Informasjon

UNIVERSITETET I OSLO
Det matematisk-naturvitenskapelige fakultet
Skriftlig eksamen i IN1000
2022 HØST
Varighet: 2.12.2022, 15:00 til 2.12.2022, 19:00
Tillatte hjelpemidler: Alle trykte og skrevne, ingen elektroniske.

Faglærere vil besøke lokalet ca kl 16-17.

Dersom du mener en programmeringsoppgave er uklar, kan du gjøre egne forutsetninger og beskrive disse.

Du kan veksle mellom ulike språk (bokmål, nynorsk eller engelsk) øverst på siden.

1 Oppgave 1a

Hva skrives ut når følgende kode kjøres?

```
a = 3
if a > 5:
    a = a*2
print(a)
```

Maks poeng: 1

2 Oppgave 1b

Hva skrives ut når koden under kjøres?

```
a = 5
b = a > 2
if a == 3 or b:
    a += 10
print(a)
```

Maks poeng: 1

3 Oppgave 1c

Hva skrives ut når koden under kjøres?

```
x=5
y=1
while x >= y:
    x = x-y
    y = y*2
print(y-x)
```

Maks poeng: 1

4 Oppgave 1d

Hva blir skrevet ut når koden under kjøres?

```
a = 0
total = 0
while a<4:
    a += 3
for b in range(a-1):
     total += b
print(total)
```

Maks poeng: 2

5 Oppgave 1e

Hva skrives ut når koden under kjøres?

```
total = 0
ordbok = {1:3, 6:5, 2:7}
for k in ordbok:
    if ordbok[k] < k:
        total += k
print(total)
```

Maks poeng: 2

6 Oppgave 1f

Hva skrives ut når følgende kode kjøres?

```
total = 0
s = "kokos og solkrem"
for i in range(1,len(s)):
    if s[i]=="o" and s[i-1] in ["k", "s"]:
        total += 1
print(total)
```

Maks poeng: 2

7 Oppgave 2a

Gitt følgende kode for hele oppgave 2:

```
class Barnehage:
    def __init__(self, barn):
        self._barn = barn
    def bursdag(self):
        self._barn.bursdag()
    def bytte(self, nytt_barn):
        self._barn = nytt_barn
class Person:
    def __init__(self, alder):
        self._alder = alder
    def bursdag(self):
        self._alder += 1
    def hent_alder(self):
        return self._alder
per = Person(2)
palle = Person(5)
maurtua = Barnehage(per)
per.bursdag()
maurtua.bursdag()
print("A:", per.hent_alder())
maurtua.bytte(palle)
palle.bursdag()
print("B:", per.hent_alder())
print("C:", palle.hent_alder())
maurtua.bytte(Person(1))
print("D:", palle.hent_alder())
```

Hva skrives ut i linja som inkluderer "A:" i koden over?

Velg ett alternativ:

- A: 3
- A: 4
- A: 2

Maks poeng: 2

8 Oppgave 2b

Gitt samme kode som i oppgave 2a, hva skrives ut i linja som inkluderer "B:" ?

(Nøyaktig samme kode som i oppgave 2a er forøvrig gjengitt her under for å ha det lettest mulig tilgjengelig):

```
class Barnehage:
    def __init__(self, barn):
        self._barn = barn
    def bursdag(self):
        self._barn.bursdag()
    def bytte(self, nytt_barn):
        self._barn = nytt_barn
class Person:
    def __init__(self, alder):
        self._alder = alder
    def bursdag(self):
        self._alder += 1
    def hent_alder(self):
        return self._alder
per = Person(2)
palle = Person(5)
maurtua = Barnehage(per)
per.bursdag()
maurtua.bursdag()
print("A:", per.hent_alder())
maurtua.bytte(palle)
palle.bursdag()
print("B:", per.hent_alder())
print("C:", palle.hent_alder())
maurtua.bytte(Person(1))
print("D:", palle.hent_alder())
```

Velg ett alternativ:

- B: 5
- B: 2
- B: 4
- B: 3

Maks poeng: 2

9 Oppgave 2c

Gitt samme kode som i oppgave 2a, hva skrives ut i linja som inkluderer "C:" ?

