

□ Forside

Skriftlig hjemmeeksamen i IN2000 2021 VÅR

Varighet: 1. juni, kl. 09:00 til 13:00 (+ 30 min)

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 semestersiden 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 Inspira 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 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 vil bli betraktet som forsøk på fusk.

Fusk:

Les om [hva som regnes som fusk på UiOs nettsider](#).

Digital håndtegning/filopplasting:

Du har fått 30 min. ekstratid til opplasting av filer (f.eks. digital håndtegning). [Sjekk hvordan du leverer inn digital håndtegning](#)

Kontaktinfo:

[Brukerstøtte eksamen](#)

Om poeng:

Det er mulig å oppnå totalt 100 poeng på denne eksamen. Poengsummen vil bli delt på 2 og lagt sammen med poengsummen på prosjektet før endelig karakter blir satt.

1 Oppgave 1

Hvilke av følgende krav er ikke-funksjonelle?

(Poeng: 2 for hver riktig, -1 for hver feil, 0 for ingen valgt)

Velg ett eller flere alternativer

- Alle byer over en viss størrelse skal vises på kartet
- Systemet skal spille av en lyd ved innlogging
- Systemet skal støtte innlogging ved bruk BankId
- Systemet skal ikke ha en nedetid på over 1%
- Systemet skal gi deg GPS koordinatene der du er
- Systemet skal være svært vanskelig å hacke

Maks poeng: 4

2 Oppgave 2

1. Prosessmodell

En bedrift med mange utviklingsteam har byttet fra prosessmodellen Kanban til Scrum. De aller fleste teamene er fornøyd med overgangen til Scrum, men ett av teamene er ikke fornøyd og uttrykker blant annet:

"Vi likte å ha et tak på antall oppgaver som skulle gjøres i parallell. Nå blir det innført tidsintervaller og mer tidspress."

Overbevis teamet om at Scrum kan være fornuftig å bruke. (6p)

2. Ikke-funksjonelle krav











Et typisk ikke-funksjonelt krav til et system er brukskvalitet. Slike krav blir ofte uttrykt svært generelt, noe som gjør det vanskelig for utviklingsteamet å teste om kravene oppfylles.

For eksempel:

"Systemet skal være enkelt å bruke for alle typer brukere og organisert slik at brukerfeil forekommer minst mulig."

Beskriv dette kravet på en annen måte som gjør det mer testbart. (6p)

Skriv ditt svar her

Format | **B** | *I* | U | x_2 | x^2 | I_x |  |  |  |  |  |  |  |  |  |  |

Words: 0

Maks poeng: 12

□ Casebeskrivelse

Badstueforeningen **Spa til Sjø** har nylig etablert seg langs Oslofjorden, bl.a. i Bjørvika og på Tjuvholmen. **Spa til Sjø** tilbyr utleie av badstuer som det også er mulig å bade fra. **Spa til Sjø** har en egen app der brukere kan booke time-slots på 3 timer. Åpningstider og antall timeslots som det er mulig å booke i løpet av en dag er spesifisert her:

Timeslotid	Klokkeslett
1	09:00 - 12:00
2	12:00 - 15:00
3	15:00 - 18:00
4	18:00 - 21:00

En bruker kan maksimum booke til 12 personer, inkludert seg selv. Dersom en bruker bestiller for 6 personer eller mindre, må en spesifisert fastpris (se under) uansett betales. For hver person ekstra (over 6) vil en engangskostnad legges til. Det er mulig å velge mellom to badstuer: den eksklusive "Selen" og den rimeligere "Sjøpølsa".

Priser er spesifisert her:

Beskrivelse	Selen	Sjøpølsa
Fastpris opptil 6 personer	kr 3000,-	kr 1800,-
Tillegg pr. person fra 7 til 12 personer	kr 200,-	kr 150,-

3 Oppgave 3

Denne oppgaven (maks 50 poeng) består av seks deloppgaver. Les case-beskrivelsen på foregående side eller i vedlagt PDF før du svarer. Alle deloppgavene (a til f, inkludert diagrammer) skal besvares i én PDF-fil som lastes opp under.

a) Oppgi interessenter med deres interesse for systemet. Angi hvilke interessenter som er aktører og om de er primære eller sekundære. (6p)

