Forside

Oppgave	Tittel	Maks poeng	Oppgavetype
ℹ	Informasjon		Informasjon eller ressurser

Del 1 stikkord tracing (1 poeng)

Oppgave	Tittel	Maks poeng	Oppgavetype
1	if-else og konkatinering	1	Fyll inn tall
2	or	1	Fyll inn tekst
3	while mm	1	Fyll inn tall
4	while	1	Fyll inn tall
5	gjennomløp liste	1	Fyll inn tall
6	Opprette et objekt	1	Fyll inn tekst

Del 2 stikkord tracing (2 poeng)

Oppgave	Tittel	Maks poeng	Oppgavetype
7	Ordbok med lister	2	Fyll inn tall
8	Summer liste	2	Fyll inn tall
9	Funksjonskall i argument	2	Fyll inn tekst
10	Liste av lister	2	Fyll inn tall
11	Kalkulator	2	Fyll inn tall
12	class Ukedag	2	Fyll inn tall
13	class Bil	2	Fyll inn tall

Del 3 stikkord prosedural

Oppgave	Tittel	Maks poeng	Oppgavetype
14	Oppgavetype matrise	5	Paring
15	Beregne fridager	6	Programmering
16	Brukernavn	5	Programmering
17	Ledende nuller	6	Programmering
18	dempDeg	7	Programmering
19	Synonymordbok (nøtt)	6	Programmering

Del 4 stikkord oo

Oppgave	Tittel	Maks poeng	Oppgavetype
20	Navngi klasse-elementer	4	Fyll inn tekst i bilde
21	Større objektorientert system	50	Programmering

Informasjon

Skriftlig PRØVEeksamen i IN1000
2021 HØST
Varighet: 10.11.2021 til 16.11.2021. Gjenåpnes ca 17.11 til 1.12.2021
Det er viktig at du leser denne forsiden nøye før du starter.
Generell informasjon:

- Viktige beskjeder under eksamen blir gitt direkte fra faglærer på emnets semesterside. Det er derfor viktig at du sjekker emnets semesterside jevnlig.
- Besvarelsen din skal reflektere ditt eget, selvstendige arbeid og skal være et resultat av din egen læring og arbeidsinnsats.
- Alle hjelpemidler er tillatt ved skriftlig hjemmeeksamen. Dersom du gjengir tekst fra bøker, nettartikler eller lignende, så må det henvises til disse kildene i besvarelsen for å unngå mistanke om ulovlig tekstlikhet. Dette gjelder også dersom du oversetter tekst fra andre språk.
- Du er selv ansvarlig for å sørge for at eksamensbesvarelsen din ikke er tilgjengelig for andre under eksamenstiden, hverken fysisk eller digitalt.
- Husk at besvarelsen skal være anonym, du skal ikke oppgi hverken ditt eller medstudenters navn.
- Om du vil trekke deg fra eksamen, trykk på hamburgermeny oppe til høyre i Inspera og velg "Jeg vil trekke meg".

Samarbeid under eksamen:
Det er ikke tillatt å samarbeide eller kommunisere med andre under eksamen. Samarbeid og kommunikasjon vil bli betraktet som forsøk på fusk. Det blir gjort plagiatkontroll av alle innleverte eksamener der tekstlikhet/ strukturlikhet mellom besvarelser blir sjekket. Om du bruker notater som er utarbeidet i samarbeid med andre før eksamen, kan dette gi treff i en plagiatsjekk. Slik tekstlikhet kan bli betraktet som lav selvstendighet eller forsøk på fusk. Unngå derfor klipp/lim fra notater laget i samarbeid med andre.
Fusk:
Les om hva som regnes som fusk på UiOs nettsider.

