Høgskulen på Vestlandet

EKSAMENSOPPGAVE/EKSAMENSOPPGÅVE

Emnekode: DAT100

Emnenavn/Emnenamn: Grunnleggende Programmering

Utdanning/kull/klasse:
Dataingeniør + Informasjonsteknologi / H2022 /
1Data og 1Informajonsteknologi

Dato: 20. desember 2022

Eksamensform: Digital skoleeksamen

Eksamenstid: 4 klokketimer

Antall eksamensoppgaver/ Tal på eksamensoppgåver: 5

Fagansvarlig/ Fagansvarleg:
Sven-Olai Høyland (472 59 543), Lars Michael Kristensen (938 66 491)

Språk: For å redusere risikoen for å få ulike bokmål- og nynorsk-versjoner, har vi begge versjoner sammen for hvert delspørsmål. Ved ulikhet mellom versjonene, bruker vi «|» mellom korte setninger og ramme rundt nynorsk for formuleringer over flere linjer. Du kan oversette navn i gitt kode til nynorsk / engelsk om du ønsker det.

Dersom du mener at noen opplysninger mangler i en oppgave-formulering, beskriv antagelsene du har gjort for å svare på oppgaven.

Oppgave | Oppgåve 1 [12%]

Du finner oppgaveteksten i Wiseflow.

Oppgave | Oppgåve 2 [13%]

Du finner oppgaveteksten i Wiseflow.

Oppgave | Oppgåve 3 - 5 [75%]

Du finner oppgavene på de neste sidene, men de skal besvares i Wiseflow.

Oppgave | Oppgåve 3 [30%]

I denne oppgaven skal du implementere klasser som kan brukes til å sende meldinger mellom en applikasjon og en værstasjon der det inngår sensorer som måler temperatur, CO_2 og luftfuktighet.

I denne oppgåva skal du implementere klassar som kan brukast til å sende meldingar mellom ein applikasjon og ein verstasjon der det inngår sensorar som måler temperatur, CO_2 og luftfukt.

Oppgave | Oppgåve 3a

Implementer den abstrakte klassen `Melding` med objektvariablen `mid` (heltall som identifiserer meldingen). Objektvariablen skal bare være synlig innenfor klassen.

Implementer den abstrakte klassen `Melding` med objektvariablen `mid` (heiltal som identifiserer meldinga). Objektvariablen skal berre vere synleg innanfor klassen.

Oppgave | Oppgåve 3b

Implementer en konstruktør | Implementer ein konstruktør

```
Melding(int mid)
```

som setter objektvariablen `mid` lik verdien gitt som parameter.

som set objektvariablen `mid` lik verdien gitt som parameter.

Oppgave | Oppgåve 3c

Implementer get/set-metode (hent/sett-metode) for objektvariablen `mid` i klassen.

Implementer get/set-metode (hent/sett-metode) for objektvariablen `mid` i klassen.

Oppgave | Oppgåve 3d

Implementer metoden

```
public String toString ()
```

som returnerer en strengrepresentasjon av tallet som er lagret i objektvariablen `mid`.

som returnerer ein strengrepresentasjon av talet som er lagra i objektvariablen `mid`.

Eksempel. Dersom `mid` har verdien `1` skal tekststrengen `"1"` returneres.

Døme. Dersom `mid` har verdien 1 skal tekststrengen `"1"` returnerast.

Oppgave | Oppgåve 3e

Implementer subklassen `Forespørsel` av klassen `Melding`. Objekter av klassen skal i tillegg til objektvariablen som arves fra superklassen ha en objektvariabel `måling` av oppramstypen `Måling` gitt nedenfor.

Implementer subklassen `Forespørsel` av klassen `Melding`. Objekt av klassen skal i tillegg til objektvariablen som blir arva frå superklassen ha ein objektvariabel `måling` av oppramstypen `Måling` gitt nedanfor.

```
public enum Måling {

      TEMPERATUR, FUKTIGHET, CO2
}
```

Klassen skal ha en konstruktør

```
      public Forespørsel(int mid, Måling måling)
```

som gir verdi til alle objektvariablene inklusiv den som er arvet fra superklassen.

Klassen skal ha ein konstruktør

```
      public Forespørsel(int mid, Måling måling)
```

som gir verdi til alle objektvariablane inklusiv den som er arva frå superklassen

Oppgave | Oppgåve 3f

Implementer metoden

```
   public String toString()
```

i klassen `Måling` som returnerer en tekststreng med verdiene som er lagret i objektvariablene.

i klassen `Måling` som returnerer ein tekststreng med verdiane som er lagra i objektvariablane.

