UNIVERSITETET I OSLO
Det matematisk-naturvitenskapelige fakultet

Skriftlig midtveiseksamen i IN1010
2025 vår
Varighet: 17. mars kl 9:00 til 11:00

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 kan få minuspoeng for galt svar, men du vil aldri få mindre enn 0 poeng for noen oppgave.

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

1 Hvilke 8 tall skrives ut?

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

Maks poeng: 6

```java
class ArrayFyll {
  public static void main (String[] args) {
    int[] tall = new int[8];
    tall[0] = 0;
    tall[1] = 1;
    for (int i=2; i < 8; i++) {
        tall[i] = tall[i-2] + tall[i-1];
    }
    for (int t: tall) {
      System.out.println(t);
    }
  }
}
```

2 Når er det naturlig å bruke arv og når er det bedre med komposisjon når det gjelder disse begrepene?

	Arv	Komposisjon
Datamaskintastatur - tast	○	○
Datamaskin - student	○	○
Operativsystem - Windows	○	○
Datamaskin - stasjonær maskin	○	○
Datamaskin - prosessor	○	○
Datamaskin - datamus	○	○
Datamaskin - datasal	○	○
Forelesning - foreleser	○	○
Datamaskin - operativsystem	○	○
ESC-tast - tast	○	○

Maks poeng: 4

3

Dette programmet skal åpne en fil som heter **data.txt** og skrive ut den første linjen i filen. Sett inn det som mangler.

Hjelp

```
Import java.io.File;
import [ ] ;
class LesFil {
   public static void main (String[] args) {
      Scanner s = null;
      [ ] {
         s = new [ ] ( new [ ] (
      [ ] ));
      } [ ] (Exception e) {
         System.out.println("Kan ikke lese data.txt");
         System. [ ] (1);
      }
      System.out.println( [ ] );
   }
}
```

"data.txt"	args	break
catch	exit	File
FileNotFoundException	finally	java.util.Scanner
s.hasNextLine()	s.next()	s.nextLine()
Scanner	String	try

Maks poeng: 6

4 Dette lille programmet skal skrive ut "Voff-voff".
Sett inn de linjene som mangler for å få til dette.

Hjelp

```
class Hundeglam {
   public static void main (String[] args) {
        [          ] ;
       h1 = [          ] ;
       h2 = [          ] ;
       h2 = h1;
       h1 = null;
        [          ] ;
        [          ] ;
   }
}

class Hund  {
   private String voff;
   Hund(String bjeff) { [          ] ; }
   public void bjeff () { System.out.println( [          ] ) ; }
}
```

this.bjeff = bjeff	Hund h1,h2	h3 = h2;
h1.bjeff()	bjeff = voff	System.out.println(bjeff);
voff = bjeff	bjeff	voff
h3.bjeff();	System.out.println(voff)	new Hund("Bjeff-bjeff");
Hund h1, h2, h3	Hund h3	new Hund("Voff-voff")

Maks poeng: 9.03

5 Hva skriver dette programmet ut?

Hjelp

Ole eier

Per eier

Anne eier

Linda eier

Trine eier

EL66654 EB65432 EK34561 DL23145

Maks poeng: 9

```java
class Bileiere {
    public static void main(String[] a) {
        Bil bentley = new Bil("EK34561");
        Bil toyota = new Bil("EB65432");
        Bil subaru = new Bil("EL66654");
        Bil audi = new Bil("DL23145");

        Bileier ole = new Bileier("Ole");
        Bileier per = new Bileier("Per");
        Bileier anne = new Bileier("Anne");
        Bileier linda = new Bileier("Linda");
        Bileier trine = new Bileier("Trine");

        ole.leggTilB(bentley);
        per.leggTilB(toyota);
        anne.leggTilB(subaru);
        linda.leggTilB(audi);
        trine.leggTilB(audi);

        ole.b(anne);
        linda.b(per);
        trine.b(linda);

        // Hva blir skrevet ut her:
        ole.skriv();
        per.skriv();
        anne.skriv();
        linda.skriv();
        trine.skriv();
    }
}
class Bil {
    String regNr;
    Bil(String n) {regNr = n;}
}
class Bileier {
    Bil minB; String navn;
    Bileier(String n) {navn = n;}
    public void leggTilB(Bil b) {minB = b;}
    public void b(Bileier andre) {
        Bil taVare = andre.minB;
        andre.minB = minB;
        minB = taVare;
    }
    public void skriv() {
        System.out.println(navn+" eier "+minB.regNr);
    }
}
```

