

i **Forside**

Institutt for datateknologi og informatikk

# IDAT 1001 Programmering 1

## Prøve Eksamen høst 2019

-----  
-----

**Faglig kontakt under eksamen:**

*Trondheim IDATT*  
Bjørn Klefstad (tlf 73559572)  
Grethe Sandstrak (tlf. 73559561)

*Gjøvik IDATG*  
Kiran Bylappa Raja (tlf 61135374)

*Ålesund IDATA*  
Arne Styve (tlf 70161287)

**Administrativ kontakt under eksamen:**

Ingrid Island (tlf. 73559564)

**Eksamensdato:** xx. xx.xxxx  
**Eksamenstid (fra-til):** 4 timer

**Tillatte hjelpemidler:**

Tilgang til sitt vante programmeringsmiljø, modelleringsverktøy og Java API.  
**FRA ÅLESUND TIDL EKSAMEN:** Alle hjelpemidler tillatt (lærebok, egne notater, internett), med **UNNTAK** av verktøy/løsninger for samhandling (chat, meldingstjenester, distribuert/sentralisert versjonskontroll, diskusjonsfora og lignende).

**Annen informasjon:**

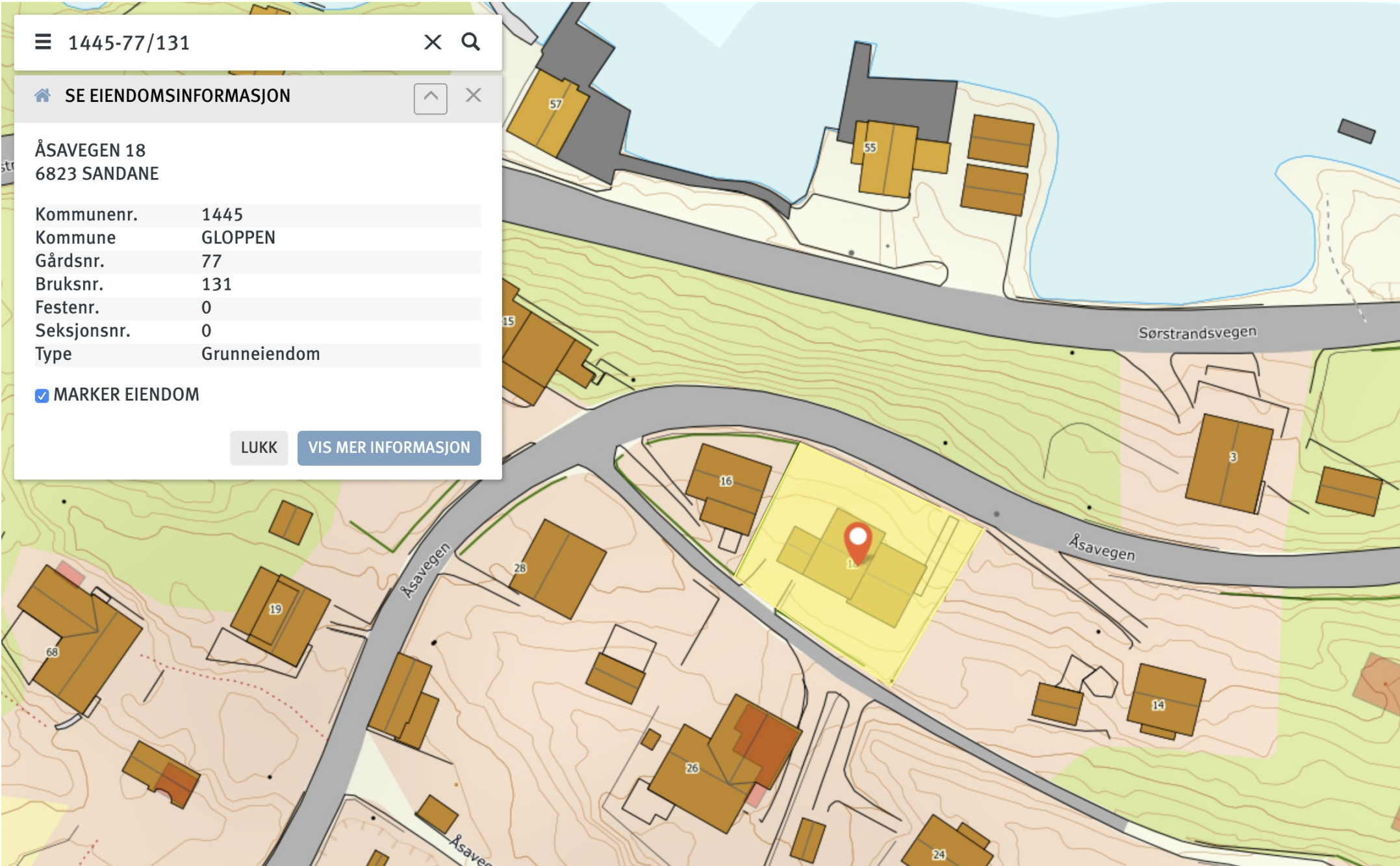
Les gjennom hele oppgavesettet før du begynner arbeidet, og disponer tiden. Dersom noe virker uklart i oppgavesettet, skal du gjøre dine egne antagelser og forklare dette i besvarelsen.  
Eksamen utføres på egen medbragt datamaskin (BYOD). Datamaskinen skal ha nødvendige verktøy installert på forhånd (BlueJ, Netbeans, Eclipse el.l.)  
Alle oppgavene skal besvares digitalt i Inspira, enten ved å svare direkte på spørsmål, eller ved å laste opp fil som vedlegg til spørsmål.  
Programmeringsoppgaver gjøres i en IDE (BlueJ, Netbeans, IntelliJ, Eclipse etc) og lastes opp samlet i siste oppgave (Innlevering).