(Nøyaktig samme kode som i oppgave 2a er forøvrig gjengitt her under for å ha det lettest mulig tilgjengelig):

```
class Barnehage:
    def __init__(self, barn):
        self._barn = barn
    def bursdag(self):
        self._barn.bursdag()
    def bytte(self, nytt_barn):
        self._barn = nytt_barn
class Person:
    def __init__(self, alder):
        self._alder = alder
    def bursdag(self):
        self._alder += 1
    def hent_alder(self):
        return self._alder
per = Person(2)
palle = Person(5)
maurtua = Barnehage(per)
per.bursdag()
maurtua.bursdag()
print("A:", per.hent_alder())
maurtua.bytte(palle)
palle.bursdag()
print("B:", per.hent_alder())
print("C:", palle.hent_alder())
maurtua.bytte(Person(1))
print("D:", palle.hent_alder())
```

Velg ett alternativ:

- C: 6
- C: 5
- C: 7

Maks poeng: 2

10 Oppgave 2d

Gitt samme kode som i oppgave 2a, hva skrives ut i linja som inkluderer "D:" ?

(Nøyaktig samme kode som i oppgave 2a er forøvrig gjengitt her under for å ha det lettest mulig tilgjengelig):

```
class Barnehage:
    def __init__(self, barn):
        self._barn = barn
    def bursdag(self):
        self._barn.bursdag()
    def bytte(self, nytt_barn):
        self._barn = nytt_barn
class Person:
    def __init__(self, alder):
        self._alder = alder
    def bursdag(self):
        self._alder += 1
    def hent_alder(self):
        return self._alder
per = Person(2)
palle = Person(5)
maurtua = Barnehage(per)
per.bursdag()
maurtua.bursdag()
print("A:", per.hent_alder())
maurtua.bytte(palle)
palle.bursdag()
print("B:", per.hent_alder())
print("C:", palle.hent_alder())
maurtua.bytte(Person(1))
print("D:", palle.hent_alder())
```

Velg ett alternativ:

- D: 6
- D: 7
- D: 1
- D: 5

Maks poeng: 2

11 Oppgave 3a (4p)

Per, Palle og Espen lurer på om de skal finne på noe til kvelden. Dersom nøyaktig to av dem ønsker å spille badmington går de for det, men ellers (om færre enn to eller flere enn to vil dette) gjør de heller noe annet.

Skriv en funksjon ***badmington(per_vil, palle_vil, espen_vil)*** som tar inn tre boolske argumenter, og returnerer *True* dersom to (og kun to) av de tre argumentene er *True*.

Altså skal for eksempel kallet *badmington(True, True, False)* returnere *True*, mens kallet *badmington(True, True, True)* skal returnere *False*

Skriv ditt svar her

Maks poeng: 4

12 Oppgave 3b (5p)

Du er medgangs-supporter for fotball-laget Brann, noe som betyr at så lenge Brann gjør det høvelig bra heier du på dem, men om de gjør det veldig dårlig er du kynisk og velger heller å heie på laget som leder på tabellen.

Skriv en funksjon ***heie(tabellplass_ordbok)*** som tar som argument en ordbok hvor nøklene er navn på fotball-lag og verdiene er plassering på tabellen. Dersom verdien i ordboka for nøkkelen "Brann" er 3 eller mindre, skal funksjonen returnere strengen "Brann". Ellers skal funksjonen returnere nøkkelen som i ordboka er koblet til verdien 1.
Du kan anta at nøkkelen *"Brann"* finnes i *tabellplass_ordbok*, og du kan anta at *tabellplass_ordbok* har et lag med verdi 1.

