UNIVERSITETET I OSLO
Det matematisk-naturvitenskapelige fakultet

Ny og utsatt skriftlig midtveiseksamen i IN1010
2024 vår
Varighet: 19. april kl 9:00–11:00

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. Hvis det ikke står noe annet, vil du få poeng for hvert riktige svar og 0 poeng for galt.

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) {
    [ ]
    [ ]
    [ ]
}
```

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

Maks poeng: 6

2 Dette lille programmet skal opprette et objekt av klassen **Brev** og skrive ut dets vekt; svaret skal være

Brevets vekt: 19.

Sett inn de kodebitene som mangler.

Hjelp

```
class EnkelProgrammering {
   public static void main (String[] args) {
      Brev b1, b2;
      b1 =                          ;
                                    ;
      System.out.println("Brevets vekt: " +                          ;
   }
}

class Brev  {
   private int vekt;
   Brev(int v) {                    ; }
   int finnVekt () { return                    ; }
}
```

b2 = new Brev(18);	b2 = b1	vekt = this.v	vekt
v = vekt	vekt = v	neste.finnVekt()	b2.vekt
b1.vekt	new Brev(19)	new Brev()	this.v = vekt
finnVekt(b2)	b2	b2.finnVekt()	b1 = b2

Maks poeng: 6

3 Dette lille programmet skal lese inntil 10 positive heltall. Et negativt tall signaliserer slutten på innlesingen hvis det er færre enn ti tall. Etterpå skal tallene skrives ut i omvendt rekkefølge.

Hjelp

```
import java.util.Scanner;
class SkrivTall {
    public static void main (String[] args) {
        int[] tallene = [          ] ;
        int antallTall = [          ] ;
        Scanner tastatur = new Scanner(System.in);
        while ( [          ] ) {
            int v = [          ] ;
            if ( [          ] ) break;
            [          ] = [          ] ;
            [          ] ;
        } // end while-loop
        for ( [          ] ) {
            [          ] ( [          ] );
        } // end for-loop
    }
}
```

antallTall = antallTall + 1	int i = antallTall-1; i >= 0; i--
tastatur.println	null
10	v
0	antallTall < 10
-1	tallene[antallTall]
antallTall = antallTall - 1	v >= 0
tastatur.nextInt()	System.out.println
tallene[0]	int i = 0; i < tallene.length; i++

tallene[i]

v < 0

new int[10]

Maks poeng: 11

4

```
class FinnHesteeiere {
    public static void main (String[] args) {
        Hest blakken = new Hest("Blakken");
        Hest trofast = new Hest("Trofast");
        Hest slitern = new Hest("Slitern");
        Hest svarten = new Hest("Svarten");
        Hesteeier ole = new Hesteeier();
        Hesteeier per = new Hesteeier();
        Hesteeier anne = new Hesteeier();
        Hesteeier linda = new Hesteeier();
        Hesteeier trine = new Hesteeier();
        ole.leggTilH(blakken);
        per.leggTilH(trofast);
        anne.leggTilH(slitern);
        linda.leggTilH(svarten);
        trine.leggTilH(svarten);
        ole.byttH(anne);
        linda.byttH(per);
        trine.byttH(linda);
        ole.skriv("Ole");
        per.skriv("Per");
        anne.skriv("Anne");
        linda.skriv("Linda");
        trine.skriv("Trine");
    }
}
class Hest {
    String navn;
    Hest (String n) { navn = n; }
}
class Hesteeier {
    Hest minH;
    public void leggTilH (Hest h) { minH = h; }
    public void byttH (Hesteeier andre) {
        Hest taVare = andre.minH;
        andre.minH = minH;
        minH = taVare;
    }
    public void skriv (String navn) {
        System.out.println(navn + ": " + minH.navn);
    }
}
```

Hva blir skrevet ut? Dra riktig hestenavn til venstre.

Hjelp

Ole: ____

Per: ____

Anne: ____

Linda: ____

Trine:

Trofast

Slitern

Svarten

Blakken

Maks poeng: 10

5 Dette lille programmet skal bruke en rekursiv metode til å beregne summen av verdiene i en array. Fullfør programmet ved å sette kodebitene på plass.

Hjelp