Lykke til!

# Problembeskrivelse

## Kommunalt tomteregister

Gloppen kommune skal utvikle et nytt system for å holde oversikt over alle eiendommer i kommunen. De skal også kunne utføre noen enkle analyser av den samlede eiendomsmassen i kommunen.



## Kravspesifikasjon

Applikasjonen skal ha følgende funksjonalitet:

- Registrere/legge inn eiendom
- Skrive ut alle eiendommer som er registrert
- Søke etter eiendom basert på kommunenr, gnr og bnr (alle 3 samtidig)
- Regne ut og vise gjennomsnitts areal av alle eiendommene i registeret

Alle eiendommer (engelsk: *real estate, lot or property*) i Norge er registrert med blant annet følgende informasjon:

- Kommunenummer - et tall fra og med 101 (Halden) til og med 5054 (Indre Fosen) (eng: *municipality number*)
- Kommunnavn (eng: *municipality name*)
- Gårdsnummer - ofte forkortet **gnr**. Et positivt heltall. (eng: *lot number*)
- Bruksnummer - ofte forkortet **bnr**. Et positivt heltall. (eng: *section number*)
- Bruksnavn - et navn på tomta. (eng: *name*). Ikke alle eiendommer har navn.
- Areal i m<sup>2</sup>. Et positivt desimaltall (eng: *area*)
- Navn på eier (eng: *Name of owner*) - kan lagres som en tekststreng

Ved utskrift, skrives kommunenummer, gnr og bnr på følgende format:

*kommunenr-gnr/bnr*. (f.eks. 1504-54/73).

Applikasjonen din skal ha et **tekstbasert brukergrensesnitt** i form av en meny med ulike menyvalg.

## Test-data

For å teste applikasjonen din, kan du benytte følgende data over 5 eiendommer i Gloppen kommune (eierne er fiktive navn):

Kommune navn	Kommune nr	Gårdsnr	Bruksnr	Bruksnavn	Areal (m2)	Eier
Gloppen	1445	77	631		1017,6	Jens Olsen
Gloppen	1445	77	131	Syningom	661,3	Nicolay Madsen
Gloppen	1445	75	19	Fugletun	650,6	Evilyn Jensen
Gloppen	1445	74	188		1457,2	Karl Ove Bråten
Gloppen	1445	69	47	Høiberg	1339,4	Elsa Indregård

## Prosjektfiler

For å løse oppgaven, kan du benytte hvilken som helst IDE (utviklingsmiljø) (Netbeans, BlueJ, IntelliJ, Eclipse eller andre).

Opprett et nytt prosjekt i din IDE i en mappe på din harddisk. Pass på at du har full kontroll over hvor på din harddisk prosjektet ditt blir lagret.