Kontaktinfo:

Brukerstøtte eksamen (for prøveeksamen: Bruk Mattermost)

1 if-else og konkatinering

```
1  tall =     # hvilket heltall skal stå her?
2  t1 = "a"
3  if tall < 8:
4      t1 = t1 + "a"
5  else:
6      t1 = t1 + "b"
7
8  tekst = "b" + t1
9
10 # skriv det største heltallet som gir tekst verdien "baa"
11
```

Hva er det *største* mulige heitallet som kan tilordnes variabelen **tall** i linje 1 slik at variabelen **tekst** får verdien "**baa**" ?: [ ] (7).

Maks poeng: 1

2 or

Skriv et logisk uttrykk i svarfeltet, slik at utskriften alltid blir "**ja**". Bruk operatoren **>** og heltallene **3** og **4.**

```
if  False or [ ] (4>3) :
    print("ja")
else:
    print("nei")
```

Maks poeng: 1

3 while mm

```
a=1
while a<10:
    a = 3*a
    for b in [3,2,1]:
        a += b
if a<25:
    a = a*2
    while a < 100:
        a += 1
while a<29:
    a = a + 3
print(a)
```

Hva blir skrevet ut her? (skriv bare verdien)

(33)

Maks poeng: 1

4 while

```
tall = 5
while tall < 10:
    tall = tall - 2
    tall = tall * 2

tall = tall -  ?
```

Hvilket helltall mangler i uttrykket på nederste linjen for at variabelen **tall** skal få verdien 10 etter at denne koden er kjørt?

Skriv kun tallet som skal erstatte spørsmålstegnet i koden (du skal ikke endre noe):

(2)

Maks poeng: 1

5 gjennomløp liste

Skriv en verdi i svarfeltet slik at utskriften alltid blir "**ja**".

```
liste = [-1, 2, 322, 95]
s = 0
for tall in liste:
    s = s + liste[1]
if s ==
```

[ ] (8) :

```
    print("ja")
else:
    print("nei")
```

Maks poeng: 1

6 Opprette et objekt

Espens fulle navn er **Espen Askeladd**. Hvordan oppretter du et nytt objekt referert av **helt** slik at navnet blir satt riktig i objektet? Fyll ut svarboksen med høyresiden som mangler i siste kodelinje.

```
class Info:
    def __init__(self, fornavn, etternavn):
        self._navn = fornavn + " " + etternavn
```

helt = [ ] (Info("Espen", "Askeladd"), Info('Espen', 'Askeladd'))

Maks poeng: 1

7 Ordbok med lister

```
ordbok = {'aa': [0, 0, 4, 1, 4, 1, 4, 1, 4, 5, 0, 0, 4, 4, 7, 8], \
          'bb': [8, 6, 4, 2, 2, 0, 1, 8, 4, 3, 6, 2], \
          'cc': [], \
          'dd': [0, 8, 3, 6, 5, 0, 3, 7, 3, 6, 1, 7, 6, 6], \
          'ee': [3, 0, 0, 7, 4, 2, 0, 8, 8, 5, 9, 6, 4], \
          'ff': [4, 7, 4], \
          'gg': [2, 2, 6, 8, 6, 0], \
          'hh': [7, 1, 9, 3, 6, 5, 8, 6, 7, 3, 9, 7, 0], \
          'ii': [], \
          'jj': [0, 2, 1, 3, 5, 8, 4, 8, 8, 2, 9, 8, 5, 4, 4, 7, 8, 6], \
          'kk': [4, 5, 8, 8, 4, 8, 9, 6, 4], \
          'll': [9, 4, 3, 4, 2, 0, 2, 0, 4, 5], \
          'mm': [3, 9, 2, 3, 9, 6, 0, 2]}
```

Hva er verdien til **svar** når koden nedenfor er utført? **ordbok** er initialisert som i bildet over.

```
ordbok["ff"].append(7)
svar = len(ordbok["ff"])
```

Skriv svaret her: (4)