```
class FinnSumTest {
   public static void main (String[] args) {
      int[] data = { 3, 5, 1, -3, 7 };
      int v = finnSum(0, data);
      System.out.println("Summen er " + v);
   }

   static int finnSum (int start, int[] a) {
      if (start == a.length)
         return [          ] ;
      else
         return a[start] + [          ] ;
   }
}
```

finnSum(0,a)	a[0]	1
start+1	finnSum(start,a)	0
finnSum(0,a) + finnSum(1,a)	a[start]	start
start-1	finnSum(start+1,a)	finnSum(a.length,a)

Maks poeng: 7

6 Vi har en klasse *Dobbeltliste* som implementerer en dobbeltlinket liste. Den har i hvert fall disse tre metodene:

- *fjernFørste* fjerner første element i listen og returnerer det. (Vi antar at listen ikke blir tom etterpå.)
- *fjernSiste* fjerner siste element i listen og returnerer det. (Vi antar at listen ikke blir tom etterpå.)
- *hentNr* henter element nr *n* (nummerert fra 0) og returnerer det, men fjerner det ikke. Vi kan anta at elementet finnes.

Flytt de riktige kodebitene på plass. Bitene kan brukes mer enn én gang.

Hjelp

```
class Dobbeltliste<T> {
   class Node {
      Node forrige, neste;
      T data;
   }
   Node første, siste;

   T fjernFørste () {
      Node resP =                    ;
                                     ;
      return                         ;
   }
   T fjernSiste () {
      Node resP =                    ;
                                     ;
      return                         ;
   }
   T hentNr (int n) {
      Node nodeP =                   ;
      for (int i = 0;  i < n;  ++i)  ;
      return                         ;
   }
}
```

første.data	siste	nodeP.data
null	nodeP	første = første.neste
siste.data	resP	siste.forrige

første	resP = resP.neste;	siste = siste.forrige
nodeP = nodeP.neste	resP.data	første.neste

Maks poeng: 9

7 Dette programmet er et enkelt testprogram for en *Kaninbeholder* som skal kunne finne ut hvilken *Kanin* som er den eldste og skrive ut dens alder. Testprogrammet lager en *Kaninbeholder* og fyller den med *Kanin*-er.

Sett inn kodebiter slik at det blir et korrekt program som gjør det det skal.

Hjelp

```
class KaninRekursjonTest {
    public static void main (String[] a) {
        Kaninbeholder kaninbur = new Kaninbeholder();
        Kanin svar = kaninbur.finnEldste();
        System.out.println("Den eldste er " + svar.alder + " år");
    }
}

class Kaninbeholder {
    Kanin første;
    Kaninbeholder () {
        første = new Kanin(12);
        for (int i = 5;  i > 0;  i--) {
            Kanin denne = første;   første = new Kanin(i);   første.neste = denne;
        }
    }

    Kanin finnEldste () { return                    ; }
}

class Kanin {
    Kanin neste;
    int alder;

    Kanin (int a) {                    ; }
    Kanin letEtterEldste () {
        Kanin resultat =                    ;
        if (                    ) {
            Kanin gammel =                    ;
            if (                    ) {                    ; }
        }
        return                    ;
    }
}
```

this	letEtterEldste(neste)	alder = a
null	gammel.alder > alder	neste != null
0	neste.letEtterEldste()	resultat
første.letEtterEldste()	gammel	resultat = gammel
gammel != null	neste == null	

Maks poeng: 12

8

```java
class RingBruk {
    public static void main (String[] a) {
        int[] tallene = { 1, 2, 3, 4, 5, 7, 9, 12, 13, 17, 19, 24, 32 };
        Ring minRing = new Ring(5);
        for (int tall: tallene) minRing.settInn(tall);
        minRing.dump();
    }
}

class Ring  {
    int antall;
    int[] mineTall;

