Menti

IN1000 gruppetime uke 7

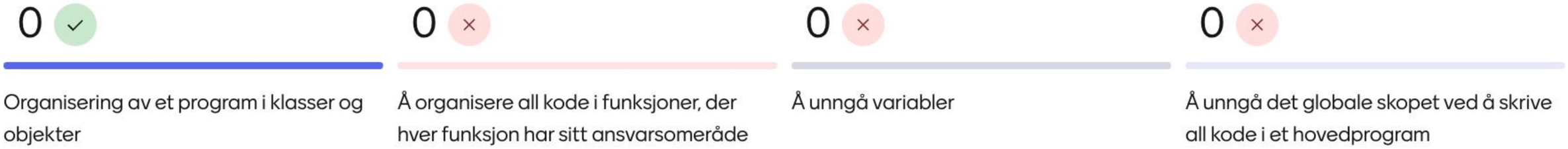
Gruppe 4: Design

Menti

Plan for i dag

- Gjennomgang av fagord
- Hva er meningen med klasser og objektorientering
- Hvordan er klasser satt opp
- Oppgaver

Hva beskriver best objektorientering?



Menti

Hva er en klasse?

0	0	0	0
Det samme som et objekt	En samling av variabler	En oppskrift for objekter	En tekststreng

Menti

Hva er et objekt?

0 ✓	0 ×	0 ×	0 ×
En konkret instans av en klasse	Det samme som en klasse	En oppskrift for å modellere virkeligheten	En pakke med moduler

Menti

Hva brukes konstruktøren __init__ til?

0 ✗	0 ✗	0 ✗	0 ✓
Å avslutte programmet	Å skrive ut dokumentasjon	Å importere en modul	Å initialisere et nytt objekt

Menti

Hva er en instansvariabel?

0	0	0	0
En variabel som aldri endrer verdi	En variabel som tilhører et bestemt objekt	En variabel som tilhører en bestemt klasse	En variabel som bare brukes i løkker

Menti

Hva er en metode?



Menti

Hva er et grensesnitt?

0	0	0	0
En oversikt over alle metoder og instansvariabler et objekt har	En kopi av et objekt	Måten annen kode aksesserer et objekt på	Et annet ord for terminal der man kjører programmer

Hva betyr innkapsling?

0 ×

Konstruksjon av et objekt der instansvariabler er utilgjengelige utenfor klassen

0 ×

Å dele all data til det globale skopet

0 ✓

Bestemmelse av metoder og instansvariabler som er tilgjengelige utenfor klassen

0 ×

Å skrive kode uten klasser

Menti

Hva er objektorientering og meningen med det?

- Objektorientering og klasser lar oss modellere og lage "verden"
- Biler for eksempel følger en ganske lik oppskrift, men har fortsatt noen forskjeller som skiller dem fra hverandre:
  - Farge
  - Størrelse
  - Maks-hastighet
  - osv.
- En klasse er altså en oppskrift (modell) som forteller oss hvordan noe skal oppføre seg, men de kan fortsatt ha ulike verdier

Menti

Konstruktør

- Her sender vi inn instansvariabler som er startverdiene når vi oppretter en instans av en klasse
- Man kan tenke at "__init__" blir kalt på når vi oppretter en instans av klassen
- self._ er bare en måte å si at dette er interne verdier som ikke skal bli direkte brukt

```python
class Eksempel:
    #Dette er konstruktøren, dette er det som skjer når vi
    #lager en instans
    def __init__(self, parameter1, parameter2):
        self._parameter1 = parameter1
        self._parameter2 = parameter2

    def hent_para1(self):
        return self._parameter1

eks = Eksempel(1, 2)

#ordentlig måte å hente private verdier
print(eks.hent_para1())

#ikke ordentlig måte å hente private verdier
print(eks._parameter1)
```

Menti

Metoder

- Fungerer helt likt som prosedyrer og funksjoner, men de tilhører selve klassen og bruker verdiene til instansen av klassen
- Husk å alltids ha self som parameter

```
class Bil:
    #Dette er konstruktøren, dette er det som skjer når vi
    #lager en instans
    def __init__(self, farge):
        self._farge = farge

    #Dette er en metode
    def endre_farge(self, ny_farge):
        self._farge = ny_farge

#Oppretter objekt
min_bil = Bil("rød")

#Kaller på metoden til objektet
min_bil.endre_farge("blå")
```

Menti

Dere har faktisk brukt klasser masse før

- str(), int(), list() osv. er måter å instansiere en klasse på
- Det er ikke funksjoner, men dere oppretter objekter av klasser
- Så måten man oppretter klasser på er akkurat likt som str(), int(), list():
  - liste= list([1, 2, 3]), oppretter et liste-objekt som har verdiene 1, 2, 3
  - bil = Car("rød"), oppretter et eget bil-objekt som har fargen rød
- Alle lister har da lik oppførsel (metoder som append()), men kan ha ulike verdier

Klasser dere lager er også mutable

- I likhet med lister så kommer dere til å lage mutable klasser
- Måten jeg lager klassen min på og bruker tjenesten er i likhet med en liste

```python
class Bil:
    def __init__(self, farge):
        self._farge = farge

    def endre_farge(self, ny_farge):
        self._farge = ny_farge

    def hent_farge(self):
        return self._farge

liste1 = list([1, 2, 3])
liste2 = liste1
liste1.append(4)

min_bil1 = Bil("blå")
min_bil2 = min_bil1
min_bil1.endre_farge("rød")

print(liste2) #printer [1, 2, 3, 4]
print(min_bil2.hent_farge()) #printer "rød"
```

Menti

Oppgaver!

Oppgaver kan dere finne på emnesiden ved gruppe 4

Deretter noe livekoding av oppgavene