

i Forside

IDAT 1001 Programmering 1

Ordiner Eksamen høst 2020

Institutt: IE/IIR

Eksamensdato: 14.12.2020

Eksamenstid (fra-til): 09:00 - 13:00 (4 t)

Eksamensform: Skriftlig Eksamen

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)

Hjelpemiddelkode/Tillatte hjelpemidler:

A / Alle hjelpemidler tillatt (lærebok, egne notater).

ANNEN INFORMASJON:

Gjør dine egne antagelser og presiser i besvarelsen hvilke forutsetninger du har lagt til grunn i tolkning/avgrensing av oppgaven. Faglig kontaktperson skal kun kontaktes dersom det er direkte feil eller mangler i oppgavesettet.

Lagring: Besvarelsen din i Inspira Assessment lagres automatisk. Jobber du i andre programmer – husk å lagre underveis.

Juks/plagiat: Eksamen skal være et individuelt, selvstendig arbeid. Det er tillatt å bruke hjelpemidler. Alle besvarelser blir kontrollert for plagiat.

Varslinger: Hvis det oppstår behov for å gi beskjeder til kandidatene underveis i eksamen (f.eks. ved feil i oppgavesettet), vil dette bli gjort via varslinger i Inspira. Et varsel vil dukke opp som en dialogboks på skjermen i Inspira. Du kan finne igjen varselet ved å klikke på bjella øverst i høyre hjørne på skjermen.

Filopplasting: Alle filer må være lastet opp i besvarelsen før eksamenstida går ut. Det er lagt til 15 minutter til ordinær eksamenstid for digitalisering av håndtegninger/filer. (Tilleggstida inngår i gjenstående eksamenstid som vises øverst til venstre på skjermen.)

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). Der det i oppgavene blir bedt om det, kopierer du kode fra IDE til svarfeltet.

OM LEVERING:

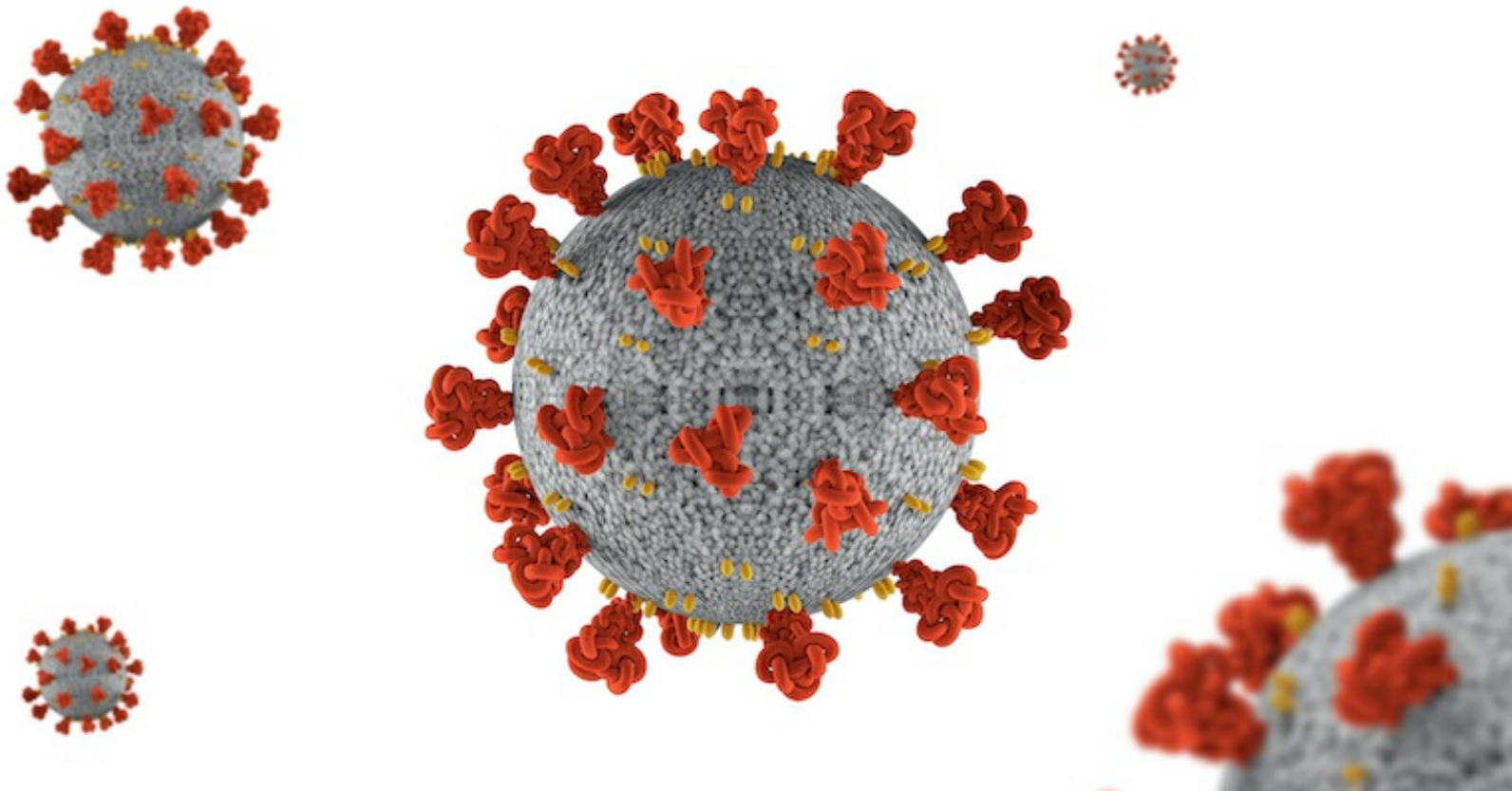
Besvarelsen din leveres automatisk når eksamenstida er ute og prøven stenger, forutsatt at minst én oppgave er besvart. Dette skjer selv om du ikke har klikket «Lever og gå tilbake til Dashboard» på siste side i oppgavesettet. Du kan gjenåpne og redigere besvarelsen din så lenge prøven er åpen. Dersom ingen oppgaver er besvart ved prøveslutt, blir ikke besvarelsen din levert.