    Ring (int ant) {
        antall = ant;  mineTall = new int[antall];
    }

    int innIndeks = 0;
    // Invariant: innIndeks peker på neste plass å sette inn.
    public void settInn (int inn){
        mineTall[innIndeks] = inn;
        innIndeks++;
        if (innIndeks == antall) innIndeks = 0;
    }

    void dump () {
        for (int ut: mineTall) System.out.println(ut);
    }
}
```

Hvilke fem tall skriver dette programmet ut?
Du får poeng for alle riktige svar og en bonus om alt er rett.

☐ ☐ ☐ ☐ ☐

Maks poeng: 8

9 Vi har en klasse *Bilkø* som implementerer en dobbeltlenket liste av *Bil*-er. Flytt de riktige kodebitene på plass slik at de to metodene gjør det de sier de skal gjøre.

Hjelp

```
class Bilkø {
   Bil første, siste;
   // Invariant: Hvis både første og siste er null: køen er tom.
   // Hvis ikke, peker første på første bil og siste på siste bil i køen.

   void taUtInni (Bil b) {
      // Ta ut bilen b som verken er først eller sist i køen.
      [                                                    ]
      [                                                    ]
   }

   void settInn (Bil b) {
      // Setter bilen b inn sist i køen.
      if ( [                                               ] ) {
         [                                                 ]
      } else {
         [                                                 ]
      }
      [                                                    ]
   }
}

class Bil {
   Bil forrige, neste;
}
```

- b.neste = null; b.forrige = null;
- første == siste
- siste = null;
- b.forrige = siste; siste.neste = b;
- b.forrige.neste = b.neste; b.neste.forrige = b.forrige;
- b.neste.neste = b.neste; b.forrige.forrige = b.forrige;

- første = b;
- siste = b;
- første == null
- første = siste
- b.forrige = første; siste.neste = null;
- b.neste = null; første.neste = null;

Maks poeng: 9

10

```
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

11

```java
class FlaskeBruk {
    public static void main (String[] a) {
        Drikkeflaske drikke = new Drikkeflaske();
        Plastflaske plast = new Plastflaske();
        Miljøplastflaske miljø = new Miljøplastflaske();
        Pappflaske papp = new Pappflaske();
        Glassflaske glass = new Glassflaske();
        PappflaskeMedPant pappPant = new PappflaskeMedPant();

        Flaske f1 = pappPant;
        Plastflaske pl1 = miljø;

        // Hvilke av disse setningene er gyldige som neste setning?
        Pant p1 = plast;
        Pant p2 = pappPant;
        Pant p3 = miljø;
        Pant p4 = (Pant)f1;
        Pant p5 = (Pant)pl1;
        Pant p6 = f1;
        Pant p7 = pl1;
        Papirsøppel ps1 = papp;
        Papirsøppel ps2 = pappPant;
        Papirsøppel ps3 = (Papirsøppel)f1;
        Papirsøppel ps4 = f1;
        Flaske f2 = drikke;
        Flaske f3 = pl1;
        Plastflaske pl2 = miljø;
        Pappflaske pf1 = f1;
    }
}

interface Pant {}
interface Papirsøppel {}

abstract class Flaske {}
class Drikkeflaske extends Flaske {}
class Plastflaske extends Flaske implements Pant {}
class Miljøplastflaske extends Plastflaske {}
class Pappflaske extends Flaske implements Papirsøppel {}
class Glassflaske extends Flaske {}
class PappflaskeMedPant extends Pappflaske implements Pant {}
```

Hvilke av disse setningene

- kan kompileres og kjøres uten feil
- vil gi kompileringsfeil eller feile under kjøringen

Du får poeng for riktig svar og minuspoeng for galt svar, men du aldri få mindre enn 0 poeng totalt.

	OK	Feil
Pant p1 = plast;	○	○
Pant p2 = pappPant;	○	○
Pant p3 = miljø;	○	○
Pant p4 = (Pant)f1;	○	○
Pant p5 = (Pant)pl1;	○	○
Pant p6 = f1;	○	○
Pant p7 = pl1;	○	○
Papirsøppel ps1 = papp;	○	○
Papirsøppel ps2 = pappPant;	○	○
Papirsøppel ps3 = (Papirsøppel)f1;	○	○
Papirsøppel ps4 = f1;	○	○
Flaske f2 = drikke;	○	○
Flaske f3 = pl1;	○	○
Plastflaske pl2 = miljø;	○	○
Pappflaske pf1 = f1;	○	○

Maks poeng: 12