

Godkendelsesopgave E

Grundlæggende Programmering
(Deadline: 27. november 2016)

1) CAR:

- a) Lav en klasse `Car`, der repræsenterer en *bil*. En bil har en ejer, `owner` (tekststreng), en nummerplade, `plate` (tekststreng), samt en pris: `price` (tal). Alle felter skal være `private` og alle metoder skal være `public`.
- b) Klassen `Car` skal have en konstruktør, der opretter en bil med en ejer samt nummerplade; prisen bør altid være nul (0) til at starte med¹.
- c) Der skal være en metode `public int getPrice()`, der returnerer prisen på bilen.
- d) Dernæst skal der være en metode til at "sætte" en pris, `setPrice(int price)`, der som argument tager et tal (`price`), som er en pris. Hvis prisen er negativ, skal der udskrives en passende fejlbesked; ellers skal feltet `price` sættes til værdien af argumentet `price`.
- e) Til slut skal klassen have en metode `public String toString()`, der skal returnere en `String` i formatet:

Ragnar's car #ABC-123: 10000 DKK

Her er en oversigt over klassens ingredienser:

Klasse:	<u>Car</u>
• Felter:	<code>owner, plate, price</code>
• Konstruktører:	<code>Car(owner, plate)</code>
• Metoder:	<code>getPrice(), setPrice(price), toString()</code>

2) DEALER:

- a) Lav nu en klasse `Dealer`, som repræsenterer en *brugtvogetsforhandler* der har et navn, `name` (som er en tekststreng) samt en array-liste af biler: `cars`, som er en `ArrayListe` af biler.
- b) `Dealer` skal have en konstruktør, der tager en tekststreng (`name`); `cars` skal initialiseres til den tomme `ArrayListe` af biler.
- c) `Dealer` skal have en metode `void addCar(Car car)`, der tilføjer en bil til array-listen.
- d) Derudover skal `Dealer` have en metode, `show()`, til at udskrive alle biler ved at kalde deres `toString()`-metoder og udskrive resultaterne via `System.out.println(...)`.
- e) Endeligt skal `Dealer` have en metode `showPriced()`, der kun udskriver de biler der har fået en pris (dvs, de biler der har en pris, som ikke er nul).

Klasse:	<u>Dealer</u>
• Felter:	<code>name, cars</code>
• Konstruktører:	<code>Dealer(name)</code>
• Metoder:	<code>addCar(), show(), showPriced()</code>

¹ Bemærk: det er i denne opgave ok at man jo således ikke kan se forskel på om en bil har en pris på nul eller om bilen endnu ikke har fået en pris.

3) TOTAL & TO-ARRAY:

- a) Lav nu en metode `public int total()` i `Dealer`, der beregner og returnerer den samlede værdi af alle bilerne i Array-listen `cars`.
- b) Lav til sidst en metode med signatur "`public Car[] toArray()`" i `Dealer`, der returnerer listen som et array af biler istedet for som en array-liste af biler. Husk at størrelsen af et array skal defineres allerede når det oprettes. Som et hint kan det oplyses at følgende kode opretter et array af 42 referencer til objekter af typen `String`:

```
String[] strings = new String[42];
```

Klasse: `Dealer`

- **Felter:** `/* as before */`
- **Konstruktører:** `/* as before */`
- **Metoder:** `/* as before */`, `total()`, `toArray()`

4) DEMO:

Test dit fulde program med følgende kode:

--- Demo.java ---	--- OUTPUT ---
<pre>public class Demo { public static void main(String[] args) { Dealer d = new Dealer("Billige Biler"); Car c1 = new Car("Ragnar", "ABC-123"); Car c2 = new Car("Rollo", "UVW-456"); Car c3 = new Car("Floki", "XYZ-789"); d.addCar(c1); d.addCar(c2); d.addCar(c3); d.show(); c1.setPrice(10000); c3.setPrice(30000); d.showPriced(); System.out.println(d.total()); Car[] cars = d.toArray(); for (int i=0; i<cars.length; i++) { System.out.println(cars[i]); } } }</pre>	<pre>*** Billige Biler *** Ragnar's car #ABC-123: 0 kr Rollo's car #UVW-456: 0 kr Floki's car #XYZ-789: 0 kr *** Billige Biler *** Ragnar's car #ABC-123: 10000 kr Floki's car #XYZ-789: 30000 kr 40000 Ragnar's car #ABC-123: 10000 kr Rollo's car #UVW-456: 0 kr Floki's car #XYZ-789: 30000 kr</pre>