6 Her er en klasse som implementerer en generisk lenkeliste. Fire av linjene inneholder feil; marker de linjene.

Velg ett eller flere alternativer:

- [ ] class EnkelListe<T> {
- [ ] private Node start = new Node();
- [ ] private class Node {
- [ ] Node neste = null;
- [ ] T verdi;
- [ ] Node (int v) {
- [ ] verdi = v;
- [ ] }
- [ ] }
- [ ] void settInnFørst (T x) {
- [ ] Node p = new T(x);
- [ ] p.neste = start;
- [ ] start = p.neste;
- [ ] }
- [ ] }

Maks poeng: 8

7 Hva skriver dette programmet ut?

Hint: Husk at f.eks. "Stein".compareTo("Dag") > 0

Hjelp

Kanin 7: [ ]

Kanin 8: [ ]

Kanin 9: [ ]

Kanin 10: [ ]

Kanin 11: [ ]

Kanin 12: [ ]

Carsten	Puck	Anne	Kine	Jabben	Eva	Kølle

Maks poeng: 12

```java
abstract class Kanin implements Comparable<Kanin>{
    final String navn; protected int vekt; protected int alder;
    Kanin(String nv, int v, int a) {navn=nv; vekt=v; alder=a;}
    @Override
    public int compareTo(Kanin med) {return vekt - med.vekt;}
    public static Kanin største(Kanin en, Kanin to) {
        if  (en.compareTo(to) > 0) return en; else return to;
    }
}
class Barnekanin extends Kanin {
    Barnekanin(String nv, int v, int a) {super(nv,v,a);}
    @Override
    public int compareTo(Kanin med){return alder-med.alder;}
}
class Hunnkanin extends Kanin {
    Hunnkanin(String nv, int v, int a) {super(nv,v,a);}
    @Override
    public int compareTo(Kanin med){return navn.compareTo(med.navn);}
}
class Hannkanin extends Kanin {
    Hannkanin(String nv, int v, int a) {super(nv,v,a);}
}
class HovedSammenlign {
    public static void main (String[] args) {
        Kanin en = new Barnekanin("Puck", 6, 2);
        Kanin to =  new Barnekanin("Kølle", 5, 3);
        Kanin tre = new Hannkanin("Carsten", 52, 40);
        Kanin fire = new Hannkanin("Jabben", 41, 51);
        Kanin fem = new Hunnkanin("Kine", 36, 32);
        Kanin seks = new Hunnkanin("Anne", 48, 28);
        Kanin syv,åtte,ni,ti,elleve,tolv;
        syv = Kanin.største(en,to);
        åtte = Kanin.største(tre,fire);
        ni = Kanin.største(fem,seks);
        ti = tre.største(en,tre);
        elleve = ti.største(fire,fem);
        tolv = Hunnkanin.største(seks,new Hunnkanin("Eva", 32, 27));

        System.out.println("Kanin 7: " + syv.navn);
        System.out.println("Kanin 8: " + åtte.navn);
        System.out.println("Kanin 9: " + ni.navn);
        System.out.println("Kanin 10: " + ti.navn);
        System.out.println("Kanin 11: " + elleve.navn);
        System.out.println("Kanin 12: " + tolv.navn);
    }
}
```

8 Hvilke tall blir skrevet ut? Legg merke til at *as* og *bs* er deklarert som **static**.

a1.as er ☐

a1.ai er ☐

a2.as er ☐

a2.ai er ☐

b2.as er ☐

b2.ai er ☐

b2.bs er ☐

b2.bi er ☐

Maks poeng: 10

```java
class Statiske {
    public static void main (String[] args) {
        A a1 = new A();
        A a2 = new B();
        a1.tell();  a2.tell();  a2.tell();
        B b2 = (B)a2;
        System.out.println("a1.as er " + a1.as);
        System.out.println("a1.ai er " + a1.ai);
        System.out.println("a2.as er " + a2.as);
        System.out.println("a2.ai er " + a2.ai);
        System.out.println("b2.as er " + b2.as);
        System.out.println("b2.ai er " + b2.ai);
        System.out.println("b2.bs er " + b2.bs);
        System.out.println("b2.bi er " + b2.bi);
    }
}
class A {
    static int as = 0;
    int ai = 0;
    void tell () { as++;  ai++; }
}
class B extends A {
    static int bs = 0;
    int bi = 0;
    void tell () { super.tell();  bs++;  bi++; }
}
```

9 Klassen Bil definerer objekter som kan settes inn i en dobbeltlenket liste. I dette programmet bruker vi ikke Node-objekter for å holde orden på listen, men lar forrige- og neste-pekerne ligge i selve Bil-objektet.

