UNIVERSITETET I OSLO
Det matematisk-naturvitenskapelige fakultet

Skriftlig midtveiseksamen i IN1010
2026 vår
Tid: 20. mars kl 15:00 til 17:00

Tillatte hjelpemidler:
Ingen

Annen informasjon

Faglærerne kommer *ikke* til å besøke eksamenslokalet under eksamen. Om du mener at det er noe *alvorlig* galt med en oppgave, kan du ta kontakt med en eksamensvakt.

Alle oppgavene er uavhengig av hverandre og kan løses i vilkårlig rekkefølge.

Med mindre det står noe annet, er poenggivingen slik: Du får poeng for rett svar og du får 0 poeng for galt svar.

Det finnes en kalkulator tilgjengelig; se link nederst på skjermen.

1 Hvilke verdier skrives ut?

a[0] er

a[1] er

a[2] er

a[3] er

a[4] er

a[5] er

Maks poeng: 9

```java
class Arrayoperasjoner {
    public static void main (String[] args) {
        int a[] = new int[6];
        int v = 3;

        a[0] = 0;
        for (int ix = 1;  ix < a.length;  ix++) {
            a[ix] = a[ix-1] + v;
            v++;
        }

        for (int ix = 0;  ix < a.length;  ix++) {
            System.out.println("a[" + ix + "] er " + a[ix]);
        }
    }
}
```

2

Dette programmet skal skrive ut summen av alle heltallene i filen **tall.data**. Fyll inn feltene som mangler.

Hjelp

```
import java.util.Scanner;
import java.io.*;

class LesTall {
   public static [ ] main ( [ ] [ ] args ) {
        [ ] s = null;
       try {
          s = new [ ] ( new [ ] ( "tall.data" ) ) ;
       } [ ] ( FileNotFoundException e ) {
            [ ] ( "Kan ikke lese tall.data." ) ;
            [ ] ( 1 ) ;
       }

       int sum = 0 ;
       while ( s. [ ] ( ) ) {
          sum = sum + s. [ ] ( ) ;
       }
        [ ] ( "Summen er " + sum ) ;
   }
}
```

System.exit	System.out.println	void	catch
String	hasNextInt	Scanner	File
nextInt			

Maks poeng: 11

3 Hva skrives ut?
Velg ett alternativ:

- Kjørefeil
- 0
- 7
- 13
- 8
- Noe annet

Maks poeng: 5

```java
class Arrayoppsett {
    public static void main (String[] args) {
        int[] a = new int[8];

        try {
            a[0] = 0;
            a[1] = 1;

            for (int ix = 2;  ix < a.length;  ix++) {
                a[ix] = a[ix-2] + a[ix-1];
            }

            System.out.println(a[8]);
        } catch (Exception e) {
            System.out.println("Kjørefeil");
        }
    }
}
```

4 Fyll inn så vi kan sammenligne Dato-er med compareTo:

Hjelp

```
class Dato implements [          ] < [          ] > {
   int år, måned, dag;

   [          ]
   public [          ] [          ] ( [          ] d2 ) {
      if ( år < d2.år ) return [          ] ;
      if ( år > d2.år ) return [          ] ;
      if ( måned < d2.måned ) return [          ] ;
      if ( måned > d2.måned ) return [          ] ;
      if ( dag < d2.dag ) return [          ] ;
      if ( dag > d2.dag ) return  1 ;
      return [          ] ;
   }
}
```

next	1	boolean	Iterator	this == d2
0	String	-1	compareTo	Dato
@Override	toString	Comparable	int	d1 - d2

Maks poeng: 9

5 Fyll inn riktig kode slik at dette blir et korrekt og fornuftig klassehierarki:

Hjelp

```
interface KanSlette {
    void slett ( ) ;
}
interface KanLese {
   String les ( ) ;
}
interface KanSkrive [          ] KanLese {
   void nyVerdi ( String v ) ;
}

abstract class LeseSkriveMinne [          ] KanSkrive, [          ] {
   String verdi;
   @Override
   public void [          ] ( String v ) { [          ] ; }
   @Override
   public void [          ] ( ) { [          ] ; }
}
class LeseMinne [          ] KanLese {
  String verdi;
  LeseMinne ( String v ) { [          ] ; }
  @Override
  public String [          ] ( ) { return [          ] ; }
}
```

slett	verdi	les	verdi = v	verdi = null	v
extends	nyVerdi	KanSlette	v = verdi	implements	KanLese

Maks poeng: 11

6 Hvilke fem tall blir skrevet ut?

(I denne oppgaven er det ikke brukt @Override fordi du skal selv bestemme om metoder er polymorfe.)