Når du er ferdig med prosjektet, skal du **lage en ZIP-fil (IKKE rar eller andre formater!)** av **hele prosjektmappen din** og laste opp ZIP-filen som svar i siste oppgave. Husk å sette av godt med tid mot slutten av eksamen til å pakke og levere ZIP-filen (sist 20 minutter av eksamenstiden). Eksamen stenger presis, og det vil ikke være mulig å laste opp ZIP-filen din etter at eksamen er stengt.

## Vurderingskriterier

Din besvarelse vil bli vurdert basert på følgende kriterier:

- Om du har fulgt (og holdt deg til) kravspesifikasjonen
- Om du har fulgt prinsippene for god design (coupling, cohesion, responsibility driven design osv)
- Om du har valgt gode, beskrivende navn på klasser, metoder og variabler/felt/parametre
- Om du har dokumentert koden din godt (Javadoc)
- Om du har god og fornuftig samhandling med bruker (god brukervennlighet i form av klare meldinger og god info osv)

I de påfølgende deloppgavene skal du gradvis utvikle applikasjonen som skal brukes for å håndtere register av eiendommer.

## Oppgave 1 - Modellering

Les problembeskrivelse og **hele** oppgavesettet og lag ett klassediagram for alle klassene du mener bør utvikles (tegn på utdelt papir).  
Vis i klassediagrammet hvordan du mener de ulike klassene bør samarbeide med hverandre. Begrunn svaret ditt i tekstfeltet nedenfor.

Med unntak av begrunnelsen løses oppgave 1 på eget ark (penn og papir). Arket blir samlet inn og scannet av eksamensvaktene på slutten av eksamen.