I siste oppgave i eksamen skal du **laste prosjektet ditt opp samlet (Innlevering) som en ZIP-fil. Pakk HELE PROSJEKTMAPPEN DIN til en ZIP-fil og last opp.**

Trekk fra eksamen: Ønsker du å levere blankt/trekke deg, gå til hamburgermenyen i øvre høyre hjørne og velg «Lever blankt». Dette kan ikke angres selv om prøven fremdeles er åpen.

Tilgang til besvarelse: Du finner besvarelsen din i Arkiv etter at sluttida for eksamen er passert.

Målform/språk: Bokmål



Problembeskrivelse

Covid19 register

Den 31 desember 2019 rapporterte Helsedepartementet i Wuhan i Kina om et influensautbrudd av ukjent opprinnelse. 44 pasienter ble innrapportert med influensasymptomer. 11 av disse var kritisk syke. Dette ble starten på en av de værste pandemiene i verden i moderne historie.

Sykdommen fikk senere navnet COVID-19, og man kom frem til at sykdommen var forårsaket av et (av mange) coronavirus som mest sannsynlig stammer fra en bestemt rase av flaggermus i Kina. Dette viruset fikk navnet "SARS-CoV-2".

Utbredelsen av COVID-19 loggføres nå over hele verden.

Innledning

Les denne kravspesifikasjonen grundig. Gjør deretter oppgavene, som vil lede deg igjennom implementasjonen av dette prosjektet.

Kravspesifikasjon

Det skal utvikles et system for å registrere underlagsdata for til bruk for å utarbeide ulike statistikker. Hver registrering skal inneholde følgende informasjon:

- Dato for registrering
- Landet registreringen gjøres i (engelsk: country) - som tekst
- Antall nye smittede (engelsk: infected) - positivt heltall
- Antall nye døde (engelsk: deaths) - positivt heltall

I tabellen under finner du noen eksempler du kan bruke i din løsning:

Dato	Land	Smittede	Døde
19.01.2020	CHINA	136	1
05.02.2020	CHINA	3872	66
07.03.2020	NORGE	3	0
09.03.2020	USA	259	4
09.03.202	CHINA	45	23
22.03.2020	NORGE	240	8
24.03.2020	USA	20341	119
25.03.2020	CHINA	28	4
06.04.2020	NORGE	110	3
10.04.2020	USA	30859	2087

Funksjonalitet

Applikasjonen skal ha følgende funksjonalitet:
Brukeren må kunne

- registrere/legge inn COVID-19 tilfeller, hver registrering er pr dag.
- skrive ut liste over alle registreringer
- søke etter en registrering basert på dato
- søke etter registreringer *etter* en gitt dato
- regne ut samlet antall døde og smittede for et gitt land

Prosjektfiler

Det er allerede laget et prosjekt for deg (både for **BlueJ**, **Netbeans** og **IntelliJ**) som inneholder et meny-system med tilhørende klasser (tilsvarende prosjektoppgaven i emnet).
Last ned følgende ZIP-fil og pakk den ut på et sted på din harddisk der du ønsker å jobbe med oppgaven.