Maks poeng: 2

8 Summer liste

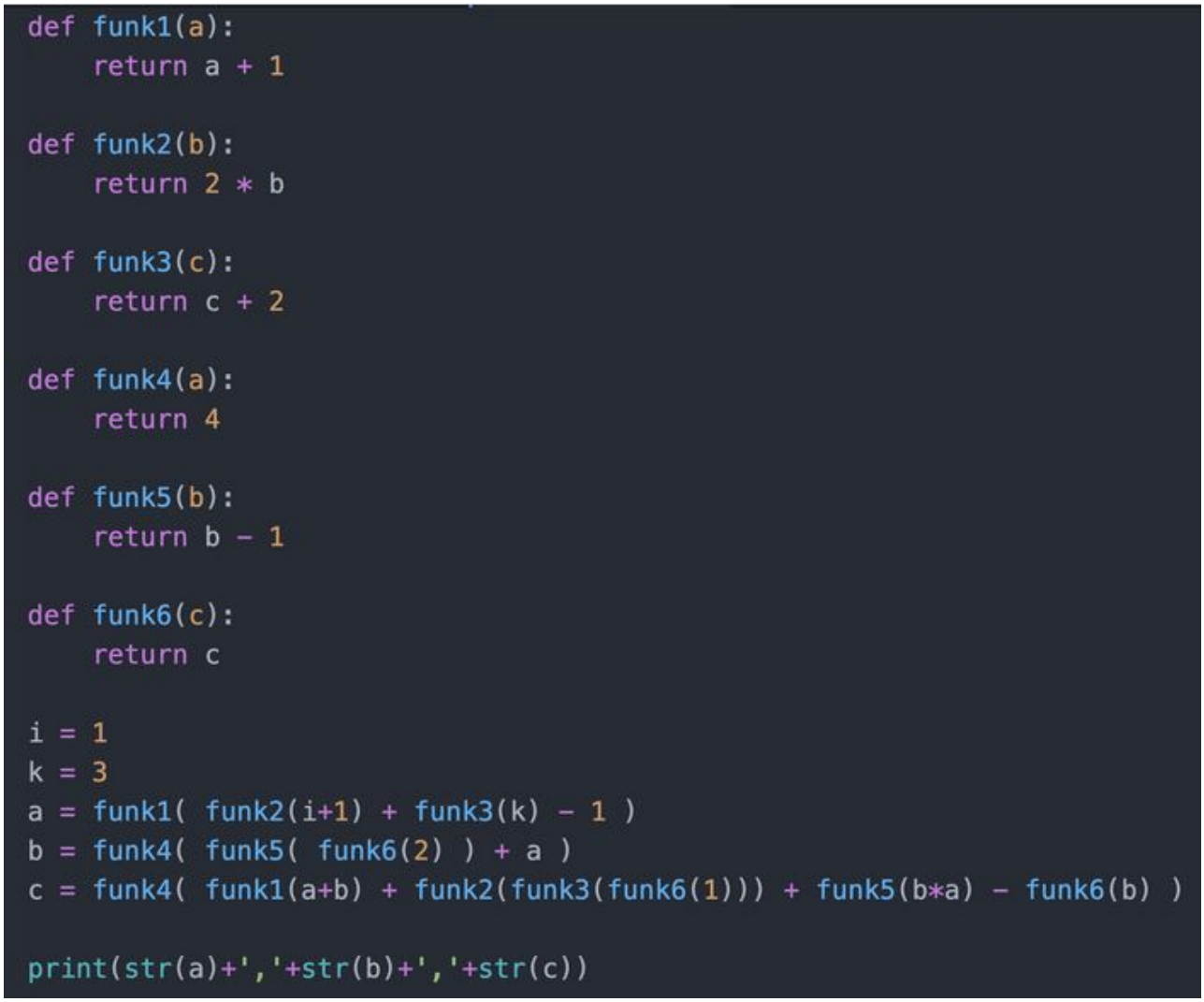
```
liste = [1,1,-1,2,-2,3,-1,3,5,-2,3,1,-7,8,-9,3,2,-3,6,5,4,-2,3,2]
total=0
i = 0
while liste[i]>0:
    total += liste[i]
    i += 1
i=len(liste)-1
while liste[i]>0:
    total += liste[i]
    i -= 1
i = 0
while liste[i] > len(liste):
    total += liste[i]
    i += 1
print(total)
```