Eksempel. For et objekt der `måling` er `CO2` og verdier ellers er som i oppgave 3d) skal strengen som returneres ha innhold som vist nedenfor

Døme. For eit objekt der `måling` er `CO2` og verdiar elles er som i oppgåve 3d) skal strengen som blir returnert ha innhald som vist nedanfor.

```
1 Forespørsel CO2
```

Oppgave | Oppgåve 3g

Implementer subklassen `Svar` av klassen `Melding`. Objekter av klassen skal i tillegg til objektvariablen som arves fra superklassen ha en objektvariabel `verdi` (flyttall).

Klassen skal ha en konstruktør

```
public Svar(int mid, double verdi)
```

som gir verdi til alle objektvariablene inklusiv de som er arvet fra superklassen.

Implementer subklassen `Svar` av klassen `Melding`. Objekt av klassen skal i tillegg til objektvariablen som blir arva frå superklassen ha ein objektvariabel `verdi` (flytttal).

Klassen skal ha ein konstruktør

```
public Svar(int mid, double verdi)
```

som gir verdi til alle objektvariablane inklusiv dei som er arva frå superklassen.

Oppgave | Oppgåve 3h

Implementer metoden

```
public String toString()
```

i klassen `Svar` som returnerer en tekststreng med verdiene som er lagret i objektvariablene.

i klassen `Svar` som returnerer ein tekststreng med verdiane som er lagra i objektvariablane.

Eksempel. For et objekt der `verdi` er 3.0 og verdier ellers er som i oppgave 3d) skal strengen som returneres ha innhold som vist nedenfor.

Døme. For eit objekt der `verdi` er 3.0 og verdiar elles som i oppgåve 3d) skal strengen som blir returnert ha innhald som vist nedanfor.

```
1 Svar 3.0
```

Oppgave | Oppgåve 3i

Implementer en metode | Implementer ein metode

```
public static boolean match(Melding m1, Melding m2)
```

som returnerer `true` hvis `mid` objektvariablen i de to objektene som gis med som parametre er lik og `false` ellers.

som returnerer `true` viss `mid` objektvariablen i dei to objekta som blir gitt med som parametrar er lik og `false` elles.

Oppgave | Oppgåve 3j

Implementer metoden

```
public static Svar mottak(Forespørsel f, double temperatur,
                          double fuktighet, double co2)
```

som oppretter og returnerer et `Svar`-objekt med verdi for objektvariable i henhold til tabellene nedenfor som illustrerer sammenhengen mellom parametre og retur-verdi.

som opprettar og returnerer eit `Svar`-objekt med verdi for objektvariable i samsvar med tabellane nedanfor som illustrerer samanhengen mellom parametrar og retur-verdi.

Forespørsel Objekt f				
mid	måling	temperatur	fuktighet	co2
1	TEMPERATUR	20.0	67.0	2.3
2	FUKTIGHET	22.0	70.0	2.5
3	CO2	18.0	65.0	2.0

Svar-objekt	
mid	verdi
1	20.0
2	70.0
3	2.0

Oppgave | Oppgåve 4 [30%]

Denne oppgaven handler om bruk av referansetabeller for lagre avlesninger av total og forbrukte antall kilowatt timer (Kwh) på en el-måler.

Du kan anta at det allerede finnes en Java-klasse `Avlesing` med objekt-variabler `totalkwh` (heltall som representerer totale antall forbrukte kilowatt timer) og `forbrukkwh` (heltall som representerer forbrukte antall kilowatt timer siden siste avlesing).

Du kan videre anta at klassen har en konstruktør som kan gi verdier til alle objekt-variablene samt get/set (hent/sett) metoder og en `toString`-metode som skriver ut verdien av objekt-variablene.

Denne oppgåva handlar om bruk av referansetabellar for å lagre avlesingar av total og forbrukte talet på kilowatt timar (Kwh) på ein el-målar.

Du kan anta at det allereie finst ein Java-klasse `Avlesing` med objekt-variablar `totalkwh` (heiltal som representerer samla talet på forbrukte kilowatt timar) og `forbrukkwh` (heiltal som representerer forbrukte talet på kilowatt timar sidan siste avlesing).

Du kan vidare anta at klassen har ein konstruktør som kan gi verdiar til alle objekt-variablane og dessutan get/set (hent/sett) metodar og ein `toString`-metode som skriv ut verdien av objekt-variablane.