**Skriv ditt svar her...**

Format

**B**

*I*

U

$x_2$

$x^2$

$I_x$

Words: 0

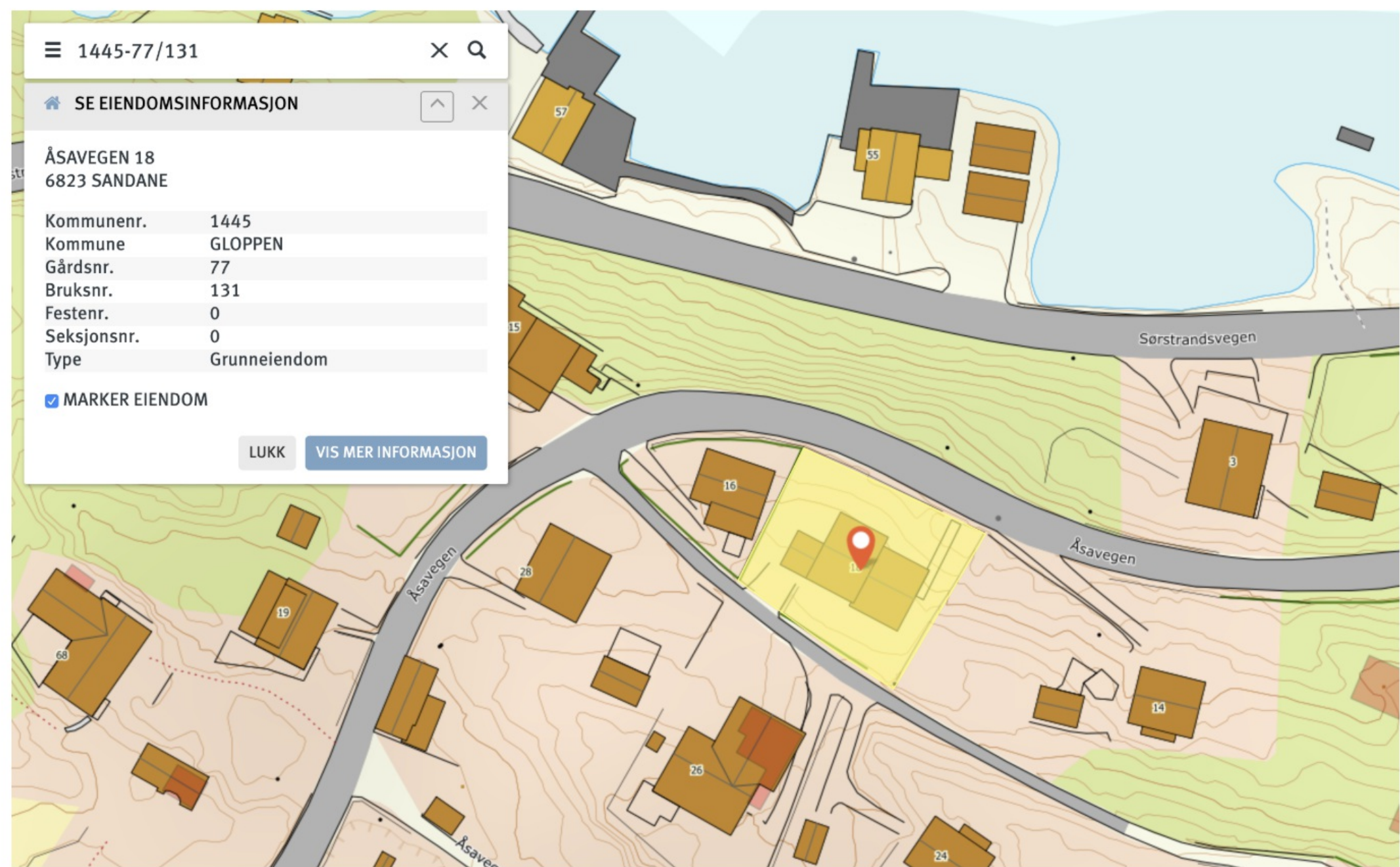
Maks poeng: 10



# Problembeskrivelse

## Kommunalt tomteregister

Gloppen kommune skal utvikle et nytt system for å holde oversikt over alle eiendommer i kommunen. De skal også kunne utføre noen enkle analyser av den samlede eiendomsmassen i kommunen.



## Kravspesifikasjon

Applikasjonen skal ha følgende funksjonalitet:

- Registrere/legge inn eiendom
- Skrive ut alle eiendommer som er registrert
- Søke etter eiendom basert på kommunenr, gnr og bnr (alle 3 samtidig)
- Regne ut og vise gjennomsnitts areal av alle eiendommene i registeret

Alle eiendommer (engelsk: *real estate, lot or property*) i Norge er registrert med blant annet følgende informasjon:

- Kommunenummer - et tall fra og med 101 (Halden) til og med 5054 (Indre Fosen) (eng: *municipality number*)
- Kommunenavn (eng: *municipality name*)
- Gårdsnummer - ofte forkortet **gnr**. Et positivt heltall. (eng: *lot number*)
- Bruksnummer - ofte forkortet **bnr**. Et positivt heltall. (eng: *section number*)
- Bruksnavn - et navn på tomta. (eng: *name*). Ikke alle eiendommer har navn.
- Areal i m<sup>2</sup>. Et positivt desimaltall (eng: *area*)
- Navn på eier (eng: *Name of owner*) - kan lagres som en tekststreng

Ved utskrift, skrives kommunenummer, gnr og bnr på følgende format:

*kommunenr-gnr/bnr.* (f.eks. 1504-54/73).

Applikasjonen din skal ha et **tekstbasert brukergrensesnitt** i form av en meny med ulike menyvalg.

## Test-data

For å teste applikasjonen din, kan du benytte følgende data over 5 eiendommer i Gloppen kommune (eierne er fiktive navn):

Kommune navn	Kommune nr	Gårdsnr	Bruksnr	Bruksnavn	Areal (m2)	Eier
Gloppen	1445	77	631		1017,6	Jens Olsen
Gloppen	1445	77	131	Syningom	661,3	Nicolay Madsen
Gloppen	1445	75	19	Fugletun	650,6	Evilyn Jensen
Gloppen	1445	74	188		1457,2	Karl Ove Bråten
Gloppen	1445	69	47	Høiberg	1339,4	Elsa Indregård

## Prosjektfiler

For å løse oppgaven, kan du benytte hvilken som helst IDE (utviklingsmiljø) (Netbeans, BlueJ, IntelliJ, Eclipse eller andre).

Opprett et nytt prosjekt i din IDE i en mappe på din harddisk. Pass på at du har full kontroll over hvor på din harddisk prosjektet ditt blir lagret.

Når du er ferdig med prosjektet, skal du **lage en ZIP-fil (IKKE rar eller andre formater!)** av **hele prosjektmappen din** og laste opp ZIP-filen som svar i siste oppgave. Husk å sette av godt med tid mot slutten av eksamen til å pakke og levere ZIP-filen (sist 20 minutter av eksamenstiden). Eksamen stenger presis, og det vil ikke være mulig å laste opp ZIP-filen din etter at eksamen er stengt.

