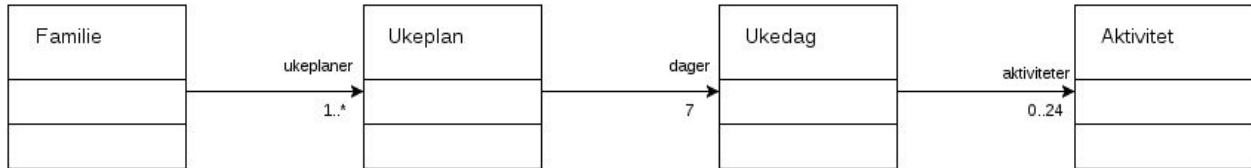


# Familiens ukeplaner (fra eksamen INF1000, 2014)

Du har påtatt deg å lage ukeplaner for hele familiens faste aktiviteter. For å gjøre det enkelt antar vi at alle aktiviteter starter på en helt time (kl 00.00, 01.00, 02.00, etc.) og at all varer nøyaktig 1 time.

Du skal bruke klassene vist i dette UML klassediagrammet:



- a) Klassen **Aktivitet** skal ha en representasjon med disse to instansvariablene: en tekststreng med navn **aktNavn** som brukes til å beskrive hva slags aktivitet det gjelder (for eksempel "svømming" eller "trener fotball 1A"), og et heltall med navn **start** som inneholder starttidspunktet for aktiviteten. Signaturen til konstruktøren skal se slik ut:

```
def __init__(self, hva, kl)
```

Skriv klassen **Aktivitet**.

- b) Skriv klassen **Ukedag** med dette foreløpige grensesnittet (dvs. disse metodene):

```
# Konstruktør
__init__(self, dag)
```

```
# Oppretter og setter inn ny aktivitet på angitt klokkeslett,
# og gir feilmelding på terminal om tidspunktet er opptatt.
settInn(self, hva, kl)
```

```
# Returnerer når på dagen den første aktiviteten starter
# (-1 hvis det ikke finnes noen aktivitet denne dagen)
tidligste(self)
```

```
# Returnerer når den seneste aktivitet den dagen starter
# (-1 hvis ingen aktiviteter)
seneste(self)
```

```
# Beregner og returnerer antall aktiviteter den dagen
antall(self)
```

Du skal programmere disse metodene.

- c) Klassen **Ukedag** skal nå utvides med metoden **settInnLedig**:

```
def settInnLedig(self, hva)
```

som setter inn en aktivitet for en ledig time den dagen. Hvis kalenderen er full, skal metoden skrive ut en feilmelding. Hvis det ikke er noen aktiviteter fra før den dagen, skal aktiviteten starte kl 12.00. Ellers skal metoden helst sette inn den nye aktiviteten i en ledig time mellom tidligste og seneste aktivitet; hvis det ikke er mulig, skal den settes rett etter den hittil seneste aktiviteten; om heller ikke det er mulig, settes aktiviteten inn rett før den hittil tidligste.

Skriv metoden **settInnLedig**.

- d) Skriv klassen **Ukeplan** med følgende grensesnitt (dvs. metoder):

```
# Konstruktør
__init__(self, hvem)

# Returnerer ukedagen med flest aktiviteter
travleste(self)
```

- e) Klassen **Ukeplan** skal nå utvides med metoden **skrivUt** som har denne signaturen:

```
def skrivUt(self)
```

Metoden skal sørge for at ukeplanen skrives ut, for eksempel slik:

```
Ukeplan for Aud:
Mandag
  Kl 9: Forelesning
  Kl 10: Forelesning
  Kl 12: Lunsj
Onsdag
  Kl 17: Trening
```

Skriv metoden **skrivUt**. Du vil sannsynligvis trenge nye metoder i flere av de andre klassene også; skriv disse metodene og angi i hvilken klasse den enkelte metoden hører hjemme. Selve formatteringen av utskriften er ikke så viktig, så ikke bruk så mye tid på det.

f) (Vanskelig)

Skriv klassen **Familie** med nødvendig datarepresentasjon og metoden (som du skal skrive):

```
def skrivAktiviteter(self)
```

Denne metoden skal skrive ut en liste over alle de ulike aktivitetene (fra **aktNavn**-variabelen) familien er med på. Aktiviteter med samme navn (dvs. har lik verdi i **aktNavn**-variabelen) skal kun skrives ut én gang selv om de forekommer flere ganger hos samme eller forskjellige familiemedlemmer.

Hint: bruk f.eks. en ordbok (*dictionary*) til å holde orden på hvilke aktivitetsnavn som har vært skrevet ut.