UNIVERSITETET I OSLO
Det matematisk-naturvitenskapelige fakultet

Prøveeksamen for midtveiseksamen i IN1010 2026 vår
Tid: 10. mars kl 16:00 til 18. mars kl 23:59

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 0 poeng for galt svar.

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

NB om prøveeksamen: Så lenge du ikke levererer inn besvarelsen, kan du gjøre så mange forsøk som du vil.

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 Her er to metoder:

settInnIArray setter verdien **v** inn i arrayen **a** i posisjonen med indeks **ix**, men bare hvis **ix** er en lovlig indeks.

hentFraArray henter ut verdien lagret i arrayen **a** i posisjonen med indeks **ix**, men hvis **ix** er mindre enn minste lovlige indeks, skal metoden returnere arrayelementet med lavest lovlig indeks; tilsvarende, hvis **ix** er større enn største lovlige indeks, skal metoden returnere elementet med høyest lovlig indeks.

Sett inn kodebitene på riktig plass.

Hjelp

```
public static void [          ] ( int [ ] a, int ix, int v ) {
    if ( [          ] ) {
        [          ] ;
    }
}

public static int [          ] (int [ ] a, int ix) {
    if (ix < 0) {
        return a [ [          ] ];
    } else if (ix >= [          ] ) {
        return [          ] ;
    } else {
        return [          ] ;
    }
}
```

a[ix] = v	ix >= 0 && ix < a.length	a.length
ix >= 0 || ix < a.length	a[a.length-1]	settInnIArray
a.legalIndex(ix)	a[a.length]	a[ix]++
a[v] = ix	hentFraArray	a.length()
a[ix]	1	0 <= ix < a.length
0		

Maks poeng: 10

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: 8

4 Hvor er det naturlig å bruke arv og når er det naturlig med komposisjon i disse relasjonene?

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

	Komposisjon	Arv
Person – student	○	○
Bakeri – brød	○	○
Bok - roman	○	○
Ulv – rovdyr	○	○
Bil – hjul	○	○
Ulv – skog	○	○
Loff – brød	○	○
Bibliotek – bok	○	○

Maks poeng: 4

5 Gitt følgende skjelett-program som ligger i en fil med navn A.java:

```
public class A {
    public static void main(String[] args) {
        <kodelinje>
    }
}
```

For hver av kodelinjene under, avgjør om programmet over vil kompilere og/eller kjøre (altså terminerer uten at noen unntak eller feil oppstår) dersom <kodelinje> byttes ut med den gitte kodelinjen. Programmet blir kompilert med

```
javac A.java
```

og kjørt med

```
java A hallo
```

Vil programmet kompilere og kjøre, kun kompilere eller ingen av delene?

	Kompilere og kjøre	Kun kompilere	Ikke kompilere
String s = args[5];	○	○	○
int i = 1 / (args.length - 1);	○	○	○
System.out.println(args[0]);	○	○	○
String s = new Object();	○	○	○
System.out.println(this.toString());	○	○	○
return;	○	○	○
Object s = "Hei!";	○	○	○
int i = null;	○	○	○
return 42;	○	○	○
A a = new A();	○	○	○

Maks poeng: 12

6

```
class Hunder {
    public static void main (String[] a) {
        Hund passopp = new Hund("Passopp");
        Hund trofast = new Hund("Trofast");
        Hund sniff = new Hund("Sniff");
        Hund snapp = new Hund("Ekorn");

        Hund bella, balder, min, din, lassie, hans, hennes, petters;

        min = snapp;
        din = passopp;
        petters = din;
        hans = min;
        din = snapp;
        hennes = trofast;
        trofast = sniff;
        lassie = hennes;
        petters = din;
        bella = lassie;

        System.out.println("Første navn: " + hans.navn);
        System.out.println("Andre navn: " + hennes.navn);
        System.out.println("Tredje navn: " + din.navn);
        System.out.println("Fjerde navn: " + bella.navn);
    }
}

class Hund {
    String navn;
    Hund(String n) { navn = n; }
}
```

Hvilke navn skrives ut av dette programmet?

	Passopp	Trofast	Sniff	Ekorn
Første navn:	○	○	○	○
Andre navn:	○	○	○	○
Tredje navn:	○	○	○	○
Fjerde navn:	○	○	○	○

Maks poeng: 12

7 Klassen EnVeisBeholder kan ta vare på objekter av subklasser til klassen EnVeis i en enkeltkjedet liste I denne oppgaven skal vi ikke benytte Node-klassen – i stedet skal klassen EnVeis inneholde en nestepeker.

