UNIVERSITETET I OSLO
Det matematisk-naturvitenskapelige fakultet

Skriftlig utsatt og ny prøve i IN1010
2025 vår
Varighet: 9. april 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, minuspoeng for galt svar og 0 poeng for ubesvart. Du vil aldri få mindre enn 0 poeng for noen oppgave.

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

1 Velg tre linjer med kode slik at denne metoden setter inn en ny verdi (gitt som tredje parameter) på posisjon pos (gitt som andre parameter) i arrayen tall (første parameter).

Metoden skal returnere det som sto i posisjon pos opprinnelig.

Hjelp

```
public static int settInnOgReturner (int [ ] tall, int pos, int ny) {
        [ ]
        [ ]
        [ ]
}
```

return tall[pos];	int taVare = tall;	return pos;	int taVare;	tall[pos] = taVare;
int taVare = tall[pos];	tall[ny] = taVare;	return taVare;	return ny;	tall[pos] = ny;
taVare = pos;	int taVare = pos;	taVare = ny;		

Maks poeng: 4.5

2 Her er en klasse Bil som kan settes inn i en enkelkjedet liste, og en klasse BilListe som opprettes med en første-peker og en siste-peker som angir en liste med to biler. Metoden settInnBak skal sette inn en ny bil bak den som parameteren erILista peker på. En kan anta at erILista ikke peker på siste Bil i lista (med endre ord at erILista.neste aldri har verdien null); derfor skal pekeren siste aldri oppdateres. Sett inn riktige kodebiter på riktige plasser.

Hjelp

```
class Bil {
  [          ] ;
}
class BilListe {
    Bil første = new Bil();
    [          ] ;
     BilListe() {
        [          ] ;
    }
    public void settInnBak(Bil erILista, [          ] ) {
        [          ] ;
        [          ] ;
    }
}
```

erILista.neste = ny.neste	Bil siste = new Bil()	første.neste = siste
erILista.neste = ny	Bil ny	Bil neste
ny.neste = erILista.neste	ny.neste = siste	

Maks poeng: 9

3 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: 10

```
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());
  }
}
```

4 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
int i = 1 / 0;	○	○	○
System.out.println(this.toString());	○	○	○
return 42;	○	○	○
String s = args[5];	○	○	○
A a = new A();	○	○	○
System.out.println(args[0]);	○	○	○
Object s = "Hei!";	○	○	○
int i = null;	○	○	○
return;	○	○	○
String s = new Object();	○	○	○

Maks poeng: 10

5 Lag en klasse som har en statisk metode som leser tekststrenger skrevet inn på tastaturet av brukeren og legger dem i en liste. Metoden skal returnere listen når brukeren gir den tomme tekststrengen som input. (Altså skal siste element i den returnerte listen alltid være den tomme tekststrengen.)

Hjelp

```
import java.util.*;

class StrengInput {
    public [          ] hent () {
        Scanner in = [          ] ;
            [          ] strengListe = new
        [          ] ;
            [          ] ;
            [          ]
            [          ]
            [          ]
        }
            [          ]
    }
}
```

String[]	strengListe = brukerStreng;
LinkedList<String>()	List<String>
for (brukerStreng: strengListe) {	strengListe = in.hasNextLine();
brukerStreng = in.nextLine();	static void
strengListe += brukerStreng;	while (! brukerStreng.equals("")) {
return strengListe;	new Scanner(System.in)
String brukerStreng = "IkkeTom"	return brukerStreng;
new Scanner(new File("data.txt"))	static String

strengListe.add(brukerStreng);

static List<String>

Maks poeng: 10.53

new String[]

6 Her er to metoder:

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

hentFraArray henter ut verdien lagret i elementet i arrayen **a** 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 [          ] ;
    }
}
```

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

Maks poeng: 8

7 Klassen EnVeisBeholder kan ta vare på objekter av subklasser til klassen EnVeis i en enkeltkjedet liste. 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 {  [          ]  }
        [          ]
   }
}
```

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

Maks poeng: 10

8 I dette programmet skal det settes inn én linje der det er markert med en kommentar. For hver av mulighetene under skal du markere om den kan benyttes og vil fungere uten feil, eller om den vil gi kompilerings- eller kjørefeil.

	OK	Vil gi feil
b2 = (C) z1;	○	○
x2 = (Intf1) z1;	○	○
y2 = (Intf2) a1;	○	○
a3 = (B) a1;	○	○
c1 = (C) y1;	○	○
x2 = c;	○	○
x2 = y1;	○	○
y2 = c;	○	○
z2 = x1;	○	○
x2 = a1;	○	○
y2 = (Intf2) a2;	○	○
z2 = (C) a1;	○	○
z2 = y1;	○	○
y2 = z1;	○	○
z2 = (Intf3) b1;	○	○

Maks poeng: 11.25

```
class A { }
class B extends A implements Intf1 { }
class C extends B implements Intf3 { }
interface Intf1 { }
interface Intf2 { }
interface Intf3 extends Intf2 { }

class HovedCast{
    public static void main (String[] args) {
        A a = new A();
        B b = new B();
        C c = new C();
        A a1,a2,a3;  B b1,b2;  C c1;
        Intf1 x1,x2;  Intf2 y1,y2;  Intf3 z1,z2;

        a1 = b;  a2 = c;  b1 = c;
        x1 = b;  y1 = c;  z1 = c;

        // Her skal det stå én linje
    }
}
```

9 Dette programmet skal skrive ut de to linjene
Gamlefela
Litago
Sett inn kodebitene på riktig plass for å få til dette.

Hjelp

```
class Spellemann {
   public static void main (String[ ] args) {
        Ku litago = new Ku ("Litago");
        Fele  gamlefela =  new Fele ("Gamlefela");
       Mann per = new Mann(gamlefela);  Mann ole = new Mann(litago);
       per.byttebort(ole);   ole.byttebort(per);
       per.skriv();    ole.skriv();
  }
}
interface KanEies { void skriv(); }
class Verdi implements KanEies {
    String navn;
     [                                ]
     public void [                                ]
}
class Ku  extends Verdi { [                                ] }
class Fele extends Verdi { [                                ] }
class Mann {
     [                                ] ;
     Mann ( KanEies start) { [                                ] ;}
     public void skriv ( ) { [                                ] ;}
     public void byttebort (Mann andre) {
          [                                ] ;
          [                                ] ;
          [                                ] ;
     }
}
```

min = andre.min	min.skriv()
skriv() {System.out.println(navn);}	Ku(String n) {super(n);}
Fele(String n) {super(n);}	andre.min = temp

KanEies temp = min	KanEies min
Verdi(String n) {navn = n;}	min = start

Maks poeng: 10

10

Bør disse begrepsparene implementeres med arv eller med komposisjon?

	Arv	Komposisjon
Pølse - matvare	○	○
Butikkansatt - butikk	○	○
Butikkansatt - ekspeditør	○	○
Brød - loff	○	○
Butikk - matvarebutikk	○	○
Matvarebutikk - matvare	○	○
Brød - brødpose	○	○
Pølse - medisterpølse	○	○
Egg - eggkartong	○	○
Butikk - butikkjede	○	○

Maks poeng: 5

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

Maks poeng: 11.7