Klassen Billiste definerer den dobbeltlenkede listen med en første-peker og en siste-peker. Metoden settInnFør setter første parameter inn i listen foran andre parameter. De ni kodebitene skal alle brukes.

Hjelp

```
class Bil {
   Bil forrige;
   [                          ] ;
}
class Billiste {
    [                          ] ;
     public void settInnFør( [                          ] );
        if( første == null) { // Listen er tom
              [                          ] ;
        }  else {.  // Først, gjør det en alltid må (og uten å miste noe):
              [                          ] ;
             if ( første == erILista )  {  // Sett inn først
                  [                          ] ;
                  [                          ] ;
              } else { // Sett inn inne i mellom
                    [                          ] ;
                    erIlista.forrige.neste = ny;
                    [                          ] ;
              }
          }
      }
}
```

erILista.forrige = ny	ny.neste = erILista	første = ny
første = ny; siste = ny;	ny.forrige = erILista.forrige	Bil neste
Bil første, siste	Bil ny, Bil erILista	første.forrige = ny

Maks poeng: 12.96

10 Hvilke av disse linjene er det lov å sette nederst i programmet (der posisjonen er markert med en kommentar)? Etter innsettingen skal man kunne kompilere og kjøre koden uten at feil oppstår.

Angi om setningen er lovlig:

	Lovlig	Ikke lovlig
b = (D) ii;	○	○
a = (E) jj;	○	○
c = (C) b;	○	○
a = ii;	○	○
a = (B) e;	○	○
c = (C) ii;	○	○
ii = e;	○	○
a = (B) ii;	○	○
b = d;	○	○
a = (B) jj;	○	○
ii = jj;	○	○

Maks poeng: 11

```
interface G { }
class A { }
class B extends A { }
class C extends B { }
class D extends B implements G { }
class E extends A implements G { }

class HovedSubInt {
  public static void main (String[] args) {
      A a = new B();
      B b = new C();
      C c = new C();
      D d = new D();
      E e = new E();
      G ii = d;
      G jj = e;
      // Her skal det stå én linje
  }
}
```

11

Hva skriver programmet ut?

Pris 1: ☐

Pris 2: ☐

Pris 3: ☐

Pris 4: ☐

Pris 5: ☐

Lengde 1: ☐

Lengde 2: ☐

Lengde 3: ☐

Maks poeng: 12

```java
class Flybilletter {
  public static void main(String[] args) {
     Business bil1B = new Business(1000);
     Premium bil1P = new Premium(500);
     Premium bil2P = new Premium(200,30);
     Økonomi bil1Ø = new Økonomi(100);
     Økonomi bil2Ø = new Økonomi(100,10);

     System.out.println("Pris 1: " + bil1B.pris());
     System.out.println("Pris 2: " + bil1P.pris());
     System.out.println("Pris 3: " + bil2P.pris());
     System.out.println("Pris 4: " + bil1Ø.pris());
     System.out.println("Pris 5: " + bil2Ø.pris());
     System.out.println("Lengde 1: " + bil1B.centimeter());
     System.out.println("Lengde 2: " + bil1P.centimeter());
     System.out.println("Lengde 3: " + bil1Ø.centimeter());
  }
}
interface  MerBenplass {int centimeter();}

abstract class FlyBillett implements MerBenplass {
    protected int basispris;
    protected int cm = 0;
    FlyBillett(int p, int c) {basispris = p; cm = c;}
    public int pris() {return basispris+centimeter()*100;}
}
class Økonomi extends FlyBillett {
    Økonomi(int pris, int c) {super(pris,c);}
    Økonomi(int pris) {super(pris, 0);}
    @Override public int centimeter() {return cm;}
}
class Premium extends Økonomi {
    Premium(int pris) {super(pris,20);}
    Premium(int pris, int c) {super(pris,c);}
}
class Business extends Premium {
    Business(int pris) {super(pris,10);}
    @Override public int pris(){return super.pris()*4;}
}
```