Du skal videre ta for deg use-caset "book badstue". Redegjør for eventuelle antagelser i besvarelsen for de følgende oppgavene.

b) Lag et aktivitetsdiagram for use-caset "book badstue". (8p)

c) Lag et sekvensdiagram for use-caset "book badstue". Modeller hovedflyt og en alternativ flyt. (10p)

d) Lag et klassediagram som reflekterer sekvensdiagrammet laget i oppgave c) Husk å få med assosiasjoner og tilhørende multiplisitet mellom klassene. (10p)

e) Foreslå to ulike forskningsmetoder som kan anvendes for å måle brukeropplevelsen (både selve bookingsystemet og selve "badstue-opplevelsen"). Redegjør for fordeler og ulemper ved hver av disse metodene, samt hvordan de i praksis kan gjennomføres. (8p)

f) Beskriv to misuse-case (trusselscenarioer) som **Spa til Sjø** bør ta hensyn til og for hvert av trusselscenarioene, foreslå tiltak for å forhindre dem. (8p)



Last opp filen her. Maks én fil.

Alle filtyper er tillatt. Maksimal filstørrelse er **2 GB**.

Maks poeng: 50

4 Oppgave 4

Spa til Sjøen har erfart at det er færre bookinger når det er dårlig vær, og at de er overbooket når det er fint vær. Derfor ønsker Spa til Sjøen å differensiere prisene for å få best mulig utnyttelse av kapasiteten. Reglene for rabatt er basert på værforhold og er som følger:

	Temp under 20 °C	Vind over 10m/s	Regn
Rabatt	20%	10%	15%

Rabatter akkumulerer, som vil si at dersom flere værforhold inntreffer samtidig vil rabattene legges sammen. Dersom to eller flere av værforholdene som er beskrevet i tabellen over forekommer, skal en ekstra rabatt på 5% gis.

a) Fullfør beslutningstabellen (decision table) nedenfor med alle kombinasjoner av rabatter. Fyll inn tall i raden "Rabatt" som representerer rabatt i prosent. (8p)

Regn	T	T	T	T	F	F	F	F
Vind over 10m/s	T	T	F	F	T	T	F	F
Temp under 20 °C	T	F	T	F	T	F	T	F
Rabatt	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

(T = True, F = False)

b) Anta at det skal fjernes et av værforholdene (betingelse) som gir rabatt. Hvor mange ulike kombinasjoner er det mulig å oppnå totalt? (1p)

Maks poeng: 9











5 Oppgave 5

Studer vedlagt PDF-dokument og besvar spørsmålene:

a) Forklar i korte trekk hva koden gjør. (5p)

b) Foreslå forbedringer til koden. Spesifiser linjenummer og hva som kan forbedres. (5p)

Skriv ditt svar her

Format | **B** | *I* | U | x_2 | x^2 | I_x |  |  |  |  |  |  |  |  |  |  |

Words: 0

Maks poeng: 10

6 Oppgave 6

```
data class TimeSlot(val id : Int, val timespan : String)

val timeslots = mutableListOf<TimeSlot>()
timeslots.add(TimeSlot(1, "09:00-12:00"))
timeslots.add(TimeSlot(2, "12:00-15:00"))
timeslots.add(TimeSlot(3, "15:00-18:00"))
timeslots.add(TimeSlot(4, "18:00-21:00"))
```

Gitt følgende datastruktur for TimeSlot, og en liste kalt "timeslots" opprettes med følgende innhold (se over).

'timeslots' er en liste som inneholder fire TimeSlot-objekter som hver representerer de fire timeslotene som kan bookes hos **Spa til Sjø**s.

Anta at weatherList er fylt med Weather-objekter etter API-kallet i forrige oppgave (se vedlegg).

Oppgaven:

Implementer funksjonen **highestTemp()** i Kotlin som skal returnere det TimeSlot-objektet som har den **høyeste** temperaturen basert på Weather-objektene i weatherList.

funksjonen skal ta to argumenter:

- timeslots (vist øverst) som er en liste som inneholder fire TimeSlot-objekter.
- weatherList (fra forrige oppgave, se vedlegg), en liste som inneholder Weather-objekter.

```
val result = highestTemp(timeslots, weatherList)
```

Funksjonen skal bl.a. kunne kalles på slik:

Og 'result' skal da settes lik TimeSlot-objektet som har **høyest** temperatur basert på Weather-objekter i weatherList.

