Kandidatnr

UNIVERSITETET I OSLO

Det matematisk-naturvitenskapelige fakultet

Eksamen i:	**PRØVEEKSAMEN INF1000**
Eksamensdag:	Prøveeksamen 22.11.2011
Tid for eksamen:	12:15-16:15
Oppgavesettet er på 17 side(r), med 13 oppgaver	
Vedlegg:	Ingen
Tillatte hjelpemidler:	Alle trykte og skrevne

- Les ***nøye*** gjennom hver oppgave før du løser den. For hver oppgave er angitt det maksimale antall poeng du kan få hvis du svarer helt riktig. Summen av poengene er 240, slik at 1 poeng tilsvarer 1 minutt av eksamenstiden. Pass på at du bruker tiden din riktig.

- Kontroller også at oppgavesettet er komplett før du begynner å besvare det. Dersom du savner opplysninger i oppgaven, kan du selv legge dine egne forutsetninger til grunn og gjøre rimelige antagelser, så lenge de ikke bryter med oppgavens "ånd". Gjør i så fall rede for forutsetningene og antagelsene du gjør.

- Dine svar ***skal*** skrives på disse oppgavearkene, og ikke på separate ark. Dette gjelder både spørsmål med avkrysningssvar og spørsmål hvor du bes om å skrive programkode. I de oppgavene hvor det skal skrives programkode, anbefales det at du først skriver en kladd på eget ark før du fører svaret inn i disse oppgavearkene på avsatt plass.

- Noen av spørsmålene er flervalgsoppgaver. På disse oppgavene får du poeng etter hvor mange korrekte svar du gir. Du får ikke poeng hvis du lar være å besvare et spørsmål, eller dersom du krysser av begge svaralternativer.

- Hvis du har satt et kryss i en avkrysningsboks og etterpå finner ut at du ikke ønsket å krysse av der, kan du skrive "FEIL" like til venstre for den aktuelle avkrysningsboksen.

- Husk å skrive såpass hardt at besvarelsen blir mulig å lese på alle gjennomslagsarkene, men ikke legg andre deler av eksamensoppgaven under når du skriver.

Oppgave 1 (1 poeng)

Hvor mange char-verdier er det plass til i arrayen **`tegn`**?

```
char [][] tegn = new char  [5][2];
```

Svar: _____

Oppgave 2 (9 poeng)

Er disse programsetningene lovlige i Java?

JA	NEI	
☐	☐	`int tall = null;`
☐	☐	`int x = new int [5];`
☐	☐	`boolean[]s = new s[8];`
☐	☐	`int i =1, j= i/3 – i ;`
☐	☐	`String char = "INF1000";`
☐	☐	`int [] x2 = new double [2];`
☐	☐	`double [][] XX = 1.23;`
☐	☐	`String[] s = {"opp", "ned"};`
☐	☐	`double [] x = new double{1.0, 2.0, 3.0, 4.0};`
☐	☐	`int[][] a, [][]b;`

Oppgave 3 (12 poeng)

Hva skrives ut i hvert av programeksemplene nedenfor?

3a)

```
String s;
if (true) {
    s = "Per";
} else {
    s= "Pål";
}
System.out.println (s);
```

Svar: _____

3b)

```
int tall = 5;
if (! (tall < 10)) {
    System.out.println (tall);
} else {
    System.out.println (tall + tall);
}
```

Svar: _____

3c)

```
tall = 0;
System.out.println (--tall);
System.out.println (tall++);
System.out.println (tall);
```

Svar: _____

3d)

```
int tall = 0;
tall = tall-- + tall++;
System.out.println (tall);
```

Svar: _____

Oppgave 4 (4 poeng)

Hva skrives ut, evt hvor mange ganger, i programeksempelet nedenfor?

```
String t = "INF1000";
for (int i = 0; i < t.length(); i++) {
    String u = "";
    for (int j = t.length(); j > i; j--) {
     u = t.charAt(j-1) + u;
    }
    u = t.substring (0,i) + u;
    System.out.println (u);
}
```

Svar: _____

.

Oppgave 5 (10 poeng)

Skriv en metode som tar et antall minutter som parameter, regner om i timer og minutter og skriver ut resultatet på skjermen.

```
void omregning(int min) {




}
```

Oppgave 6 (20 poeng)

Skriv en metode som tar en String med et mobilnummer som input. Metoden skal returnere mobilnummeret som en String uten innledende '+', og med landkode først. Du kan anta at nummeret i parameteren alltid er 8-sifret, men *kan* i tillegg inneholde landkode (alltid 2-sifret) med eller uten '+' foran. Dersom landkode mangler kan du anta at det er et norsk nummer, og legge til "47" først i teksten som returneres.

```
String rettMobil(String mob) {
```

```
}
```

Oppgave 7 (8 poeng)

Anta at følgende kodelinjer utføres:

```
String t = "llmmnop";
String u = t.charAt (1) + "mmno";
if (t.indexOf(u)>0)
    System.out.println (1);
if (t.startsWith (u))
    System.out.println (2);
if (t.endsWith (u))
    System.out.println (3);
```

Hva skrives ut på skjermen?