<TODO: Legg inn ny fil!!>>

ZIP-fil for nedlasting (klikk på lenka for å laste ned): [CovidStatsApplication.zip](#)

Løs oppgaven ved å bygge videre på prosjektet du finner i ZIP-filen.
Nedlastet prosjekt inneholder tre mapper:

- **CovidStatsApplication-BlueJ** - Inneholder prosjektet for bruk i BlueJ
- **CovidStatsApplication-NB** - Inneholder et Netbeans-prosjekt med alle nødvendige filer. Kildekoden ligger i src-mappen i denne mappen (altså er dette et selvstendig Netbeans-prosjekt som ikke deler kildekode med BlueJ-prosjektet).
- **CovidStatsApplication-IntelliJ** - Inneholder et IntelliJ-prosjekt med alle nødvendige filer. Kildekoden ligger i src-mappen.

Dersom du bruker annet IDE enn BlueJ, IntelliJ og/eller Netbeans (f.eks. Eclipse), så opprett en ny prosjekt-mappe på samme nivå som de to mappene over, og kopier .java-filene over til den nye prosjektmappen.

Prosjektet du har lastet ned og pakket ut inneholder et meny-system med en ferdig meny for prosjektet. For hvert menyvalg, gjøres et kall til en metode som er definert i **CovidStatsApp**. I resten av besvarelsen skal du fylle inn i metodene i denne klassen. NB! Du skal IKKE endre på metodene sine signaturer (ikke legge til parametre og/eller returtyper). Denne klassen er å regne som en del av brukergrensesnitt-laget (UI).

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, selvforklarende 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 (klare meldinger osv)

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

Oppgave 1 - Modellering

Denne oppgaven skal du løse ved å tegne på spesialpapir som du har fått utdelt av eksamensvakten (bruk blyant eller pen).

Oppgave 1 a) - Klassediagram

Les problembeskrivelse/kravspesifikasjonen og lag/tegn ett **klassediagram** for alle klassene du mener bør utvikles.

Vis i klassediagrammet hvilke **relasjoner** som finnes mellom klassene (hvem kjenner til hvem) og om det er **kardinalitet** knyttet til relasjonen.

Beskriv i tekstfeltet under hvordan du gikk frem for å identifisere kandidater til klassene. Beskriv også hva som er **oppgaven/rollen/ansvaret** til hver av klassene (tenk Javadoc av klassen).

Oppgave 1 b) - Aktivitetsdiagram

Fra kravspesifikasjonen, tegn et **aktivitetsdiagram** som viser programflyten sett fra sluttbruker.

Skriv ditt svar her...

Format

B

I

U

x_2

x^2

I_x

Ω

Σ

Words: 0

Maks poeng: 10

2



Oppgave 2

En viktig ferdighet som programvareutvikler er å finne frem i dokumentasjonen av biblioteker og rammeverk som kan være nyttig å bruke i din løsning.

I denne oppgaven skal du slå opp i dokumentasjonen for JDK 1.8 (API) og se nærmere på klassen **LocalDate**.

<TODO: Henvis til hvor på NTNU-PC'en kandidaten finner Javadoc>:
Åpne **docs/api/index.html** for å se JavaDoc for JDK 1.8

(a) **Oppgave 2 a)**

I hvilken **pakke** finner du klassen **LocalDate** ?
Hva må du skrive i koden din for å fortelle at du kommer til å bruke akkurat denne klassen ?

Skriv ditt svar her...

Format

B

I

U

x₂

x²

*I*_x

ABC

Words: 0

Maks poeng: 3

(b) Oppgave 2 b)

Hvilken metode i klassen vil du bruke for å lage en instans av dagens dato ?

Finnes det noen metode i klassen for å undersøke om en dato er tidligere enn en annen dato ?

Skriv i så fall **signaturen** til denne metoden i svarfeltet under.

