. Såfremt sælgerens identitet er i orden, så afsluttes salget ved at købssummen trækkes fra købers saldo (den køber der har afgivet det største accepterede bud). Herefter trækker auktionshuset sit salær i overensstemmelse med Tabel 1, og endelig indsættes det resterende beløb på sælgers saldo. På dette tidspunkt skal køretøjet ikke længere figurere som værende til salg i auktionshuset, men skal i stedet overgå til en liste af solgte køretøjer. Listen af solgte køretøjer skal kunne gennemløbes udenfor AuktionsHus klassen. Metodens returværdi angiver om handlen blev succesfuldt gennemført.

Udover ovenstående funktionalitet til at administrere salg af køretøjer, så skal auktionshuset også understøtte følgende søgefunktioner blandt køretøjerne der er til salg:

- 1) Find køretøjer hvis navn indeholder en angivet søgestreng.
- 2) Find køretøjer der har et minimum angivet antal siddepladser samt toiletfaciliteter.
- 3) Find køretøjer der kræver stort kørekort (kategori C, D, CE eller DE) og vejer under en angivet maksimalvægt.
- 4) Find alle personbiler til privatbrug som har kørt under et angivet antal km, og hvor minimum salgsprisen samtidig ligger under et angivet beløb. Køretøjerne skal returneres i sorteret rækkefølge efter antal kørte km.
- 5) Find alle køretøjer hvor køretøjets sælger er bosiddende inden for en bestemt radius af et angivet postnummer. I denne forbindelse kan radius blot anskues som et tal der skal lægges til/trækkes fra postnummeret. F.eks. vil en søgning efter køretøjer indenfor en radius af 1500 fra postnummer 8000, inkludere alle køretøjer hvor sælgers postnummer ligger mellem 6500 og 9500.

Til slut, lav en metode der returnerer den gennemsnitlige energi-klasse for alle køretøjer til salg. I den forbindelse kan energi-klasserne konverteres til tal (A = 1, B = 2, C = 3, D = 4). Afrund resultatet til nærmeste heltal inden den tilhørende energiklasse returneres.

Overordnet

Lav et passende klassesdesign der opfylder ovenstående krav, inkl. et passende klassehierarki der reducerer koderedundans. Brug ligeledes passende synlighedsmodifikatorer og datatyper. Navngivning af klasser, egenskaber og metoder står Jer frit for. Ligeledes står det Jer frit for om – og i givet fald hvordan – I vil foretage data validering udover de beskrevne krav. Endelig står det Jer frit for at tilføje yderligere funktionalitet (klasser, metoder og properties) udover de eksplisit angivne krav.

Opgaveløsningen skal kunne demonstreres gennem en klasse der implementerer en simpel tekstuel brugergrænseflade (en konsol-applikation). En "hardcodet" demonstration af funktionaliteten vil være tilstrækkelig. I forventes således ikke at understøtte bruger input.

Opgaven rummer en del variationsmuligheder mht. opfyldelse af de funktionelle krav, hvorfor der ikke kun findes én rigtig måde at løse den på. Der vil dog være fokus på at I laver et hensigtsmæssigt klassesdesign, samt at I sørger for at jeres kode er læsbar.

Eventuelle uklarheder i opgaveformuleringen forventes afklaret som en del af opgaveløsningen.

Krav til program og dokumentation

- Det udviklede program skal være objekt-orienteret, og det skal skrives i C#.
- Kildeteksten af det udviklede program skal afleveres i to eksemplarer på elektronisk form (CD-ROM, DVD eller USB memory stick). **Blu-Ray medier modtages ikke!**
- Det udviklede program skal være **demonstrerbart**, dvs. det skal kunne afvikles uden bugs.
- Hver offentlig/internal type skal placeres i en .cs fil med samme navn som typen (eks.: Køretøj klasse => Køretøj.cs, LastBil klasse => LastBil.cs, osv).
- Med det udviklede program skal der følge en kort beskrivelse af programmet (ca. én side). Denne beskrivelse skal omfatte:
 - Mini-gruppens navn samt deltagernes navne. Se mini-gruppe listen i mappen "OOP eksamensopgave 2014" på OOP kursets Moodle hjemmeside. Omgængere der laver opgaven individuelt, skal ikke angive mini-gruppe navn.
 - Status af programmet, inklusiv en beskrivelse af eventuelle antagelser, begrænsninger, udvidelser, eller variationer i forhold til opgaveformuleringen.
 - Angivelse af eventuelle kilder (dog ikke kursusbogen eller kursusslides).
 - Underskrift fra alle deltagere. ***En underskrift bekræfter deltagelse i programmeringsarbejdet, og at ingen andre end de underskrevne har bidraget til arbejdet.***