EnveisBeholder skal ha to metoder: settInnSist og taUtFørst.

Fyll inn de ti kodebitene på riktig sted.

Hjelp

```
abstract class  EnVeis {
    [          ] ;
}
class  EnVeisBeholder < [          ] > {
    [          ]
    public void settInnSist ( [          ] ) {
      if (første == null) { [          ] }
       else { [          ] }
  }
  public E taUtFørst ( ) {
      if (første == null)  { [          ] }
      E ut = første;
      if (første == siste) { [          ] }
      else { [          ] }
        [          ]
   }
}
```

første = (E) første.neste;	E extends EnVeis	siste.neste = ny; siste = ny;
første = ny; siste = ny;	første = null; siste = null;	E ny
return null;	EnVeis neste	return ut;
private E første, siste;		

Maks poeng: 10

8 Dette er en klasse **Lager** som kan inneholde 10 String-er. Den skal utstyres med en iterator som går gjennom arrayens innhold *baklengs*, dvs fra indeks 9 og ned til indeks 0.
(Klassen inneholder også diverse metoder, men de er vi ikke interesserte i denne gangen.)

Hjelp

```
class Lager implements [          ] {
   String[] innhold = new String[10];

   @Override public [          ] [          ] () {
      return [          ] ;
   }

   class LagerIterator implements [          ] {
      int ix = [          ] ;
      @Override public boolean hasNext () {
         return [          ] ;
      }
      @Override public String next () {
         String res = [          ] ;
          [          ] ;
         return res;
      }
   }
}
```

Iterator <String>	iterator	ix >= 0	innhold[ix]
innhold.length - 1	Iterable <String>	ix = ix - 1	new LagerIterator ()

Maks poeng: 13.5

9

```java
class TestBoeker {
    public static void main (String[] args) {
        Bok bok1 = new Bok("Snorres kongesagaer", 1000);
        System.out.println("Snorre internpris " + bok1.internpris());
        Bok bok2 = new Laerebok("Linux for dummies", 400);
        System.out.println("Linux internpris " + bok2.internpris());
        Bok bok3 = new IN1010bok("Big Java", 1200);
        System.out.println("Big Java " + bok3.internpris());
        Bok bok4 = new Roman("Krig og fred", 2000);
        System.out.println("Krig og fred " + bok4.internpris());
        Bok bok5 = new Krimbok("Sherlock Holmes", 240);
        System.out.println("Sherlock Holmes " + bok5.internpris());
    }
}

class Bok {
    String tittel;  double pris;
    Bok (String t, double p) { tittel = t;  pris = p; }
    double internpris () { return pris; }
}

class Laerebok extends Bok {
    Laerebok (String t, double p) { super(t, p); }
    @Override double internpris () { return pris * 0.8; }
}

class IN1010bok extends Laerebok {
    IN1010bok (String t, double p) { super(t,p); }
    @Override double internpris () { return super.internpris() * 0.5; }
}

class Roman extends Bok {
    Roman (String t, double p) { super(t, p); }
}

class Krimbok extends Roman {
    Krimbok (String t, double p) { super(t,p); }
    @Override double internpris () { return pris * 0.75; }
}
```

Dette programmet skriver ut 5 verdier. Hvilke verdier skrives ut?

Snorre internpris

Linux internpris

Big Java

Krig og fred

Sherlock Holmes 

Maks poeng: 10

10 Hvilke 8 tall skriver dette programmet ut? Du får ekstrapoeng om du klarer alle riktige. Pass på at én variabel er deklarert "static".

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

Maks poeng: 15

```java
class A {
   protected static int ant = 1;
   protected int a = 2;
   public int resultat() {return a;}
}
class B extends A {
   protected int b = 3;
   @Override
   public int resultat() {return a+b;}
}
class C extends B {
   @Override
   public int resultat() {return a+b+ant;}
   public int resultat(int d) {
      ant = ant+d;
      return resultat();
   }
}
class D extends C {
   @Override
   public int resultat(int d) {
      return resultat()+super.resultat(d);
   }
}
class E extends B {  }

class HovedPoly {
  public static void main (String[] args) {
     A a = new A();
     B b = new B();
     C c = new C();
     D d = new D();
     E e = new E();
     A aa = c;

     System.out.println(a.resultat());
     System.out.println(b.resultat());
     System.out.println(c.resultat());
     System.out.println(d.resultat());
     System.out.println(e.resultat());
     System.out.println(c.resultat(4));
     System.out.println(d.resultat(5));
     System.out.println(aa.resultat());
  }
}
```