Skriv ditt svar her...

Maks poeng: 5

(c) Oppgave 2 c)

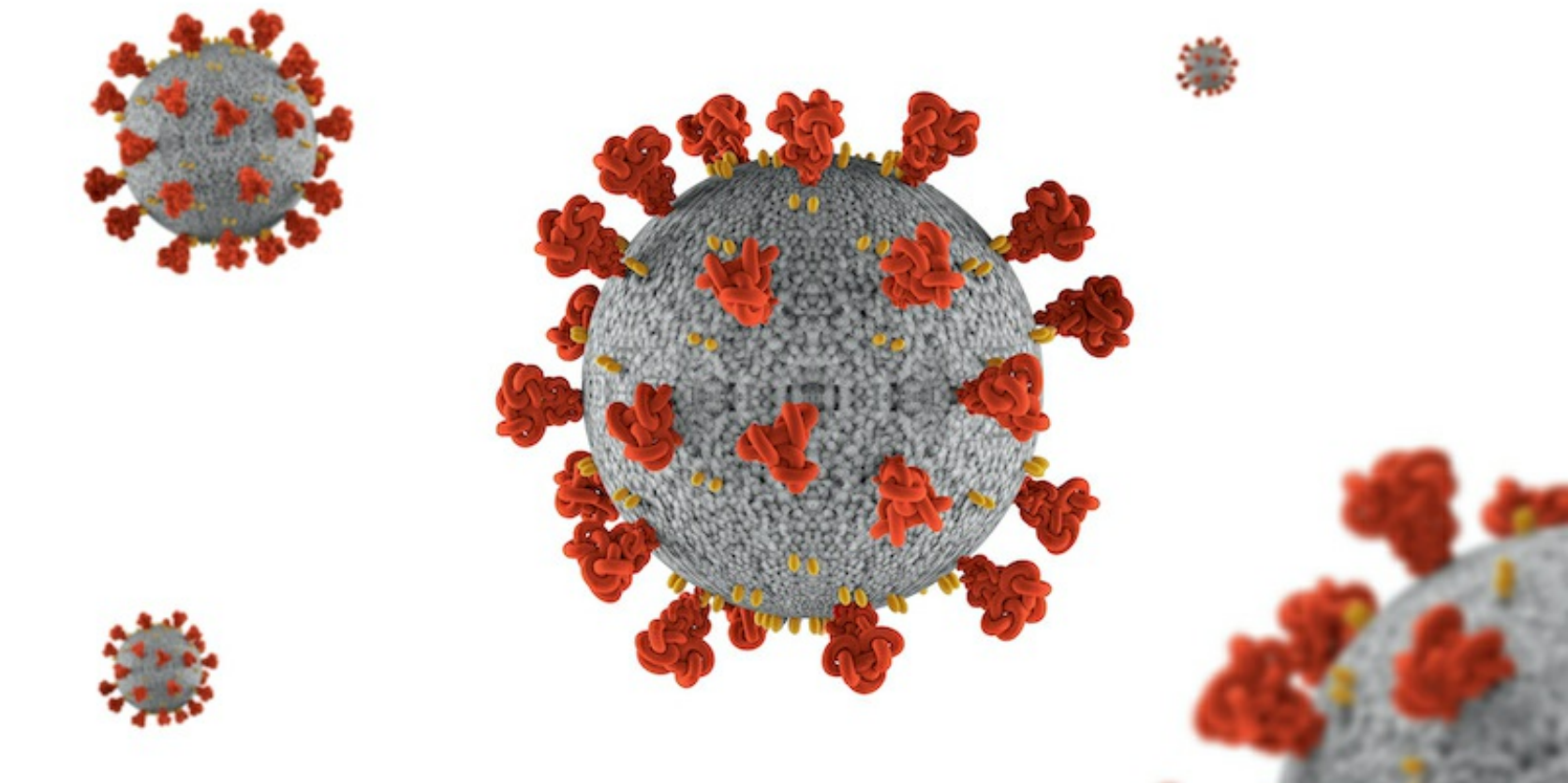
LocalDate-klassen har en metode med følgende signatur:

`LocalDate` [parse](#)([CharSequence](#) text)

Hva gjør denne metoden?

Skriv ditt svar her...

Maks poeng: 5



Oppgave 3

Gjennom resten av oppgavene skal du implementere applikasjonen.

