SENSORVEILEDNING

For eksamen i: **inf-1400**

Dato: **24. 05.2016**

Sensorveiledningen er på 4 sider inklusiv forside

Fagperson/intern sensor: Lars Ailo Bongo

Telefon: 92015508

Oppgave 1a (10%)

Vi introduserer en ny klasse, Romskip. Denne har attributtene og metodene som er felles for Romskip- klassene. Ved å la SkyveRomskip og SensorRomskip arve fra Romskip trenger vi ikke inkludere felles- atributtene i klassene siden attributtene blir arvet.

Oppgave 1b (15%)

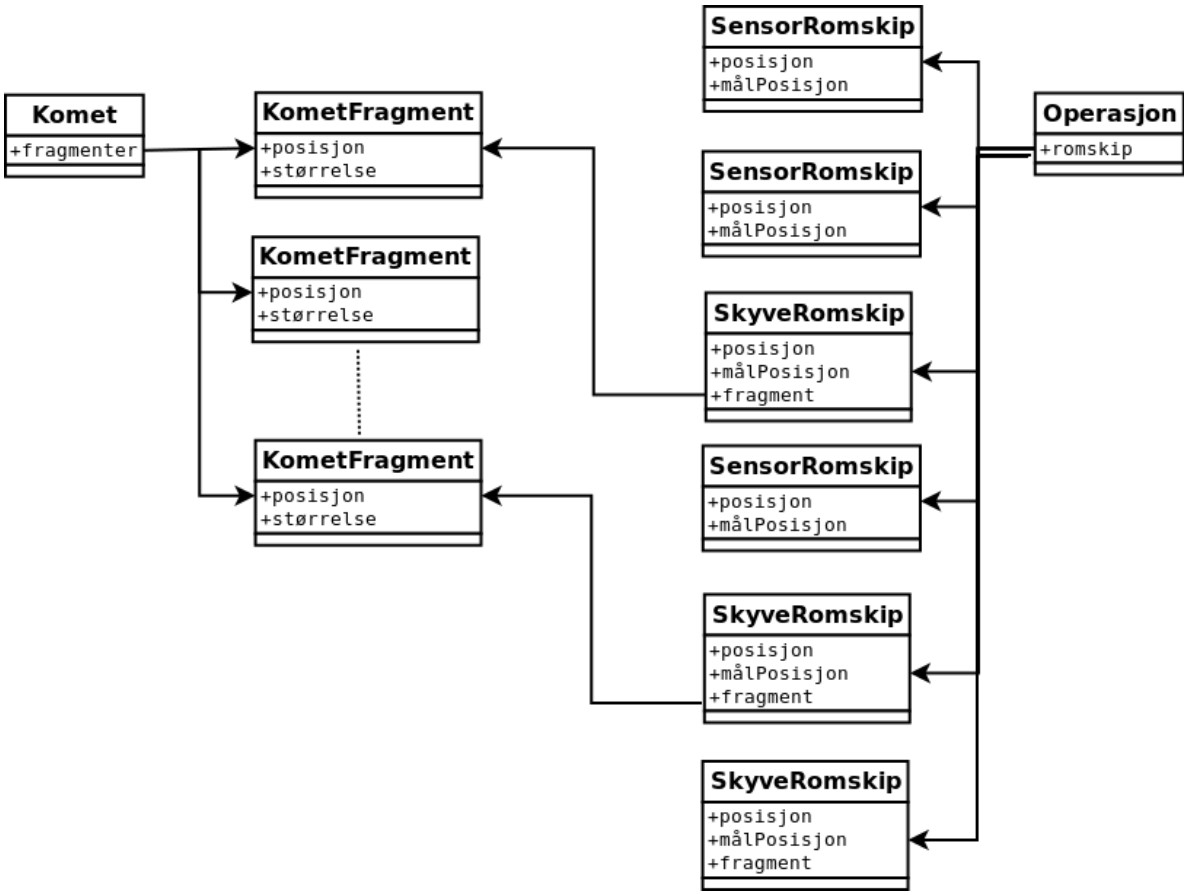
```
# Det er to måter å definere attributter på:
# # # # # # #

enten direkte i klassen
   class Foo:
       attrib = None /default val
eller i __init__ (men det krever at man husker på å kalle init på
parent)
   class Foo:
       def __init__(self):
           self.attrib = None
class Romskip:
    def __init__(self):
        self.posisjon = None
        self.malPosisjon = None
class SensorRomskip(Romskip):
    def __init__(self):
        Romskip.__init__(self)
class SkyveRomskip(Romskip):
    # Eksempel på alternativ attributtdefinisjon
    fragment = None
    def __init__(self):
        Romskip.__init__(self)
# Denne vil ikke virke siden alle instanser av klassen
# vil dele den samme listen.
class OperasjonFail:
    romskip = []
# Enklere å definere med __init__
class Operasjon:
    def __init__(self):
        self.romskip = []
class KometFragment:
    posisjon = None
    storrelse = None
class Komet:
```

```
    def __init__(self):
        fragmenter = []
```

Oppgave 1c (10%)

Tegne datastrukturen. Her har jeg misbrukt UML i Dia. Nøyaktig hvordan de tegner datastrukturen er ikke så nøye, det viktigste er at det kommer klart fram hvordan instansene henger sammen.



Oppgave 1d (15%)

Implementering av metoder I Operasjon:

```
    def plasserSensorRomskip(self, posisjoner):
        sensorskip = [x for x in self.romskip if isinstance(x,
SkyveRomskip)]
        for p in posisjoner:
            if len(sensorskip) < 1:
                # har ikke nok sensorRomskip
                break
            skip = sensorskip.pop()
            skip.settMalPosisjon(p)
```

I Romskip:

```
    def settMalPosisjon(self, pos):
        self.malPosisjon = pos
```

I SkyveRomskip:

```
    def settMalPosisjon(self, pos):
        if self.fragment is not None:
            self.fragment = None
        Romskip.settMalPosisjon(self, pos)
```

SensorRomskip arver fra Romskip og trenger ikke en egen. SkyveRomskip utvider funksjonaliteten til Romskip.settMalPosisjon.

Oppgave 2a (45%)

Dette er en stor oppgave, og en type oppgave som ikke kan gis I en papir eksamen. Studentne er derfor ikke vant til å løse denne typen oppgave, og sensur bør ta hensyn til dette. Det er ikke ett detaljert løsningsforslag til denne oppgaven, siden det er mange mulige løsninger. Karakteren bør settes basert på ett helhets inntrykk av hvor godt designet er.

Oppgave 2b (5%)

Her forventer jeg en beskrivelse av datastrukturer optimalsiert for søk, og hvordan de kan brukes I programmet.