

Ekstraprøve IT-2 høst 25

Emner : 2.6 Objektorientert programmering

: 2.9 Reelle datasett med JSON

Tid : 07.01.2025 kl. 09:45-11:20 (90 min)

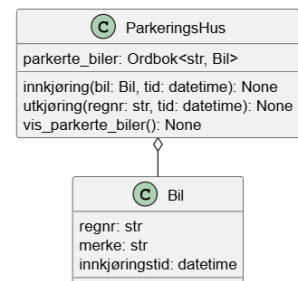
Hjelpemidler : Alle ikke-kommuniserende hjelpemidler er tillatt

- Instruksjoner**
- 1) Opprett en mappe for besvarelsen.
 - 2) Når du er ferdig, pakk sammen alle filene i en zip-fil, som du navngir med ditt navn.
 - 3) Send filen som besvarelse til lærer på Teams

Oppgave 1: OOP (50%)

a) Du skal lage et objektorientert program for et parkeringshus. Kravene til systemet er

- Når en bil kjører inn i parkeringshuset skal bilen og tidspunktet registreres og skrives ut.
- Når en bil kjører ut av parkeringshuset skal parkeringstiden og avgiften beregnes og skrives ut. Prisen er 0,50 kr pr minutt.
- Følg klassediagrammet her ved siden



- b) Test ut funksjonaliteten i systemet ved å opprette objekter og metoder fra hovedblokken `if __name__ == "__main__":`. Se eksempel under.
- c) Hvilke feil kan oppstå i et parkeringshussystem som oppfyller kravene beskrevet ovenfor, dersom systemet baserer seg på manuell registrering?

TIPS

Bruk en ordbok med registreringsnummer som nøkkel og bilobjektet som verdi for parkerte biler. Da kan vi hente og fjerne en bil fra ordboken med `.pop(regnr)`.

Parkeringstid kan registreres på følgende måte

```
# Eksempel på beregning av tidsdifferanse med datetime
from datetime import datetime

# datetime(år, måned, dag, time, minutt)
tid_inn = datetime(2026, 1, 7, 9, 45)
tid_ut = datetime(2026, 1, 7, 11, 20)

# bruk datetime.total_seconds() til å hente ut antall sekunder
tidsdiff = tid_ut - tid_inn
print(f"Mellom {tid_inn} og {tid_ut} er det {tidsdiff.total_seconds()/60} minutter")

Mellom 2026-01-07 09:45:00 og 2026-01-07 11:20:00 er det 95.0 minutter
```

Forts. neste side.

Eksempel på kjøring:

<pre>if __name__ == "__main__": phus = ParkeringsHus() bil = Bil("AB12345", "Tesla") tid = datetime(2026, 1, 7, 7, 45) phus.innkjøring(bil, tid) bil = Bil("EK13002", "eGolf") tid = datetime(2026, 1, 7, 7, 55) phus.innkjøring(bil, tid) tid = datetime(2026, 1, 7, 15, 50) phus.utkjøring("AB12345", tid) bil = Bil("TV12345", "Ford") tid = datetime(2026, 1, 7, 16, 00) phus.innkjøring(bil, tid) phus.vis_parkerte_biler()</pre>	<p>Opprettet Tesla med reg.nr.: AB12345 AB12345 kjørte inn 2026-01-07 07:45:00</p> <p>Opprettet eGolf med reg.nr.: EK13002 EK13002 kjørte inn 2026-01-07 07:55:00</p> <p>AB12345 kjørte ut 2026-01-07 15:50:00 Parkerings tid: 485 min., avgift: 242.50 kr</p> <p>Opprettet Ford med reg.nr.: TV12345 TV12345 kjørte inn 2026-01-07 16:00:00</p> <p>Parkerte biler: EK13002 eGolf TV12345 Ford</p>
---	--

Oppgave 2: JSON (50%)

I denne oppgaven skal du håndtere data fra 100 tilfeldige passasjerer på Titanic. Dataene er lagret i filen: titanic_100.json og det som er registrert på hver passasjer er:

- klasse: type billett hen reiste med
 - alder: alderen til den reisende
 - avgift: avgiften som ble betalt for billetten i dollar
 - påstigningshavn: hvilken havn hen steg på
 - overlevde: hvis hen overlevde
- a) Lag et program som leser inn filen og viser andelen (prosent) av passasjerene som reiste med første, andre og tredje klasse.
- b) Utvid programmet med å summere billettinntektene til alle passasjerer i henholdsvis: første, andre og tredje klasse.
- c) Vis klassen, alder, og om passasjeren overlevde for alle passasjerer som steg på i Quenstown. Listen skal sorteres etter stigende rekkefølge etter alder.
- Se eksempel under på utskrift:

```
...
Klasse: 3
Alder: 28
Passasjerer overlevde ikke :(

Klasse: 3
Alder: 42
Passasjerer overlevde :)
```