Oppgave | Oppgåve 4a

Implementer en klasse `Dagsforbruk` med objektvariablen `avlesninger` som er en referansetabell med `Avlesing`-objekt. Referansetabellen skal brukes til å lagre en referanse til et `Avlesing`-objekt for hver av de 24 timer i døgnet.

Implementer ein klasse Dagsforbruk med objektvariablen `avlesninger` som er ein referansetabell med `Avlesing`-objekt. Referansetabellen skal brukast til å lagra ein referanse til eit `Avlesing`-objekt for kvar av dei 24 timar i døgnet.

Oppgave | Oppgåve 4b

Implementer en konstruktør | Implementer ein konstruktør

```
public Dagsforbruk()
```

som oppretter en tabell av lengde `24` der det kan lagres referanser til `Avlesing`-objekt og setter objektvariablen `avlesinger` til å peke på tabellen.

som opprettar ein tabell av lengde 24 der det kan lagrast referansar til Avlesing-objekt og set objektvariablen `avlesinger` til å peika på tabellen.

Oppgave | Oppgåve 4c

Implementer metoden

```
public void registerAvlesing(int time, Avlesing avl)
```

som setter inn avlesningen gitt ved parameteren `avl` i referansetabellen på posisjonen mellom 0 og 23 som er gitt ved parameteren `time`.

som set inn avlesinga gitt ved parameteren `avl` i referansetabellen på posisjonen mellom 0 og 23 som er gitt ved parameteren `time`.

Oppgave | Oppgåve 4d

Implementer metoden

```
public void visForbruk()
```

som skriver ut på skjermen avlesningene som er registrert i referansetabellen.

For posisjoner i referansetabellen som peker på et `Avlesing`-objekt skal posisjon skrives ut sammen med det totale og forbrukte antall kilowatt timer. For posisjoner som inneholder en `null`-peker skal posisjon og teksten `"ikke registert"` skrives ut.

som skriv ut på skjermen avlesingane som er registrert i referansetabellen.

For posisjonar i referansetabellen som peikar på eit `Avlesing`-objekt skal posisjon skrivast ut saman med det totale og forbrukte talet på kilowatt timar. For posisjonar som inneheld ein `null`-peikar skal posisjon og teksten `"ikkje registert"` blir skriven ut.

Oppgave | Oppgåve 4e

Implementer metoden

```
public boolean altRegistert()
```

som returnerer `true` om der er `Avlesing`-objekt på alle posisjoner i tabellen. Ellers returneres `false`.

som returnerer `true` om der er `Avlesing` -objekt på alle posisjonar i tabellen. Elles blir `false` returnert.

Oppgave | Oppgåve 4f

Implementer metoden

```
public int finnMaksKwh()
```

som returner den største verdien av forbrukte kilowatt timer som er lagret i tabellen (objektvariablen `forbrukkwh` i `Avlesing` -objekt). Ta høyde for at der kan være `null`-pekere i tabellen.

som returner den største verdien av forbrukte kilowatt timar som er lagra i tabellen (objektvariablen `forbrukkwh` i `Avlesing`-objekt). Ta høgd for at der kan vere `null`-peikarar i tabellen.

Oppgave | Oppgåve 4g

Implementer metoden

```
public double gnsKwh()
```

som returnerer det gjennomsnittlige antall forbrukte antall kilowatt per time over 24 timer. Du kan anta at der ikke er noen `null`-pekere i tabellen.

som returnerer det gjennomsnittlege talet på forbrukte talet på kilowatt per time over 24 timar. Du kan anta at der ikkje er nokon `null`-peikarar i tabellen.

Oppgave | Oppgåve 4h

Implementer metoden

```
public double beregnPris(double[] timepriser)
```

som gitt en tabell med `timepriser` for de 24 timene i døgnet, beregner prisen på forbruket som er registret i referansetabellen.

Hint: for hver time (posisjon) i tabellen skal forbrukes multipliseres med timeprisen i den tilsvarende posisjon i parameteren `timepriser`.

som gitt ein tabell med timeprisar for dei 24 timane i døgnet bereknar prisen på det forbruket som er registeret i referansetabellen.

Hint: for kvar time (posisjon) i tabellen skal forbrukast bli multiplisert med timeprisen i den tilsvarande posisjonen i parameteren `timepriser`.