Hva skrives ut her? (7)

Maks poeng: 2

9 Funksjonskall i argument

I denne oppgaven skal du vise at du kjenner til hva som skjer når vi bruker funksjonskall i argumentet til et annet funksjonskall.



```
def funk1(a):
    return a + 1

def funk2(b):
    return 2 * b

def funk3(c):
    return c + 2

def funk4(a):
    return 4

def funk5(b):
    return b - 1

def funk6(c):
    return c

i = 1
k = 3
a = funk1( funk2(i+1) + funk3(k) - 1 )
b = funk4( funk5( funk6(2) ) + a )
c = funk4( funk1(a+b) + funk2(funk3(funk6(1))) + funk5(b*a) - funk6(b) )

print(str(a)+','+str(b)+','+str(c))
```

Hva blir skrevet ut av koden i bildet?
Skriv svaret her (nøyaktig slik det skrives ut på terminalen):

(9,4,4)

Maks poeng: 2

10 Liste av lister

```
matrise = [ [60, 28, 77, 23, 84, 10, 10, 73, 83, 35, 15, 10, 22, 18, 88],  \
            [92, 94, 31, 55, 63, 85, 62, 85, 10, 80, 54, 96, 80, 16, 68],  \
            [56, 53, 91, 25, 69, 18, 94, 85, 11, 86, 98, 70, 64, 44, 42],  \
            [11, 59, 44, 29, 36, 44, 20, 78, 99, 21, 53, 22, 24, 73, 64],  \
            [18, 11, 75, 82, 27, 52, 69, 76, 61, 50, 91, 24, 49, 82, 20],  \
            [67, 39, 27, 14, 49, 39, 63, 60, 78, 55, 71, 33, 15, 27, 82],  \
            [25, 22, 70, 32, 47, 32, 97, 19, 56, 24, 59, 25, 63, 25, 65],  \
            [88, 23, 34, 34, 91, 18, 28, 73, 40, 58, 95, 86, 59, 75, 86],  \
            [90, 86, 56, 94, 35, 83, 11, 64, 62, 64, 66, 75, 77, 62, 80] ]
```

Hvilken verdi evaluerer uttrykket **matrise[0][2] + matrise[7][1]** til? (100)

Maks poeng: 2

11 Kalkulator

```
def kalkulator(operasjoner): # operasjoner må ha type str
    tall = 0
    for operasjon in operasjoner:
        if operasjon == "d":
            tall = tall * 2
        elif operasjon == "1":
            tall = tall + 1
        elif operasjon == "2":
            tall = tall + 2
        elif operasjon == "3":
            tall = tall + 3
        elif operasjon == "4":
            tall = tall + 4
        elif operasjon == "5":
            tall = tall + 5
        elif operasjon == "6":
            tall = tall + 6
        elif operasjon == "7":
            tall = tall + 7
        elif operasjon == "8":
            tall = tall + 8
        else:
            tall = tall
    return tall
```

Hva blir returnert hvis vi kaller denne funksjonen med argumentet **"1d7d8"** ?

kalkulator("1d7d8") returnerer verdien..: (26)

Maks poeng: 2

12 class Ukedag

Hvor mange Ukedag-objekter eksisterer (kan man få tak i) etter at koden på bildet har kjørt?