Her kommer kravspesifikasjonen en gang til:

Kravspesifikasjon

Det skal utvikles et system for å registrere underlagsdata for til bruk for å utarbeide ulike statistikker. Hver registrering skal inneholde følgende informasjon:

- Dato for registrering
- Landet registreringen gjøres i (engelsk: country) - som tekst
- Antall nye smittede (engelsk: infected) - positivt heltall
- Antall nye døde (engelsk: deaths) - positivt heltall

I tabellen under finner du noen eksempler du kan bruke i din løsning:

Dato	Land	Smittede	Døde
19.01.2020	CHINA	136	1
05.02.2020	CHINA	3872	66
07.03.2020	NORGE	3	0
09.03.2020	USA	259	4
09.03.202	CHINA	45	23
22.03.2020	NORGE	240	8
24.03.2020	USA	20341	119
25.03.2020	CHINA	28	4
06.04.2020	NORGE	110	3
10.04.2020	USA	30859	2087
10.04.2020	CHINA	55	1

Funksjonalitet

Applikasjonen skal ha følgende funksjonalitet:
Brukeren må kunne

- registrere/legge inn COVID-19 tilfeller, hver registrering er pr dag.
- skrive ut liste over alle registreringer
- søke etter en registrering basert på dato

- søke etter registreringer *etter* en gitt dato
- regne ut samlet antall døde og smittede for et gitt land

Prosjektfiler

Det er allerede laget et prosjekt for deg (både for **BlueJ**, **Netbeans** og **IntelliJ**) som inneholder et meny-system med tilhørende klasser (tilsvarende prosjektoppgaven i emnet).

Last ned følgende ZIP-fil og pakk den ut på et sted på din harddisk der du ønsker å jobbe med oppgaven.

<TODO: Legg inn ny fil!!>>

ZIP-fil for nedlasting (klikk på lenka for å laste ned): [CovidStatsApplication.zip](#)

Løs oppgaven ved å bygge videre på prosjektet du finner i ZIP-filen.

Nedlastet prosjekt inneholder tre mapper:

- **CovidStatsApplication-BlueJ** - Inneholder prosjektet for bruk i BlueJ
- **CovidStatsApplication-NB** - Inneholder et Netbeans-prosjekt med alle nødvendige filer. Kildekoden ligger i src-mappen i denne mappen (altså er dette et selvstendig Netbeans-prosjekt som ikke deler kildekode med BlueJ-prosjektet).
- **CovidStatsApplication-IntelliJ** - Inneholder et IntelliJ-prosjekt med alle nødvendige filer. Kildekoden ligger i src-mappen.

Dersom du bruker annet IDE enn BlueJ, IntelliJ og/eller Netbeans (f.eks. Eclipse), så opprett en ny prosjekt-mappe på samme nivå som de to mappene over, og kopier .java-filene over til den nye prosjektmappen.

Prosjektet du har lastet ned og pakket ut inneholder et meny-system med en ferdig meny for prosjektet. For hvert menyvalg, gjøres et kall til en metode som er definert i **CovidStatsApp**. I resten av besvarelsen skal du fylle inn i metodene i denne klassen. NB! Du skal IKKE endre på metodene sine signaturer (ikke legge til parametre og/eller returtyper). Denne klassen er å regne som en del av brukergrensesnitt-laget (UI).

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, selvforklarende 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 (klare meldinger osv)

(a) **Oppgave 3 a)**

Lag en klasse i valgt IDE (BlueJ, IntelliJ, Netbeans osv) som representerer en COVID registrering (eng: covid location stats):

- 1. Gi klassen et fornuftig og forklarende navn.
- 2. Definere fornuftige felt av relevante datatyper (for dato kan du for eksempel bruke klassen **LocalDate** fra oppgave 2).
- 3. Implementer konstruktøren med nødvendige parametre
- 4. Implementer aksessor-metoder for klassen.
- 5. Implementer mutator-metoder for klassen (de du mener er nødvendige/nyttige).
- 6. Dokumenter klassen og metodene (JavaDoc)

Kopier koden for klassen fra din IDE inn i svarfeltet under.

Fyll inn koden din her...

1

Maks poeng: 10

(b) **Oppgave 3 b)**

Er det noen av verdiene til parametrene som inngår i konstruktøren til klassen din som burde sjekkes i forhold til størrelse/tallområde/verdi?

Har du i så fall noen tanker om hvordan du kan håndtere en slik kontroll?

