

EKSAMENSFORSIDE

Skoleeksamen med tilsyn

Generell informasjon om eksamen:

Emnekode: PRG1100-1

Emnenavn: Grunnleggende programmering 2

Dato: 24.05.2024

Start klokkeslett: 09:00

Antall timer: 4

Emneansvarlig: Ståle Vikhagen

Campus: Ringerike

Fakultetet: HH

Antall oppgaver: 5

Antall vedlegg: 0

Totalt antall sider (inkludert forside og vedlegg): 7

Tillatte hjelpemidler (jfr. emneplan):

Alt trykt og skriftlig materiale.

Opplysninger om vedlegg:

Ingen vedlegg.

KANDIDATEN MÅ SELV KONTROLLERE AT OPPGAVESETTET ER FULLSTENDIG

Merknader:

Programmeringsoppgavene skal løses med kode i vedleggsverktøyet (vedleggsverktøyet åpnes i en ny fane).

NB! Koden må lagres med det navnet som er oppgitt i oppgaven. I besvarelsen henviser du til vedlegget med din kode. Generell struktur for lagring av oppgavene er <kandidatnr>-oppgave#.py, der <kandidatnr> byttes ut med ditt kandidatnr og # med oppgaven sitt nr. Eksempel, kandidat 4321 sitt svar på oppgave 2 skal se slik ut i besvarelsen med henvisning til vedlegget: Oppgave 2, svar på oppgave 2 finner du i 4321-oppgave2.py. Det er viktig at du løser programmeringsoppgavene på denne måten, for at sensor skal kunne vurdere besvarelsen best mulig.

Oppgavesettet består av **5 oppgaver og alle oppgavene skal besvares.**

Les gjennom hele oppgavesettet før du begynner å løse oppgavene. Les hver oppgave nøye slik at du svarer på det du blir spurt om. Vektingen av oppgaven viser hvor mye den teller ved sensuren.

Legg spesielt merke til informasjonen om filstruktur og databasestruktur beskrevet på side 3-5.

Eksamensoppgavetekst:

Caset: BilDeling

Firmaet BilDeling henvender seg til et segment i bilmarkedet hvor personer ikke er opptatt av å eie egen bil. De kan her lett få tilgang til leiebil ved behov, hvor kunden kan åpne og starte bilen ved hjelp av mobiltelefonen etter at kunden har registrert seg med et abonnement på BilDeling hvor en oppgir mobiltelefonnr, navn og et betalingskort. Kunden kan åpne og starte bilen ved hjelp av mobilen, i og med at app'n kommuniserer med ekstramontert utstyr i bilene. BilDeling består av 1500 tilgjengelige biler, og hele bilparken er el-biler.

Forretningsideen er som følger: bilene står parkert på godkjente parkeringsplasser, og den som leier en bil må parkere på en slik plass innen et avgrenset geografisk område som er definert i app'n. Bilen kan benyttes utenfor dette området i en leieperiode, men må alltid parkeres innenfor det geografiske området (leieforholdet lar seg ikke avslutte hvis dette ikke er tilfellet). En kunde som ønsker å leie en bil kan bruke app'n til å se hvor biler står parkert. Leie av bil gjøres via app'n på mobilen når kunden står ved en bil, og som frigjør/låser opp bilen. Som en del av abonnementsavtalen med BilDeling er kunden forpliktet til alltid å levere bilen fulladet.

For spørsmål hvor du skal programmere mot sekvensielle tekstfiler er filstrukturen som følger:

bil.txt, for hver bil lagres <Regnr, Merke, Modell, Startdato, Posisjon>

EE44444

Hyundai

Kona Electric

20102023

Stortorvet 36

...OSV

kunde.txt, for hver kunde lagres <Mobilnr, Fornavn, Etternavn, Betalingskortnr>

+4798765432

Anton

Antonsen

1111222233334444

...OSV

KANDIDATEN MÅ SELV KONTROLLERE AT OPPGAVESETTET ER FULLSTENDIG

utleie.txt, for hvert utleie lagres <Regnr, Utlevert, KmUt, Mobilnr, Innlevert, KmInn, Beløp>

EE44444

08052024

23456

+4798765432

10052024

23865

984

...OSV

Ved et påbegynt utleie får feltene Innlevert, KmInn og Beløp verdien 'NULL', som endres til aktuell verdi ved innleveringen. En slik post i utleie.txt vil da se sånn ut:

EB45678

24052024

34234

+4712345678

NULL

NULL

NULL

Krav og tips til filhåndteringen:

- I filene ligger det **ett felt pr linje**. Strukturen < Regnr, Merke, Modell, Startdato, Posisjon > er bare for å vise hva en post består av
- All filhåndtering skal være at en leser linje for linje («post for post») fra fil med test på end-of-file og skriver linje for linje («post for post») til fil
- Filene er lagret i samme katalog som programmet

For spørsmål hvor du skal programmere mot database er databasestrukturen som følger:

Bil(Regnr, Merke, Modell, Startdato, Posisjon)

Kunde(Mobilnr, Fornavn, Etternavn, Betalingskortnr)

Utleie(Regnr*, Utlevert, KmUt, Mobilnr*, Innlevert, KmInn, Beløp)

Ved et påbegynt utleie får feltene Innlevert, KmInn og Beløp NULL-verdi, som endres til aktuell verdi ved innleveringen.