Implementer funksjonen her:

(NB: Kotlin-syntaks støttes dessverre ikke i Inspira)

1

Maks poeng: 15

Question 10
Attached



Gitt følgende kode:

```
9 class MainActivity : AppCompatActivity() {
10     override fun onCreate(savedInstanceState: Bundle?) {
11         super.onCreate(savedInstanceState)
12         setContentView(R.layout.activity_main)
13
14         // a container for Weather objects for a specific day
15         val weatherList = mutableListOf<Weather>()
16
17         // API-query with "starttime" attribute appended
18         val path = "https://spatilsjoesAPI.no/hourlyCasts/starttime="
19
20         for (i in 0..23) {
21             runBlocking { this: CoroutineScope
22                 val response : String = Fuel.get( path: path + i).awaitString()
23
24                 val from = response.split( ...delimiters: "from")[1].substring(4, 6)
25                 val to = response.split( ...delimiters: "to")[1].substring(4, 6)
26                 val temp = response.split( ...delimiters: "temp")[1].substring(4, 7)
27                 val rain = response.split( ...delimiters: "precipitation")[1].substring(4, 7)
28
29                 weatherList.add(Weather(Time(from, to), temp, rain))
30             }
31         }
32     }
33 }
34
35 data class Weather(val time : Time?, val temp : String?, val rain : String)
36 data class Time(val from : String?, val to : String?)
```

Og følgende endepunkt der du kan spesifisere "starttime" mellom 0 og 23 (eksempelet viser med 10):

```
https://spatilsjoesAPI.no/hourlyCasts/starttime=10
```

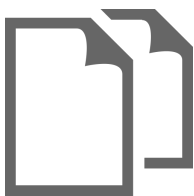
Gir følgende respons:

```
{
  "time": {
    "from": "10:00",
    "to": "11:00"
  },
  "temp": "20.5",
  "precipitation": "2.8",
  "wind": {
    "direction": "NW",
    "speed": "3.5"
  }
}
```

Forklaringer:

starttime	Verdien "starttime" representerer et klokkeslett og endepunktet gir værvarel for en time fra den verdien som er satt. Som eksempelet over viser vil "starttime=10" gi værvarelet mellom kl. 10 og kl. 11. Du kan anta at lokasjonen for værvarel er der Spa til Sjø -badstuene befinner seg.
temp	Temperatur i Celsius
precipitation	Nedbør i mm
'speed' under 'wind'	Vindhastighet i m/s (meter per sekund)

Question 5
Attached



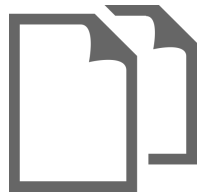
Badstueforeningen **Spa til Sjø** har nylig etablert seg langs Oslofjorden, bl.a. i Bjørvika og på Tjuvholmen. **Spa til Sjø** tilbyr utleie av badstuer som det også er mulig å bade fra. **Spa til Sjø** har en egen app der brukere kan booke time-slots på 3 timer. Åpningstider og antall timeslots som det er mulig å booke i løpet av en dag er spesifisert her:

Timeslotid	Klokkeslett
1	09:00 – 12:00
2	12:00 – 15:00
3	15:00 – 18:00
4	18:00 – 21:00

En bruker kan maksimum booke til 12 personer, inkludert seg selv. Dersom en bruker bestiller for 6 personer eller mindre, må en spesifisert fastpris (se under) uansett betales. For hver person ekstra (over 6) vil en engangskostnad legges til. Det er mulig å velge mellom to badstuer: den eksklusive “Selen” og den rimeligere “Sjøpølsa”.