Skriv ditt svar her...

Format

B


I


U


x_2


x^2


I_x


























Words: 0

Maks poeng: 5

(c) **Oppgave 3 c)**

Opprett et register for å holde på COVID-19 registreringene i applikasjonen din. Du velger selv hvilken klasse fra rammeverket med samlinger (*collection framework*) i Java du bruker.

Registeret bør som minimum ha følgende funksjonalitet:

- Legge til en COVID-19 registrering
- Søke etter COVID-19 registrering basert *på dato* (returnerer første registrering for gitt dato)
- Søke etter COVID-19 registreringer *etter en gitt dato* (her skal metoden returnere en samling av registreringer som oppfyller søkekriteriet)
- Returnere en iterator som andre objekter kan benytte for å iterere over registreringene i registeret.
- Returnere antallet COVID-19 registreringer i registeret.

Løs oppgaven i ditt IDE (BlueJ, Netbeans, IntelliJ), og kopier deretter koden din inn i svarfeltet under.

Lim inn koden for registeret ditt her...

1

Maks poeng: 22

(d) **Oppgave 3 d)**

Registeret bør også ha

- en metode som returnerer summen av antall dødsfall for et gitt land
- en metode som returnerer summen av antall dødsfall totalt

Implementer metodene, og kopier og lim inn koden for bare disse metodene i svarfeltet under.

Kopier og lim inn koden din her...

1	
---	--

Maks poeng: 10



Oppgave 4 - Brukergrensesnitt

I de neste oppgavene skal du ferdigstille applikasjonen ved å lage et tekstbasert grensesnitt. Du kan velge å ta utgangspunkt i klassen **CovidStatsApp** og ferdigstille denne, eller lage din egen klasse basert på forslaget under.

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 getMenuChoice()
{
    int menuChoice = 0;
    System.out.println("\n***** Covid Statistics Application v0.1 *****\n");
    System.out.println("1. Add a COVID-19 entry");
    System.out.println("2. List all COVID-19 entries");
    System.out.println("3. Find COVID-19 entry by date");
    System.out.println("4. Find all entries after date");
    System.out.println("5. Show the total deaths for a country");

    //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_COVID_ENTRY_TO_REGISTER:
            //TODO: Fill inn your code here....
            break;

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

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

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

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

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

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

```

... der følgende konstanter er definert:

```

// Constants representing the different menu choices
private static final int ADD_COVID_ENTRY_TO_REGISTER = 1;
private static final int LIST_ALL_COVID_ENTRIES = 2;
private static final int FIND_COVID_ENTRY_BY_DATE = 3;
private static final int FIND_COVID_ENTRY_AFTER_DATE = 4;
private static final int CALCULATE_TOTAL_DEATHS = 5;
private static final int EXIT = 6;

```

(a) **Oppgave 4 a)**

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.

Når du har gjort det, kopierer du koden til **hele klassen** og limer inn i svarfeltet under.

Lim inn koden fra hele klassen her...

1

Maks poeng: 25

(b) **Oppgave 4 b)**

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


x_2


x^2


I_x




















Ω





Σ

ABC



Words: 0

Maks poeng: 10

18/19

5 **Opplasting av prosjektfil**

Når du er ferdig med din besvarelse så lager du en **ZIP-fil** (IKKE RAR eller annet format!!) av **hele prosjektmappen din** med alle undermapper og IDE-spesifikke filer (se figur under) og laster opp ZIP-filen din.




Figur 1 - Komprimer mappen **CovidStatsApplication** (uthevet i blå over), eller tilsvarende mappe hos deg, til en ZIP-fil.

Du lager ZIP-fil av en mappe på følgende måte:


- I windows: høyreklikk på mappen i filutforskeren din, og velg "Send til->Komprimert (zippet) mappe"
- På MacOSX: høyreklikk mappen i Finder, og velg "Komprimer..."

Sjekk at størrelsen på ZIP-filen viser mer enn noen få bytes. Er du i tvil om du har fått ZIP'et alt, pakk ut ZIP-filen til annet sted på din datamaskin og gå over innholdet. Ser alt greit ut, kan du levere inn ZIP-filen her.



Last opp ZIP-filen her. Maks én fil. (Kun filer av typen .ZIP vil bli akseptert).

Følgende filtyper er tillatt: **.zip** Maksimal filstørrelse er **50 GB**.

 Velg fil for opplasting

Maks poeng: 0