Det skal kobles mot databasen via **mysql.connector** etter følgende «koblingsstreng»:

- `mysql.connector.connect(host='localhost', port=3306, user='Bilsjef ', passwd='eksamen2024', db='Bildeling')`

Grafisk brukergrensesnitt, skal være basert på:

- tkinter
- grid som geometry manager

Prosedural programmering, TUI-basert grensesnitt, mot filstruktur**Oppgave 1 (vekt 1), lagres som <kandidatnr>-oppgave1.py**

Lag et program som leser all informasjon om bilene inn i en to-dimensjonal liste, og som skriver ut Regnr, Merke og Startdato for alle bilene ved en gjennomgang av lista.

Oppgave 2 (vekt 1), lagres som <kandidatnr>-oppgave2.py

Lag et program som leser all informasjon om kundene inn i en to-dimensjonal dictionary og som skriver ut Fornavn og Etternavn for en bestemt kunde når programmet spør brukeren om å oppgi Mobilnr. Programmet skal kjøre så lenge brukeren ønsker.

Prosedural programmering, GUI-basert grensesnitt, mot database**Oppgave 3 (vekt 2), lagres som <kandidatnr>-oppgave3.py**

Lag et program hvor du søker på bilens registreringsnr og får en liste over alle utleier for denne bilen, sortert på utleiedato. Når en velger et utleieforhold i lista får en informasjon om hvem som har leid bilen (mobilnr, fornavn og etternavn). Vinduet skal se slik ut:

Bil med utleieforhold og kundeopplysninger

Oppgi regnr:

Utleieforhold

2024-04-11 19:15:32
2024-04-12 07:55:32
2024-05-01 12:15:00

Kundens mobilnr er:

Fornavn:

Etternavn:

KANDIDATEN MÅ SELV KONTROLLERE AT OPPGAVESETTET ER FULLSTENDIG

Oppgave 4 (vekt 2), lagres som <kandidatnr>-oppgave4.py

Lag et program for å avslutte et utleieforhold. Brukeren oppgir bilens registreringsnr og informasjon om utleieforholdet vises. Brukeren registrerer resterende informasjon om utleieforholdet og ber om at utleieforholdet avsluttes og databasen oppdateres. Vinduet skal se slik ut:

The screenshot shows a GUI window titled "Bil leveres tilbake" with standard window controls (minimize, maximize, close). The interface is organized into sections:

- Top section:** "Oppgi regnr:" with a text input field containing "AA11111" and a button "Finn utleieforhold".
- Informasjon om utleieforholdet:**
 - "Start på leieforholdet:" with a date/time input field "2024-05-01 12:15:00".
 - "Kmstand utlevert:" with a text input field "14500".
 - "Leid av:" section with "Fornavn:" (input "Test") and "Etternavn:" (input "Testesen").
- Avslutning av utleieforholdet:**
 - "Tidspunkt innlevert:" with a date/time input field "2024-05-03 11:30:00".
 - "Parkeringssted:" with a text input field "Bislett stadion".
 - "Kmstand innlevert:" with a text input field "15450".
 - "Leiebeløp:" with a text input field "1730".
 - A button "Avslutt utleieforhold" is located to the right of the "Leiebeløp" field.
- Bottom right:** A button "Avslutt".

Objektorientert programmering, GUI-basert grensesnitt, mot database**Oppgave 5 (vekt 2), lagres som <kandidatnr>-oppgave5.py**

Problemstillingen i oppgave 4 skal nå programmeres objektorientert.

- Lag klassediagram for de(n) klassen(e) du trenger. Objektene instansieres ved data overført fra feltene i GUI-laget
- Lag ny kode for å «avslutte et utleieforhold», krav til løsningen:
 - objektene instansieres ved data overført fra feltene i GUI-laget
 - oppdatering av databasen skal skje ved substitusjon i sql-setninger hvor verdiene hentes fra objektenes instansevariable
 - du trenger ikke å reproducere koden for vinduet i oppgave 4

KANDIDATEN MÅ SELV KONTROLLERE AT OPPGAVESETTET ER FULLSTENDIG