Bensinstasjon – fredags python

I denne oppgaven skal du lage noen utvalgte objekter som skal kunne simulere en gitt bensinstasjon. *Ikke å forvente at man klarer alle deloppgavene med tanke på at dere ikke en gang har hatt en uke med klasser og objekter. Men gjør så mange du klarer og spør om hjelp om du står fast :D*

1)

Lag klassen Bil. Bil skal ha en konstruktør som tar inn: hvilket drivstoff bilen bruker og registreringsnummer.

Metoder i bil-klassen bør være:

 - hentRegNr() som skal returnere registreringsnummeret.

 - hentDrivstoff() som skal returnere hvilket drivstoff som bilen bruker.

Test:

- Lag en ny fil som heter testBil.py, og lag et par bil-objekter og test metodene.

2)

Lag klassen Pumpe. Vi tenker oss at en pumpe enten har bensin eller diesel, man skal også kunne holde styr på om pumpen er ledig. Pumpe skal ha en konstruktør som tar inn: Om det er en bensin eller diesel pumpe, pumpe-id (skal være et tall, og bør ikke være likt noen andre pumpe-id’er). Konstruktøren skal også sette at self.\_ledig = True.

Metoder:

 - erLedig() som skal returnere om pumpen er ledig eller ikke.

 - settLedig() som skal sette ledig til enten True eller False, True om den er
 ledig og False om den ikke er ledig.

 - hentId() som skal returnere id’en til pumpa.

 - hentDrivstoff() som skal returnere hvilket drivstoff pumpen bruker

Test:

- Lag en ny fil som heter testPumpe.py, og lag et par pumpe-objekter og test metodene.

3)

Lag klassen Bensinstasjon. Bensinstasjon skal ha en liste med pumper, denne skal bli tatt inn i konstruktøren.

Metoder:

* finnPumpe( ) som tar inn et argument *drivstoff* som sier noe om hvilken type drivstoff pumpen vi skal finne har. Så skal den gå igjennom pumpene på bensinstasjonen og returnere en pumpe som er ledig og som har riktig drivstoff.

Lag også en metode i klassen Bil som skal hete fyllDrivstoff(pumpe). For å illustrere at pumpa er i bruk sett pumpa sin self.\_ledig til å være False.

Lag så ende en metode i klassen bil som heter draFraPumpe(pumpe). Denne skal si til pumpen at bilen har kjørt og at pumpen nå er ledig.

4)

Lag et hovedprogram. Her skal du importere de tre klassene. Lag to pumper, en som har bensin og en som har diesel. Lag deretter et bensinstasjon-objekt med som tar inn en liste med de to pumpene du nettopp lagde. Lag så 4 biler og finn på et sannsynlig hendelsesforløp. (bruk metodene til bensinstasjonene og bilene). Gjør endring i klassene dersom det trengs.

5 utfordring)

Legg til en pris på bensin og diesel og integrer prisen i fyllDrivstoff(). Her skal fyllDrivstoff() ta inn et argument til som sier noe om hvor mange liter drivstoff som bilen skal fylle på, så skal den returnere en pris for drivstoffet som er fylt på.

Her kan du også forandre klassene slik du vil for at systemet skal bli best mulig.