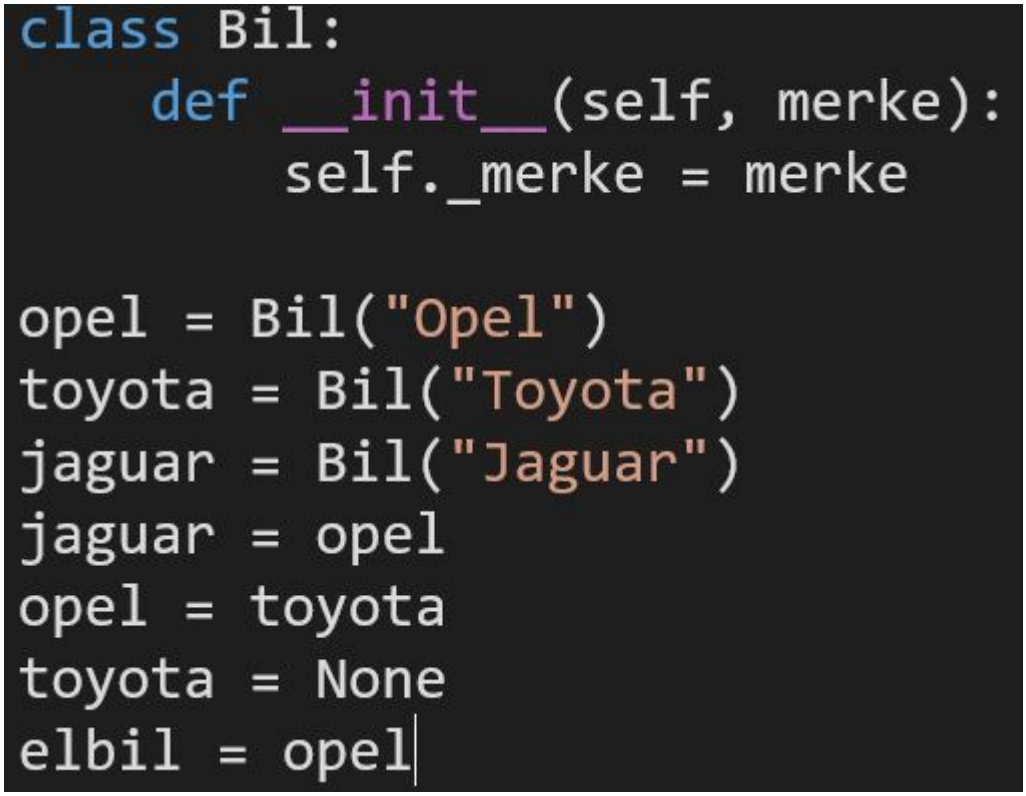
```
class Ukedag:
    def __init__(self, dag):
        self.dag = dag

torsdag = Ukedag(4)
imorgen = Ukedag(4+1)
fredag = imorgen
helg = Ukedag(6)
fredag = None
helg = None
lordag = helg
```

Skriv svaret her: (2)

Maks poeng: 2

13 class Bil

Hvor mange bil-objekter eksisterer (kan man få tak i) etter at koden på bildet har kjørt?



```
class Bil:
    def __init__(self, merke):
        self._merke = merke

opel = Bil("Opel")
toyota = Bil("Toyota")
jaguar = Bil("Jaguar")
jaguar = opel
opel = toyota
toyota = None
elbil = opel
```

Skriv svaret her: (2)

Maks poeng: 2

14 Oppgavetype matrise

Erstatt med oppgavetekst.

Velg den kolonnen som beskriver koden til venstre i hver rad mest korrekt

	Evaluerer til string	Evaluerer til boolean	Ikke et gyldig uttrykk
navn = "Jonas"	○	○	○ ✔
str(len(["to", "tre"]))	○ ✔	○	○
while not ferdig:	○	○	○ ✔
False	○	○ ✔	○
input("Navn: ")	○ ✔	○	○

Maks poeng: 5

15 Beregne fridager

Skriv en funksjon **fridager(julaften)** som returnerer antall fridager fra og med 25. desember, til og med 31. desember. Det er alltid fri lørdag og søndag, dessuten 25. og 26. desember. Om noen av disse faller på samme dag blir det færre enn 4 dager fri. Julaften er 24. desember.