## Vurderingskriterier

Din besvarelse vil bli vurdert basert på følgende kriterier:

- Om du har fulgt (og holdt deg til) kravspesifikasjonen
- Om du har fulgt prinsippene for god design (coupling, cohesion, responsibility driven design osv)
- Om du har valgt gode, beskrivende navn på klasser, metoder og variabler/felt/parametre
- Om du har dokumentert koden din godt (Javadoc)
- Om du har god og fornuftig samhandling med bruker (god brukervennlighet i form av klare meldinger og god info osv)

I de påfølgende deloppgavene skal du gradvis utvikle applikasjonen som skal brukes for å håndtere register av eiendommer.

(a) **Oppgave 2 a)**

Implementer/lag en klasse som representerer en eiendom:

- 1. Gi klassen et fornuftig og forklarende navn.
- 2. Definere fornuftige felt av relevante datatyper.
- 3. Implementer konstruktøren/konstruktører med nødvendige parametre
- 4. Implementer aksessor-metoder for klassen.
- 5. Implementer mutator-metoder for klassen (de du mener er nødvendige/nyttige). Gi en kort begrunnelse for hvorfor du har valgt å implementere mutator-metodene (skriv begrunnelsen som del av klassedokumentasjonen).
- 6. Dokumenter koden

Kopier kildekoden for klassen fra ditt IDE til svarfeltet under:

**Skriv ditt svar her...**

1	
---	--

Maks poeng: 19

(b) **Oppgave 2 b)**

Kommunenummer sammen med gards- og bruksnummer utgjør en unik ID av en eiendom.  
Lag en metode i eiendom-klassen som returnerer en tekststreng av disse 3 numrene på følgende format:

*kommunenr-gnr/bnr*

M.a.o. for en tomt i Gloppen kommune (1445) med gardsnummer 54 og bruksnummer 73, skal metoden returnere strengen:

*"1445-54/73"*

Lim inn koden for metoden din i svarfeltet under. Du velger selv navn på metoden.

**Skriv ditt svar her...**

1	
---	--

Maks poeng: 6



3(a) **Oppgave 3 a)**

Opprett en klasse som representerer selve eiendoms-registeret i din applikasjon.  
Begrunn valg av hvordan du vil håndtere en samling (tabell) av eiendommer. M.a.o. hvorfor du f.eks. valgte å bruke en *fixed size array eller* ArrayList osv.  
Du kan skrive begrunnelsen som del av dokumentasjonen til klassen.

Legg til metoder for:

- å registrere en eiendom i registeret
- å slette en eiendom fra registeret
- å returnere antall eiendommer i registeret
- å finne en eiendom basert på kommune nummer, gardsnummer (gnr) og bruksnummer.

Lim inn hele klassen i svarfeltet under når du har kodet ferdig oppgaven.  
**Skriv ditt svar her...**

1

Maks poeng: 13

3(b) **Oppgave 3 b)**

I eiendomsregister-klassen, legg til en metode som regner ut gjennomsnittsareal av alle eiendommene i registeret.

Du velger selv retur-type og navn på metoden.

Lim inn hele metoden (med dokumentasjon) i svarfeltet under.

**Skriv ditt svar her...**

1	
---	--

Maks poeng: 4

3(c) **Oppgave 3 c)**

Lag en metode som søker opp og returnerer **samtlig**e eiendommer med et gitt gårdsnummer. Metoden skal f.eks. brukes til å finne alle eiendommer med gårdsnummer lik 77.  
Du velger selv hvilken datatype metoden skal returnere, og navnet på metoden.

Lim inn koden til hele metoden (inkludert dokumentasjon) i feltet under.