Oppgave | Oppgåve 5 [15%]

Denne oppgaven omhandler bruk av to-dimensjonale tabeller med tegn (typen `char`) for å representere obligatoriske oppgaver for studenter der `'L'` brukes om oppgaven er levert, `'G'` brukes om oppgaven er godkjent og tegnverdi `0` (standardverdien for char som automatisk tildeles i Java) brukes om ikke noe er registrert.

Eksempel. Et emne med 3 studenter og 4 oppgaver kan representer av en tabell med 3 rader (rekker) og 4 elementer i hver rad (rekke).

Denne oppgåva omhandlar bruk av to-dimensjonale tabellar med teikn (typen `char`) for å representere obligatoriske oppgåver for studentar der `'L'` blir brukt om oppgåva er levert, `'G'` blir brukt om oppgåva er godkjent og teiknverdi `0` (standardverdien for char som automatisk blir tildelt i Java) blir brukt om ikkje noko er registrert.

Døme. Eit emne med 3 studentar og 4 oppgåver kan representerast av ein tabell med 3 rader (rekker) og 4 element i kvar rad (rekke).

Starten på klassen der du skal implementere metoder er gitt nedenfor. Objektvariabelen `innleveringer` skal brukes til å representere innleveringer.

Starten på klassen der du skal implementere metodar er gitt nedanfor. Objektvariabelen `innleveringer` skal brukast til å representere innleveringar.

```
public class Vurderinger {

      private char[][] innleveringer;

      // her skal du skrive metoder
}
```

Oppgave | Oppgåve 5a

Implementer en konstruktør | Implementer ein konstruktør

```
public Vurderinger(int studentantall, int oppgaveantall)
```

som oppretter en to-dimensjonal tabell med antall rader gitt med `studentantall` og antall elementer i hver rad gitt med parameteren `oppgaveantall` og som setter objektvariablen `innleveringer` til å peke på den to-dimensjonale tabellen.

som opprettar ein to-dimensjonal tabell med talet på rader gitt med studenttal og talet på element i kvar rad gitt med parameteren `oppgaveantall` og som set objektvariablen `innleveringer` til å peike på den to-dimensjonale tabellen.

Oppgave | Oppgåve 5b

Implementer metoden

```
public char hentStatus(int s, int o)
```

som henter status for studenten i rad `s` og oppgaven i posisjon `o`.

som hentar status for studenten i rad `s` og oppgåva i posisjon `o`.

Oppgave | Oppgåve 5c

Implementer metoden

```
public void sett(int s, int o, char status)
```

som setter elementet i rad `s`, posisjon `o` til verdien gitt med parameteren `status`.

som set elementet i rad `s`, posisjon `o` til verdien gitt med parameteren `status`.

Oppgave | Oppgåve 5d

Implementer metoden

```
public void erGodkjent(int s, int o)
```

som returnerer `true` dersom studenten på rad `s` har fått godkjent oppgaven i posisjon indeks `o` og `false` ellers.

som returnerer `true` dersom studenten på rad `s` har fått godkjent oppgåva gitt i posisjon `o` og `false` elles.

Oppgave | Oppgåve 5e

Implementer metoden

```
public void vis()
```

som skriver ut innholdet av tabellen på formatet angitt nedenfor der det skrives ut et mellomrom om der ikke er registrert hverken godkjent eller levert for en gitt innlevering. Tallet i første kolonne i utskriften er indeks (posisjon) for raden som representerer en student.

som skriv ut innhaldet av tabellen på formatet gitt nedanfor der det blir skrive ut eit mellomrom om der ikkje er registrert verken godkjent eller levert for ei gitt innlevering. Talet i første kolonne i utskrifta er indeks (posisjon) for rada som representerer ein student.

```
0|G| | | |
1| | | | |
2| | | |G|
```

Oppgave | Oppgåve 5f

Implementer metoden

```
public boolean alleGodkjent(int s)
```

som returnerer `true` om studenten representert på rad `s` har fått alle innleveringer godkjent. Ellers returneres `false`.

som returnerer `true` om studenten representert på rad `s` har fått alle innleveringar godkjent. Elles blir `false` returnert.

Oppgave | Oppgåve 5g

Implementer metoden

```
public boolean alleLevert(int o)
```

som returnerer `true` dersom alle studenter har levert eller fått godkjent innleveringen representert av kolonne `o`. Ellers returneres `false`.

som returnerer `true` dersom alle studentar har levert eller fått godkjent innleveringa representert av kolonne `o`. Elles blir `false` returnert.