Parameteren **julaften** skal inneholde en streng som angir hvilken ukedag julaften faller på. Det vil si at kallet **fridager("sondag")** skal evaluere til **4,** mens **fridager("fredag")** skal evaluere til **2.**

.

Skriv ditt svar her

Maks poeng: 6

16 Brukernavn

Skriv en funksjon **beOmNavn(navneliste)** med en liste av strenger som parameter. Funksjonen skal be bruker oppgi navn inntil navnet som oppgis finnes i **navneliste**, da returneres dette navnet. Navnelisten skal ikke skrives ut. Du kan anta at alle navn i listen og som oppgis av bruker kun inneholder små bokstaver (lowercase).

Skriv ditt svar her

Maks poeng: 5

17 Ledende nuller

Funksjonen *stringFraInt* i koden nedenfor returnerer en tekst med 6 tegn ved å fylle inn '0'-tegn foran heltallet i parameteren tall.

Skriv en ny funksjon **intTilString(tall, antallSiffer)**, som returnerer en tekst med en lengde som er gitt av parameteren **antallSiffer**. Dersom lengden (antall siffer) av parameteren **tall** er mindre enn **antallSiffer**, så skal funksjonen fylle på '0'-tegn foran heltallet i parameteren **tall**, slik som det er gjort i *stringFraInt*.

Skriv ditt svar her

Maks poeng: 6

18 dempDeg

Mange irriterer seg over overdrevet bruk av utropstegn. Skriv en funksjon **dempDeg (tekst)** som returnerer en kopi av parameteren **tekst**, men der alle utropstegn ("!") bakerst i teksten er fjernet (om det er noen) og erstattet av et punktum (".").

For eksempel skal kallet **dempDeg("Hei!!!")** returnere strengen "Hei.", mens kallet **dempDeg(" Hei du! Og du.")** skal returnere strengen " Hei du! Og du." .

Skriv ditt svar her

Maks poeng: 7

19 Synonymordbok (nøtt)

Du har kommet over ei liste over synonymer (ord som betyr omtrent det samme). Lista er ei liste av lister, hvor ei av synonymlistene f.eks. kan være:

["godt", "bra", "flott", "ok", "fint", "vel"]

Du ønsker å lage en papirversjon av synonymordboka slik at man kan slå opp på et gitt ord for å finne alle ordets synonymer. Du skal lage en funksjon som returnerer ei Python-ordbok (dict) som hjelper deg i dette arbeidet:

lagSynonymordbok(listeAvLister)

Vær oppmerksom på homonymer. Det er ord som skrives likt, men betyr forskjellige ting. F.eks. kan ordet *gift* både være en relasjon mellom to personer, eller betegne noe som er farlig å få i seg. For å komme rundt dette problemet lages synonymorboka slik at den inneholder ei synonymliste *for hver betydning av et gitt ord* der ordet er et homonym. Dette betyr at ordlista har en streng som nøkkel og ei liste av lister av strenger som verdi. Et eksempel:

synonymordbok["blad"] = [["løv"], ["magasin"]]

fordi ordet *blad* både kan være en del av en plante, men også noe man kan kjøpe i en bladkiosk. Funksjonaliteten framgår forøvrig av testprogrammet nedenfor.

Skriv ditt svar her

Maks poeng: 6

20 Navngi klasse-elementer

Navngi elementene som starter til venstre for hver tekstboks

```
1 class Node :
2     def __init__(self, nytt) :
3         self._innhold = nytt
4         self._neste = None
5
6     def nyEtterfolger (self, ny) :
7         self._neste = ny
8
9     def hentNeste (self) :
0         return self._neste
1
2     def hentInnhold(self):
3         return self._innhold
4
5 minListe = Node("0")
6 tmp = minListe
7 for nr in range(1,5):
8     tmp.nyEtterfolger(Node(str(nr))) # Innholdet skal være en streng
9     tmp = tmp.hentNeste()
0
1 tmp = minListe
2 while tmp is not None:
3     print(tmp.hentInnhold())
4     tmp = tmp.hentNeste()
5
```

(Konstruktør)

(Metode)

Maks poeng: 4

21 Større objektorientert system

I denne oppgaven skal du skrive et større objektorientert system. Du skal skrive løsning til alle deloppgaver i den vedlagte PDF-filen i svar-feltet til denne oppgaven. Pass på at alle metoder ligger i den klassen de tilhører, og marker hvilken deloppgave koden svarer på i kommentarer over klassen/ metodene.