Svar: _____

Oppgave 8 (8 poeng)

Hva returneres fra metodekallet **`metode (5,4)?`**

```
int metode (int tall1, int tall2) {
int svar = tall1 * tall2;
while (svar > tall1) {
    svar = svar-tall1;
}
return svar;
}
```

Svar: _____

Oppgave 9 (30 poeng)

P. Smart har gjennom en årrekke hjulpet et økende antall venner og slektninger med reparasjoner og vedlikehold av deres påhengsmotorer. Etter hvert er det blitt mange motorer og reparasjoner, og han ønsker seg derfor et program som kan holde orden på data om de motorene han har sett på og hva han har foretatt seg med hver enkelt. Du skal hjelpe til med å implementere dette systemet i Java.

P. Smart ønsker å bruke klassen **`Rep`**, med String-variable **`hva`** og **`dato`**, og int-variabelen **`minBrukt`** til å lagre data om de gangene han har gitt bistand. Datoen lagres som en String på formen aammdd, **`hva`** er en tekststreng der han fritt kan beskrive det som er gjort og **`minBrukt`** inneholder antall minutter brukt. Du skal skrive klassen Rep med variable og konstruktør der disse variablene er parametre.

For enkelt å kunne få oversikt over de gangene han har hjulpet til med bestemte typer problemer, skal du også skrive metoden **`finnes (String s)`** i klassen **`Rep`** som søker etter stikkord **`s`** i variabelen **`hva`**.

I denne oppgaven kan det være nyttig å bruke String-metodene:

`int indexOf (String s)` // søker etter strengen s i denne strengen, returnerer indeks der den ble funnet eller -1 hvis den ikke finnes. Metoden skiller mellom store og små bokstaver.

`String toUpperCase()` // returnerer en kopi av denne strengen, men med alle små bokstaver gjort om til store.

Oppgave 9a)

Skriv klassen **`Rep`**.

Oppgave 9b)

P. Smart ønsker en klasse **`Eier`** for å holde styr på data om eiere. Foreløpig trenger han bare to variable i klassen: **`String navn`** og **`String mobil.`** Deklarer klassen **`Eier`** med variable og konstruktør med parameter for hver variabel. Mobilnummeret som oppgis skal sjekkes og endres slik at det er i et fast format. Du kan anta at metoden **`rettMobil()`** fra oppgave 6) er ferdig deklarert i klassen **`Eier`**.

Oppgave 9c)

P. Smart har skissert klassen **`Motor`** som vist nedenfor. Skriv ferdig metodene i klassen.

```
class Motor {
    Eier minEier;
    String merke;
    int hk;
    HashMap <String, Rep> utfoert;

    Motor (Eier minEier, String merke, int hk) {









    }

    void nyRep (String hva, String naar, int min) {
// Dersom det ikke er registrert noen reparasjon av denne
motoren
// på gitt dato, opprettes et nytt Rep-objekt. Ellers legges
// informasjon om utført arbeid og antall minutter til det
// eksisterende Rep-objektet.
```

```
    }


    int beregnAntMin () {
// Metoden legger sammen reparasjonstiden for alle reparasjoner
på
// denne motoren, og returnerer dette









    }

}
```

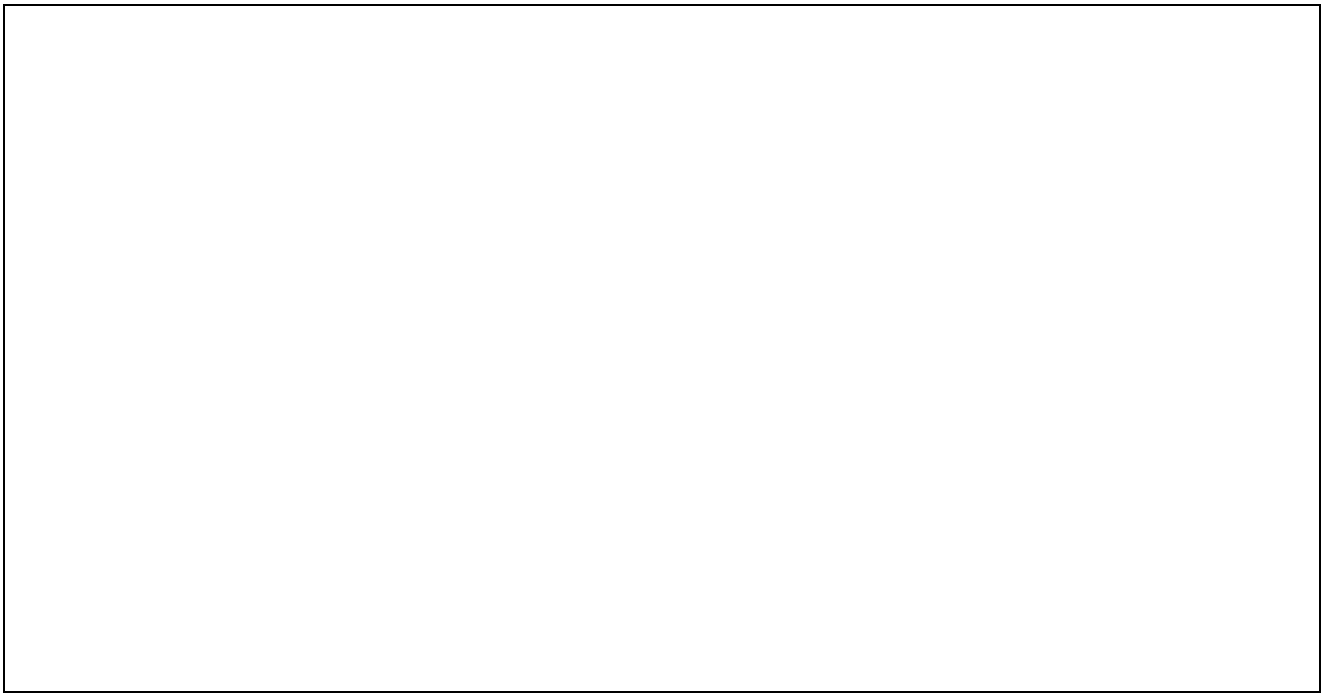
Oppgave 10 (28 poeng)