**Skriv ditt svar her...**

1	
---	--

Maks poeng: 10

## Oppgave 4 - Brukergrensesnitt

Forslag til metoder for å håndtere en tekstbasert meny:

```
/**
 * Presents the menu for the user, and awaits input from the user. The menu
 * choice selected by the user is being returned.
 *
 * @return the menu choice by the user as a positive number starting from 1.
 *         If 0 is returned, the user has entered a wrong value
 */
private int showMenu()
{
    int menuChoice = 0;
    System.out.println("\n***** Property Register Application v0.1 *****\n");
    System.out.println("1. Add property");
    System.out.println("2. List all properties");
    System.out.println("3. Search property");
    System.out.println("4. Calculate average area");
    //TODO: Add more menus
    System.out.println("9. Quit");
    System.out.println("\nPlease enter a number between 1 and 9.\n");
    Scanner sc = new Scanner(System.in);
    if (sc.hasNextInt()) {
        menuChoice = sc.nextInt();
    } else {
        System.out.println("You must enter a number, not text");
    }
    return menuChoice;
}
```

```
/**
 * Starts the application. This is the main loop of the application,
 * presenting the menu, retrieving the selected menu choice from the user,
 * and executing the selected functionality.
 */
public void start() {
    boolean finished = false;
    // The while-loop will run as long as the user has not selected
    // to quit the application
    while (!finished) {
        int menuChoice = this.showMenu();
        switch (menuChoice)
        {
            case ADD_PROPERTY:
                //TODO: Fill inn your code here....
                break;

            case LIST_ALL_PROPERTIES:
                //TODO: Fill inn your code here....
                break;

            case FIND_PROPERTY:
                //TODO: Fill inn your code here....
                break;

            case CALCULATE_AVERAGE_AREA:
                //TODO: Fill inn your code here....
                break;

            case EXIT:
                System.out.println("Thank you for using the Properties app!\n");
                finished = true;
        }
    }
}
```



```
        break;

        default:
            System.out.println("Unrecognized menu selected..");
            break;
    }
}
```

... der følgende konstanter er definert:

```
// Constants representing the different menu choices
private final int ADD_PROPERTY = 1;
private final int LIST_ALL_PROPERTIES = 2;
private final int FIND_PROPERTY = 3;
private final int CALCULATE_AVERAGE_AREA = 4;
private final int EXIT = 9;
```

(a) **Oppgave 4 a) - Klientprogram**

I denne oppgaven skal du lage et enkelt tekstbasert brukergrensesnitt på applikasjonen din. Lag en ny klasse som skal representere og ha ansvaret for brukerinteraksjonen. I teksten til høyre finner du forslag til kode for å vise en meny og hente inn menyvalg fra bruker (*int showMenu()*), og en metode som er hovedmetoden som håndterer valgt meny. Du trenger ikke å la din løsning være basert på denne eksempelkoden.

Fullfør koden for klassen med alle funksjonene fra kravspesifikasjonen:

- Registrere/legge inn eiendom
- Skrive ut alle eiendommer som er registrert
- Søke etter eiendom basert på kommunenr, gnr og bnr (alle 3 samtidig)
- Regne ut og vise gjennomsnitts areal av alle eiendommene i registeret

Når klassen er ferdig, kopierer du **hele klassen** fra din IDE og limer inn koden inn i svarfeltet under.

**Skriv ditt svar her...**

1

Maks poeng: 22

(b) **Oppgave 4 b)**

Tegn **sekvensdiagram** som viser hva som skjer når en bruker velger menyvalget "Skriv ut alle eiendommer i registeret".

Du tegner sekvensdiagrammet på utlevert papir. Husk å lever til eksamensvakt når du er ferdig med **hele** eksamen.

Maks poeng: 4



5    **Oppgave 5**




Med bakgrunn i din endelige løsning av kravspesifikasjonen, hvordan vil du si at designet og implementasjonen din er utført i henhold til design-prinsippene **kobling (eng: coupling)**, og **samstemthet (eng: cohesion)** ? Gi gjerne konkrete eksempler fra din egen kode.



Skriv ditt svar her...



Format



**B***I*U $\times_2$  $\times^2$  $I_x$












ABC



Words: 0

Maks poeng: 12


6    **Laste opp prosjekt**

Last opp **hele prosjektmappen din** som en **ZIP-fil** (IKKE rar, arc el.l.). Husk å lage ZIP-fil av **hele prosjektmappen din**, slik at sensor enkelt kan åpne prosjektet i samme IDE som du har benyttet.



Last opp ZIP-filen her. Maks én fil.

Følgende filtyper er tillatt: **.zip**    Maksimal filstørrelse er **35 GB**.

 Velg fil for opplasting

Maks poeng: 0