Du finner også PDF'en HER (klikk på "HER") , dersom du ønsker å åpne den i eget vindu.

Du står fritt til å innføre egne variabler, metoder og klasser utover de som er beskrevet i oppgaveteksten. Husk å unngå særnorske tegn og bokstaver i koden din.

Om du ønsker kan du skrive inn programmet ditt i den editoren du pleier å bruke og kopiere inn hit. Test i så fall med noen få linjer først om resultatet blir riktig og leservennlig i Inspera. Pass på at du får med alle klasser og metoder du har skrevet før tiden er ute - du blir kun vurdert for det som er levert her. Unngå import av biblioteker/ kode som ikke er nevnt/ inngår i oppgaven!

Skriv ditt svar her

Maks poeng: 50

Question 21
Attached

Oppgave 4 Administrasjon av obligatoriske oppgaver (50 poeng)

Universitetet i Ruritania skal digitalisere sin håndtering av studentenes obligatoriske oppgaver («obliger»), og du skal lage deler av en pilot med et objektorientert design basert på innkapsling, dvs. instansvariabler og non-public metoder (navn starter med **_**) skal kun benyttes i den klassen de er definert. **Det er i mange av deloppgavene nødvendig å bruke metoder fra andre klasser/ deloppgaver. Du kan bruke disse selv om du ikke har skrevet dem.**

Emnene har ulike antall obligatoriske oppgaver som kreves godkjent, og hver oblig i et emne må registreres manuelt hvert semester. I Ruritania må en student ha alle oppgaver gitt i emnet godkjent for å få gå opp til eksamen, men i denne piloten foretas det ikke kontroll av om studenten har fått godkjent forrige oblig før godkjenning av besvarelse for neste oblig. Foruten obliger har et emne registrerte studenter og ansatte rettere. Både rettere og studenter har unike brukernavn.

Innleveringen av besvarelser for en oblig foregår i et annet system som du ikke skal lage. Innleverings-systemet lager en fil for hver oblig, med et entydig navn etterfulgt av «.txt», for eksempel *oblig1.txt*. Filen inneholder en linje per student på følgende format (eksempel med to linjer):
student1 obl1student1.txt
student2

I eksempelet er *student1* et entydig brukernavn og *obl1student1.txt* er navnet på filen der student1s besvarelse ligger. s*tudent2* i eksempelet har ikke levert noen besvarelse på oblig 1.

Hver besvarelse skal vurderes av en retter for emnet. Når fristen for en obligatorisk oppgave er utløpt, fordeles besvarelser på retterne, som vurderer en og en til godkjent (1) eller underkjent (0).

Piloten har funksjonalitet for hente ut en eksamensliste med brukernavn på alle studenter som er kvalifisert for eksamen – dvs har godkjent alle registrerte obliger i emnet.

Oppgave 4a) 5 poeng

Definer en klasse **Emne** (inkludert konstruktør) som administrerer obliger, rettere og studenter i et emne. Konstruktørens parametere gir verdier til emnekode (streng), registrerte studenter (ordbok med Student-objekter) og rettere (liste med Retter-objekter).