Beskrivelse	Selen	Sjøpølsa
Fastpris opptil 6 personer	Kr 3000,-	Kr 1800,-
Tillegg pr. person fra 7 til 12 personer	Kr 200,-	Kr 150,-

Question 8
Attached



Gitt følgende kode:

```
9 class MainActivity : AppCompatActivity() {
10     override fun onCreate(savedInstanceState: Bundle?) {
11         super.onCreate(savedInstanceState)
12         setContentView(R.layout.activity_main)
13
14         // a container for Weather objects for a specific day
15         val weatherList = mutableListOf<Weather>()
16
17         // API-query with "starttime" attribute appended
18         val path = "https://spatilsjoesAPI.no/hourlyCasts/starttime="
19
20         for (i in 0..23) {
21             runBlocking { this: CoroutineScope
22                 val response : String = Fuel.get( path: path + i).awaitString()
23
24                 val from = response.split( ...delimiters: "from")[1].substring(4, 6)
25                 val to = response.split( ...delimiters: "to")[1].substring(4, 6)
26                 val temp = response.split( ...delimiters: "temp")[1].substring(4, 7)
27                 val rain = response.split( ...delimiters: "precipitation")[1].substring(4, 7)
28
29                 weatherList.add(Weather(Time(from, to), temp, rain))
30             }
31         }
32     }
33 }
34
35 data class Weather(val time : Time?, val temp : String?, val rain : String)
36 data class Time(val from : String?, val to : String?)
```

Og følgende endepunkt der du kan spesifisere "starttime" mellom 0 og 23 (eksempelet viser med 10):

```
https://spatilsjoesAPI.no/hourlyCasts/starttime=10
```

Gir følgende respons:

```
{
  "time": {
    "from": "10:00",
    "to": "11:00"
  },
  "temp": "20.5",
  "precipitation": "2.8",
  "wind": {
    "direction": "NW",
    "speed": "3.5"
  }
}
```


Forklaringer:

starttime	Verdien "starttime" representerer et klokkeslett og endepunktet gir værvarel for en time fra den verdien som er satt. Som eksempelet over viser vil "starttime=10" gi værvarelet mellom kl. 10 og kl. 11. Du kan anta at lokasjonen for værvarel er der Spa til Sjø -badstuene befinner seg.
temp	Temperatur i Celsius
precipitation	Nedbør i mm
'speed' under 'wind'	Vindhastighet i m/s (meter per sekund)

Question 9
Attached



Gitt følgende kode:

```
9 class MainActivity : AppCompatActivity() {
10     override fun onCreate(savedInstanceState: Bundle?) {
11         super.onCreate(savedInstanceState)
12         setContentView(R.layout.activity_main)
13
14         // a container for Weather objects for a specific day
15         val weatherList = mutableListOf<Weather>()
16
17         // API-query with "starttime" attribute appended
18         val path = "https://spatilsjoesAPI.no/hourlyCasts/starttime="
19
20         for (i in 0..23) {
21             runBlocking { this: CoroutineScope
22                 val response : String = Fuel.get( path: path + i).awaitString()
23
24                 val from = response.split( ...delimiters: "from")[1].substring(4, 6)
25                 val to = response.split( ...delimiters: "to")[1].substring(4, 6)
26                 val temp = response.split( ...delimiters: "temp")[1].substring(4, 7)
27                 val rain = response.split( ...delimiters: "precipitation")[1].substring(4, 7)
28
29                 weatherList.add(Weather(Time(from, to), temp, rain))
30             }
31         }
32     }
33 }
34
35 data class Weather(val time : Time?, val temp : String?, val rain : String)
36 data class Time(val from : String?, val to : String?)
```

Og følgende endepunkt der du kan spesifisere "starttime" mellom 0 og 23 (eksempelet viser med 10):

```
https://spatilsjoesAPI.no/hourlyCasts/starttime=10
```

Gir følgende respons:

```
{
  "time": {
    "from": "10:00",
    "to": "11:00"
  },
  "temp": "20.5",
  "precipitation": "2.8",
  "wind": {
    "direction": "NW",
    "speed": "3.5"
  }
}
```

Forklaringer:

starttime	Verdien "starttime" representerer et klokkeslett og endepunktet gir værvarel for en time fra den verdien som er satt. Som eksempelet over viser vil "starttime=10" gi værvarelet mellom kl. 10 og kl. 11. Du kan anta at lokasjonen for værvarel er der Spa til Sjø -badstuene befinner seg.
temp	Temperatur i Celsius
precipitation	Nedbør i mm
'speed' under 'wind'	Vindhastighet i m/s (meter per sekund)