Altså skal kallet *heie({"Rosenborg":4, "Odd":1, "Molde":3, "Brann":2})* returnere *"Brann"*, mens kallet *heie({"Rosenborg":2, "Odd":1, "Molde":3, "Brann":4})* skal returnere *"Odd"*.

Skriv ditt svar her

Maks poeng: 5

13 Oppgave 3c (7p)

Du skal fordele gjester langs den ene siden av et langt bord. Du har på forhånd delt personene inn i to grupper - introverte og ekstroverte - og ønsker å fordele disse på annenhver plass langs bordet.

Skriv en funksjon ***bordsetting(introverte, ekstroverte)*** som tar som argument to lister med strenger, og som returnerer én enkelt liste som inneholder alle strengene fra de to listene mottatt som argument. Annenhver streng i den nye listen skal være fra listen introverte og annenhver fra ekstroverte. Det er ellers vilkårlig hvordan rekkefølgen av strengene er. Du kan anta at listene *introverte* og *ekstroverte* er like lange.

Altså kan kallet *bordsetting(["Per","Palle","Espen"], ["Putti", "Plutti", "Pott"])* for eksempel evaluere til listen *["Per","Putti","Palle","Plutti","Espen","Pott"]*.

Skriv ditt svar her

Maks poeng: 7

14 Oppgave 3d (7p)

Du ønsker hjelp til å gruppere sammen personer som har felles interesse.

Skriv en funksjon ***lag_interessegrupper(personers_interesse)*** som tar som argument en ordbok hvor nøklene er strenger som er navn på personer, mens hver verdi er en streng som sier hvilken interesse personen har. Funksjonen skal returnere én enkelt ordbok, hvor hver nøkkel er en interesse (streng) og hver verdi er en liste av personer som var oppført med denne interessen i ordboken personers_interesse. Alle interessene som fantes i ordboken personers_interesse skal være med i den returnerte ordboken.

Altså skal for eksempel kallet *lag_interessegrupper({"Per":"Mat", "Palle":"Film", "Espen":"Mat"})* evaluere til ordboken *{"Mat":["Per","Espen"], "Film":["Palle"]}*

Skriv ditt svar her

Maks poeng: 7

15 Oppgave 4

Oppgaveteksten til oppgave 4 ligger i vedlagte pdf-dokument. Du kan legge til egne metoder i løsningen din ved behov, kommenter i så fall disse kort. Du kan også bruke metoder beskrevet i oppgaveteksten selv om du ikke selv skriver dem.

Skriv ditt svar her

Maks poeng: 50

16 Oppgave 5a (5p)

Du ønsker hjelp til å holde orden på hvor du har havnet etter at du har kastet terningen en del ganger i et spill som kalles Stigespillet. Reglene er som følger: man begynner på rute 0, etter hvert kast flytter man frem like mange ruter som antall øyne på terningen, og om man lander på en rute hvor en stige begynner følger man stigen frem til ruten som stigen ender opp på.

Skriv en funksjon stigespill(terningkast, stiger) med to parametre: terningkast er en liste med terningkast (heltall), mens stiger er en ordbok hvor både nøkkel og verdi er heltall (som representerer hver sin rute på spillebrettet). Anta at en spiller begynner på posisjon 0 (på spillebrettet). For hver verdi i listen terningkast øker du posisjonen med denne verdien og sjekker om den oppdaterte verdien tilsvarer en av nøklene i ordboka stiger (om det er en stige som begynner på den resulterende verdien for posisjon). Om det finnes en slik nøkkelverdi endrer du posisjon til å bli verdien tilhørende denne nøkkelen i ordboka stiger. Deretter fortsetter du med neste verdi i lista terningkast og gjør samme sjekk mot ordboka stiger. Funksjonen skal returnere verdien til posisjon etter at du har gått gjennom alle verdier i lista terningkast på denne måten.

Altså skal for eksempel kallet stigespill([5,4,2,2],{5:12, 18:7}) evaluere til heltallet 9.