Oppgave 4b) 5 poeng

Skriv metoden **administrer** i klassen Emne. Metoden skriver ut emnekoden for emnet og en meny på terminalen før den ber om kommando. Den tar imot følgende lovlige kommandoer fra en bruker:
O: Ny oblig
F: Frist ute, start retting
L: Lag eksamensliste
A: Avslutt

Annen input skal gi feilmelding og nytt forsøk. En kommando bør gjenkjennes selv om det er blanke foran eller bak, og uansett om brukeren skriver liten eller stor bokstav (metodene **strip** og **upper** er nyttige her). For å utføre kommandoene skal metoden kalle på følgende non-public metoder som også ligger i klassen **Emne**:
O -> **_opprettOblig**
F -> **_startRetting**
L -> **_skrivEksamensListe**
Disse metodene skal skrives i senere deloppgaver. De opererer kun på instansvariabler i klassen, og har ingen parametere (annet enn **self**) eller returverdier.

Oppgave 4c) 10 poeng

Skriv en klasse **Student** med instansvariabler for brukernavn og fullt navn (begge får verdi fra parametere til konstruktøren) og en ordbok som skal ta vare på resultater av rettinger, der obligId er nøkkel og resultatet verdien. Klassen har følgende metoder i tillegg til konstruktør:

registrer med parametere **oblig** (en streng som identifiserer en oblig entydig) og **resultat** (et heltall som angir om en besvarelse er vurdert som godkjent). Metoden registrerer hvilket resultat en student har fått på en oblig.

altGodkjent med parameter **antObliger** som angir antall obliger registrert i emnet. Metoden returnerer en boolsk verdi: **True** hvis studenten har fått alle obliger godkjent, **False** om en eller flere obliger mangler eller har et annet resultat enn godkjent (kodet som heltallet 1).

Oppgave 4d) 2 poeng

Skriv klassen **Retter** med instansvariabel retterens brukernavn (parameter til konstruktøren) og metoden:

vurder med en parameter som angir et filnavn for en besvarelse. I denne foreløpige utgaven av piloten ignorerer programmet ditt filen med besvarelsen, og returnerer alltid heltallet 1 (godkjent).

Oppgave 4e) 8 poeng

Skriv en klasse **Oblig** med tre instansvariabler: Entydig id for obligen, en leveringsfrist på formen ååmmdd (f. eks. 190906 for 6. september 2019) og om obligen er rettet eller ikke. De to første initialiseres med parametere til konstruktøren. Skriv følgende metoder i klassen Oblig:

klarForRetting med parameter som angir dagens dato. Metoden returnerer **True** hvis fristen er før dagens dato og besvarelsene for obligen ikke allerede er rettet – ellers **False**.

hentBesvarelser leser en fil med oversikt over studenters besvarelser (navn og format på denne filen er beskrevet i oppgave-innledningen) og returnerer en ordbok der en students brukernavn er nøkkel og filnavn med studentens besvarelse er verdi.

Oppgave 4f) 5 poeng

Utvid klassen **Oblig** med en ny metode

fordelRetting tar som parametere en ordbok med studenters besvarelser og en liste med rettere, og lar retterne vurdere én og én besvarelse (bruk metoden **vurder** i klassen **Retter**) slik at alle retterne ideelt sett vurderer like mange. Altså skal ingen retter vurdere mer enn 1 besvarelse mer enn de andre. Resultatene av retternes vurderinger samles i en ordbok der nøkkelverdiene er studentenes brukernavn. Metoden markerer obligen som rettet og returnerer ordboken med resultater.

Oppgave 4g) 15 poeng

Skriv følgende non-public metoder i klassen **Emne**:

_opprettOblig genererer et unikt oblignavn, og ber bruker på terminalen om å oppgi frist på formen ååmmdd. Deretter legges en ny oblig til emnet.

_startRetting ber om dagens dato og sjekker om noen av obligene i emnet har en utløpt frist uten at besvarelsene er rettet. I så fall henter den alle besvarelser for obligen, fordeler dem på rettere og registrerer resultatet av rettingen i Student-objekter (på riktig oblig) for studenter som har levert.

_skrivEksamensListe bygger opp en liste med brukernavn for alle studenter som har fått godkjent alle obliger registrert for emnet, og skriver til sist ut denne på terminal med en passende overskrift.