- [ ]
- [ ]
- [ ]
- [ ]
- [ ]

Maks poeng: 10

```java
class Kjæledyr {
    // Kjæledyr veier vanligvis 1000 gram:
    protected int vekt = 1000;
    public void spis (int mengde) { vekt += mengde; }
    public void skriv () { System.out.println(vekt); }
}

class Gullfisk extends Kjæledyr{
    public void spis () { vekt = vekt + 1; }
}

class Hamster extends Kjæledyr {
    public void spis () { vekt = vekt + 2; }
    public void spis (int mengde) {
        super.spis(mengde/2); // Hamstere søler mye!
        spis();
    }
}

class SjekkKjæledyrene {
    public static void main (String[] args) {
        Gullfisk g   = new Gullfisk();
        Hamster h    = new Hamster();
        Kjæledyr kd1 = new Kjæledyr();
        Kjæledyr kd2 = h;
        Kjæledyr kd3 = g;

        h.spis(100);
        h.spis();
        g.spis();
        kd1.spis(50);
        kd2.spis(100);
        kd3.spis(10);

        h.skriv();  g.skriv();
        kd1.skriv();  kd2.skriv();  kd3.skriv();
    }
}
```

7 Hvilke tall blir skrevet ut?

Hint: Legg merke til at variabelen *total* er deklarert som **static**.

min: ☐ og ☐

din: ☐ og ☐

sin: ☐ og ☐

Maks poeng: 9

```java
class Teller {
    static private int total = 0;
    private int lokalAnt = 0;

    public void opp (int ant) {
        lokalAnt += ant;  total += ant;
    }
    public void opp () {
        total++;  lokalAnt++;
    }
    public void skriv (String id) {
        System.out.println(id + ": " + total + " og " + lokalAnt);
    }
}

class SjekkTeller {
    public static void main (String[] args) {
        Teller min    = new Teller();
        Teller din    = new Teller();
        Teller sin    = new Teller();
        Teller oles   = min;
        Teller annes  = sin;

        min.opp(5);  oles.opp(5);
        din.opp(2);  din.opp();
        sin.opp(2);  sin.opp(1);
        annes.opp(3);  annes.opp();

        min.skriv("min");  din.skriv("din");  sin.skriv("sin");
    }
}
```

8 Angi om det er arv (med klasser og subklasser) eller komposisjon som er best egnet til å implementere disse relasjonene.

(I denne oppgaven får du poeng for riktig svar, minuspoeng for galt svar og 0 poeng om du ikke svarer. Du vil imidlertid aldri få mindre enn 0 poeng totalt.)

Velg hva som er best egnet:

	Komposisjon	Arv
Studieprogram - institutt	○	○
Innleveringsoppgave - frivillig innleveringsoppgave	○	○
Kurs - student	○	○
Kurs - innleveringsoppgave	○	○
Innleveringsoppgave - obligatorisk innleveringsoppgave	○	○
Kurs - masterkurs	○	○
Kurs - programmeringskurs	○	○
Kurs - foreleser	○	○
Studieprogram - kurs	○	○
Innleveringsoppgave - vedlegg	○	○

Maks poeng: 8

9 Sett inn kodebiter slik at metoden *settInn* setter inn et Avtale-objekt etter prioritet i en enkeltkjedet liste.

Legg merke til at objektene som lenkes sammen, ikke er Node-objekter, men objekter av en subklasse av klassen Avtale som har en neste-peker.

Hjelp

```
abstract class Avtale implements Comparable <Avtale> {
    Avtale  neste = null;
}

class Avtalebok {
   Avtale første = null;

   void settInn (Avtale a) throws IllegalArgumentException {
      if (a == null)
            [          ]  new  [          ] ( ) ;
      a.neste = null;
      if (første == null) {
            [          ] ;
      } else if (a.compareTo(første) < 0) {
         // Sett inn først:
         a.neste = første;  [          ] ;
      } else {
         Avtale  [          ] ;
         while (pek.neste != null && a.compareTo(pek.neste) > 0) {
               [          ] ;
         }
         //  Sett inn på slutten eller inne i listen; samme kode i begge tilfeller:

         [          ] ;
         [          ] ;
      }
   }
}
```

a = første	pek = a	FileNotFoundException
pek = pek.neste	pek = første	pek.neste = a;
pek.neste = første	første = a	pek.neste = pek
a = pek	a.neste = pek.neste	pek = første.neste

catch | throw | IllegalArgumentException | throws

Maks poeng: 13.6

10 Hvilke tilordninger er tillatt der det er markert ***/* Tilordning */*** ?

(I denne oppgaven vil du få poeng for rett svar, minuspoeng for galt svar og 0 poeng om du ikke svarer. Du vil imidlertid aldri få mindre enn 0 poeng totalt.)

Angi om tilordning er lovlig eller ikke.

	Ikke tillatt	Tillatt
varA = (A) pBI;	○	○
varI = pB_BI;	○	○
varB = pA_B;	○	○
varB = pA;	○	○
varA = (I) pBI;	○	○
varB = pB_BI;	○	○
varA = pB;	○	○
varB = (B) pA_B;	○	○
varI = pBI;	○	○
varI = (I) pB_BI;	○	○
varA = pA_B;	○	○
varB = pBI;	○	○
varA = pA;	○	○

Maks poeng: 14.3

```java
class A { }
class B extends A { }

interface I { }
class BI extends B implements I { }

class Tilordning {
    public static void main (String[] args) {
        A pA = new A();
        A pA_B = new B();
        B pB = new B();
        B pB_BI = new BI();
        BI pBI = new BI();

        A varA;
        B varB;
        I varI;

        /* Tilordning */
    }
}
```