Skriv ditt svar her

Maks poeng: 5

17 Oppgave 5b (5p)

Per spiller stigespillet beskrevet i oppgave 5a (med de samme stigene). Etter tre kast står han på rute 5. Hva kan det være at han har fått på tre etterfølgende kast som gjorde at han endte på akkurat rute 5?

Skriv en funksjon ***hvilke_tre_kast(slutt_rute, stiger)*** som tar som argument et heltall *slutt_rute* (som er ruten man har endt på), en ordbok *stiger* (samme som i oppgave 5a) og som returnerer en nøstet liste. I den returnerte nøstede listen skal hver indre liste bestå av tre heltall (tallene for tre etterfølgende kast), og den ytre listen skal bestå av alle de ulike mulige indre listene som ville gjøre at man i stigespillet (som definert i oppgave 5a) ender opp på *slutt_rute*.

Hint: Du kan gjerne gjøre bruk av funksjonen du skrev i oppgave 5a som del av løsningen på denne oppgaven.

Altså skal for eksempel kallet *hvilke_tre_kast(5, {3:15, 17:4})* returnere *[[1, 1, 3], [1, 3, 1], [2, 2, 1], [3, 2, 1]]* (eller de samme indre listene i en annen rekkefølge)

Skriv ditt svar her

Maks poeng: 5

Oppgave 4. Reservasjonssystem (50 poeng)

I denne oppgaven skal du skrive en første versjon av et program for reservasjon av rom på ett eller flere hoteller. I denne versjonen håndterer programmet kun reservasjoner for kommende natt. Programmet består av fire klasser du skal skrive (du trenger ikke skrive noe hovedprogram):

Reservasjonssystem. Systemet håndterer ett eller flere hoteller og støtter to typer reservasjoner:

- Reservasjon av et rom med kjent rom-nummer på et navngitt hotell
- Gruppereservasjoner der reservasjonssystemet selv finner og reserverer *passende rom*[1] til så mange som mulig i gruppen, uavhengig av hotell.

Gruppe. En gruppe består av en eller flere personer som ønsker å bestille sammen, og som har felles krav til fasiliteter på rommet/ rommene de ser etter. Personer i en gruppe bryr seg ikke om hvem de blir plassert på rom med så lenge de er i samme gruppe.

Hotell. Et hotell har et navn og en samling gjesterom av klassen **Rom**.

Rom. Hvert rom har et romnummer og et antall senger. Rommet har null eller flere fasiliteter (for eksempel bad eller utsikt). Om rommet er utleid har det en eller flere gjester.

Du velger selv hensiktsmessige instansvariabler utover de som er oppgitt. Instansvariabler skal ikke aksesseres utenfra klassen (de er non-public). Personer representeres kun med navn (en streng som kan inneholde blanke). Fasiliteter og krav til fasiliteter representeres med en tekstlig beskrivelse (en streng uten blanke, for eksempel "utsikt" eller "peis").

Husk å bruke metoder i andre klasser der dette er hensiktsmessig. Du kan anta at returverdier og argumenter til metoder er gyldige, og trenger ikke fange opp eller sjekke for feilsituasjoner.

Oppgave 4a) (3 poeng)

Skriv klassen **Gruppe**. Konstruktøren skal ha én parameter; en liste med krav til fasiliteter i rom. Klassen skal tilby følgende metoder i tillegg til konstruktør:

- **legg_til_personer().** Parameter er en liste med personnavn som legges til gruppen (i tillegg til de som eventuelt var med i gruppen fra før)
- **hent_personer().** Returnerer navnene i gruppen
- **hent_krav()**. Returnerer fasilitetene gruppen krever i sine rom.