10a) Testprogram for klassene Rep og Motor

Lag et test-program som benytter koden fra oppgave 9 til å gjøre følgende:

- Opprette ett eier-objekt
- Opprette ett motor-objekt, med den nyopprettede eieren som eier
- Registrere 3 reparasjoner på motoren
- Skrive ut totalt antall timer og minutter brukt på reparasjoner for denne motoren.

Du velger selv verdier for variablene i de objektene du oppretter.



10b) UML objektdiagram

Lag et UML objektdiagram som viser datastruktur med de objektene du opprettet i testprogrammet i oppg. 10a). Du trenger ikke vise metoder i objektene.

Oppgave 11 (80 poeng)

Du skal skrive ferdig en første versjon av et system for P. Smart basert på klassene Rep, Motor og Eier fra oppgave 9. Systemet skal inneholde en klasse MotorSystem som holder rede på opptil 100 påhengsmotorer.

Flere motorer skal kunne peke til samme eier. Systemet må derfor lete gjennom eksisterende eiere for å sjekke om eieren av en ny motor allerede er registrert på en annen motor, og i så fall la den nye motoren peke til samme objekt. For å sjekke om en eier allerede er registrert ønsker han å bruke eierens mobilnummer, siden han har sikret seg at mobilnumrene er lagret på en uniform måte.

Fyll ut metodene nedenfor i klassen **`MotorSystem`**, basert på opplysningene ovenfor:

- Metode **`lagreData`** som skriver alle data på fil (du velger selv formatet på filen – skilletegn og/ eller antall av for eksempel reparasjoner for en motor)
- Metode **`beregnTotMin`** som går gjennom alle reparasjoner i systemet og regner ut antall minutter som er brukt totalt – og skriver det ut på skjerm i timer og minutter (bruk metoden **`omregning`** fra opggave 5)
- Konstruktør **`Motorsystem`** som bygger opp datastrukturen og leser alle data fra fil.

```
// import-setninger

class MotorRep {
  public static void main (String [] args) {

    final String FILNAVN = "motor.txt";
    MotorSystem ms = new MotorSystem (FILNAVN);

    // kall på ordreløkke e.l. – dette skal du IKKE skrive

  } // end main()
} //end class MotorRep


class MotorSystem {
   // deklarasjon av variable – skal skrives
```

```
Motorsystem (String filnavn) { // konstruktør - skal skrives

}
```

```
    lagreData(String filnavn) { // denne skal skrives
```

```
}
beregnTotMin () { // denne skal skrives
}
```

```
    // evt hjelpemetoder

} // end class MotorSystem
```

Oppgave 12 (20 poeng)

Universitetet i Langtvekkistan planlegger å lage et system som rapporterer direkte til Studentenes Lånekasse i landet hvilke studenter som ikke har god nok studieprogresjon i forhold til normen hvert semester. De vil for hver slik student sende en epost til Lånekassa om hvor få eksamener disse studentene har tatt, basert på sitt eget studentregister der blant annet eksamensresultater lagres. Disse studentene vil så få redusert lån neste semester.

Du skal skrive en vurdering om det er tillatt for Universitetet i Langtvekkistan etter bestemmelsene i " Lov om behandling av personopplysninger" (de har den samme loven i Langtvekkistan som i Norge) å sende slike beskjeder om studenter som ikke jobber bra nok til Lånekassa. Begrunn svaret med å vise til konkrete paragrafer som du mener er relevante, og hvorfor de er det.