Oppgave 4b) (8 poeng)

Skriv klassen **Rom**. Romnummer, antall senger og fasiliteter er parametere til konstruktøren. Klassen skal tilby følgende metoder i tillegg til konstruktør:

- **reserver().** Metoden har en parameter som angir navn på en eller flere gjester som skal bo på rommet. Metoden registrerer at rommet er opptatt og navn på gjesten(e).
- **hent_ant_senger().** Returnerer antall senger i rommet.
- **passer().** Metoden har en parameter for krav til fasiliteter og returnerer True hvis rommet *passer*, ellers False. Metoden tar ikke hensyn til antall senger.
- **__str__().** Returnerer en streng med romnummer og antall senger, samt fasiliteter i rommet.

[1] *passende rom er i denne oppgaven definert som ledige rom som har alle fasiliteter som kreves av en gruppe*

Oppgave 4c) (10 poeng)

Skriv klassen **Hotell**. Hotellnavnet er parameter til konstruktøren, som leser all informasjon om rommene i hotellet fra en fil med navn *hotellnavn.txt* der *hotellnavn* er navnet på hotellet. Hvert rom har en linje i filen. En slik linje inneholder ett heltall for romnummer, ett heltall for antall senger, og null eller flere ord der hvert ord angir en fasilitet. En fil med to rom kan f. eks. se slik ut:

400 4
401 2 bad utsikt peis

Rommene i et hotell lagres i en ordbok der nøklene er romnumre og referanser til objekter av klassen **Rom** er verdier. Klassen skal tilby følgende metoder i tillegg til konstruktør:

- **reserver_rom().** Reserverer et rom for en eller flere gjester. Romnummer og navn på gjesten(e) er parametere.
- **finn_passende_rom()**. Metoden har en parameter som angir krav til fasiliteter, og returnerer en liste med **Rom**-objekter som passer. Metoden tar ikke hensyn til antall senger.

Oppgave 4d) (7 poeng)

I denne og de neste deloppgavene skal du skrive klassen **Reservasjonssystem**. Konstruktøren tar en liste med hotellnavn som parameter, og oppretter en ordbok der hotellnavn er nøkler og referanser til hotell-objekter verdier. Du kan anta at ingen hotell-objekter er opprettet før det opprettes et **Reservasjonssystem**. I denne deloppgaven skal du foruten konstruktøren skrive følgende metode:

- **reserver_rom_i_hotell()**. Metoden reserverer et rom i et hotell. Parametere til metoden er hotellnavn, romnummer og navn på gjesten(e) som skal bo i rommet.

Oppgave 4e) (7 poeng)

I denne deloppgaven skal du skrive en intern (non-public) hjelpemetode i **Reservasjonssystem**:

- **_finn_alle_passende_rom**(). Metoden har en parameter for fasiliteter som kreves og returnerer en ordbok med alle passende rom i alle hoteller. Ordboken har hotellnavn som nøkler og en liste med ledige, passende **Rom**-objekter som verdi for hver nøkkel. Hoteller uten passende rom skal ikke være med i ordboken.

Oppgave 4f) (12 poeng)

I denne deloppgaven skal du skrive metoden i **Reservasjonssystem** som reserverer rom til en gruppe:

- **gruppe_reservasjon().** Metoden har en parameter for gruppen det skal reserveres rom til, og returnerer en bekreftelse på resultatet. Bekreftelsen er en liste med en eller flere strenger.

Dersom ingen rom i noen hoteller passer returneres strengen "Ingen passende rom", og ingen reservasjon utføres. Metoden går ellers gjennom passende rom i alle hoteller og bestiller ett og ett rom til så mange personer fra gruppen som det er senger i rommet. Når alle gjestene har fått rom, eller det ikke er flere passende rom, returnerer metoden en bekreftelse med oversikt over reserverte rom (med hotellnavn og informasjon om rommet). Dersom ikke alle i gruppen har fått rom skal bekreftelsen avsluttes med en advarsel om hvor mange personer som ikke har fått plass.

Oppgave 4g) 3 poeng (teori)

Hva er grensesnittet til klassen **Reservasjonssystem?** Skriv svaret i en kommentar øverst i programmet. Du kan bruke Python-kode, naturlig språk, eller en kombinasjon.