Možnih je 110 točk, 100% = 100 točk.

1. __/ 15 2. ___/ 10 3. ___ / 5 4. ___/5 5. ___/25 6. ___/ 25 7. ___/25 Skupaj: _____ / 105

Pri nalogah 1 do 4 je treba **obkrožiti pravilen odgovor**. Pri vsakem vprašanju je **en** odgovor pravilen, ostali so napačni. Za napačen odgovor dobiš **negativne točke**. Morebitne popravke **jasno** označi. Če si se pri označenem odgovoru premislil in želiš pustiti vprašanje neodgovorjeno, to napiši z **jasnim stavkom**.

1.) Podana sta razreda

```
public class Osnova {

  private int k1;
  int k2;
  public int k3 = 0;

  private static int sA = 1;
  static int sB = 0;
  public static int sC = 1;

  public int get1() { return k1; }

  public static void setA(int ii) {
    sA = ii;
  }

  static void next() {
    sB = sB + sA;  sC = 2 * sC;
  }

  public Osnova roll() {
    int ll = k1;
    k1 = k2;  k2 = k3;  k3 = ll;
    return this;
  }

  public String toString() {
    return k1 + " : " + k2 + " : " + k3;
  }

  public Osnova(int i1, int i2, int i3) {
    k1 = i1;  k2 = i2 * sB;  k3 = i3 + sC;
    next();
  }

  public Osnova(int i1) {
    k1 = i1;  k2 = sB;
    next();
  }

  public static void main(String[] args) {
    sA = 7;  sB = 3;  sC = 1;
    next();
    Osnova pp = new Osnova(1, 10, 7);
    Osnova qq = new Osnova(2);
    Osnova rr = new Osnova(3);
    System.out.println("p = "+pp);
    System.out.println("q = "+qq);
    System.out.println("r = "+rr);
    qq.k3 = pp.k3;
    rr.k3 = qq.roll().k2;
    System.out.println("q' = "+qq);
    System.out.println("r' = "+rr);
    qq.roll();
    System.out.println("q.g1 = "+qq.get1());

  } // main

} // class
```

```
public class Vec extends Osnova {

  private int kv;

  public static int sD = 1;

  static void next() {
    sB = sB + 1;  sC = sC / 2;  sD++;
  }

  public String toString() {
    return super.toString() + " : " + kv;
  }

  public Vec(int ii, int jj) {
    super(ii,ii,ii);
    kv = jj;
    next();
  }

  public Vec(int ii) {
    super(sD);
    kv = ii;
    next();
  }

  public static void main(String[] args) {
    setA(2);
    Vec pp = new Vec(1, 10);
    Vec qq = new Vec(2);
    Vec rr = new Vec(3);
    System.out.println("p = "+pp);
    System.out.println("q = "+qq);
    System.out.println("r = "+rr);
    qq.k3 = pp.k3;
    rr.k3 = qq.roll().k2;
    System.out.println("q' = "+qq);
    System.out.println("r' = "+rr);
    qq.roll();
    System.out.println("q.g1 = "+qq.get1());
    /* ♣ */
  } // main

} // class
```

[5T] Če poženemo program Vec (z ukazom `java Vec`), je **pred**zadnja vrstica izpisa

(a) `r' = 3 : 2 : 2 : 3`
(b) `r' = 3 : 6 : 3 : 3`
☻ `r' = 3 : 6 : 2 : 3`
(d) `r' = 1 : 4 : 2 : 3`

[5T] Ali lahko na mestu označenem z `/* ♣ */` dodamo stavek `k2 = sD;`?

(a) da
☻ ne, ker `k2` tam ne obstaja
(c) ne, ker `k2` tam ni viden
(d) ne, ker niti `sD` niti `k2` tam nista vidna

[5T] Ali lahko telo metode `toString` v razredu `Vec` napišemo takole

```
return k1 + " : " + k2 + " : " + k3 + " : " + kv;
```

(a) da
(b) ne, ker `k1`, `k2` in `k3` tam niso vidni
(c) ne, ker `k1` in `k2` tam nista vidna
☻ ne, ker `k1` tam ni vidna

OPOMBA: Pri testu za DIRI 2003 ta naloga ne pride v poštev, ker bo obravnavana snov, potrebna za uspešno reševanje, še preveč »sveža«.

2.) Dan je javni razred `Point`, ki ga uporabljamo za predstavitev točke v xy.koordinatnem sistemu

```
public class Point
{
  private int myX; // koordinate
  private int myY;

  public Point( ){
    myX = 0;
    myY = 0;
  }

  public Point(int a, int b){
    myX = a;
    myY = b;
  }

  // ... ostale metode niso prikazane
}
```

Iz tega razreda izpeljemo razred `NamedPoint` z namenom, da bi točke lahko tudi poimenovali

```
public class NamedPoint extends Point
{
  private String myName;

  // konstruktorji

  // ... ostale metode niso prikazane
}
```

Naj bodo tole predlagani konstruktorji za ta razred:

I. `public NamedPoint() { myName  = ""; }`

II.
```
public NamedPoint(int d1, int d2, String name){
    myX = d1;
    myY = d2;
    myName = name;
}
```

III.
```
public NamedPoint(int d1, int d2, String name){
    super(d1, d2);
    myName = name;
}
```

[5T] Kateri so pravilni konstruktorji za razred `NamedPoint`?

(a)	nobeden	☻	I in III
(b)	le I	(f)	II in III
(c)	le II	(g)	I in II
(d)	le III	(h)	I, II in III

[5T] Kateri so pravilni konstruktorji za razred `NamedPoint`, če POVSOD v gornjih razredih `private` zamenjamo s `public`?

(a)	nobeden	(e)	I in III
(b)	le I	(f)	II in III
(c)	le II	(g)	I in II
(d)	le III	☻	I, II in III

OPOMBA: Pri testu za DIRI 2003 ta naloga ne pride v poštev, ker bo obravnavana snov, potrebna za uspešno reševanje, še preveč »sveža«.

3.) Dan je naslednji del programa:

```
int j = 1;
int k = 1;
while(j <= n) {
  k = 1;
  while (k <=  n) {
    System.out.print(j + " " + k + "/");
    k = k * 2;
  }
  j++;
}
```

[5T] Kaj izpiše gornji del programa, če je `n` enak 3?

☻	`1 1/1 2/2 1/2 2/3 1/3 2/`
(b)	`1 1/1 2/1 3/2 1/2 2/2 3/3 1/3 2/3 3/`
(c)	`1 1/1 2/2 1/2 2/`
(d)	`1 1/1 2/1 4/2 1/2 2/2 4/3 1/3 2/3 4/`
(e)	nič od zgoraj navedenega

4.) Dana je naslednja metoda, ki ji manjka del kode. Metoda naj bi vrnila tabelo celih števil `sum`, za katero velja, da je `sum[i] = arr[0]  + arr[1] + ...  + arr[i]`. Primer: če je vrednost parametra `arr` tabela {1,4,1,3}, naj bi metoda vrnila tabelo {1,5,6,9}.

```
public int[] partialSum(int[] arr) {
  int[] sum  = new int[arr.length];
```

```
  for (int j = 0; j < sum.length; j++) sum[j] = 0;

  /* manjkajoca koda */

  return sum;
}
```

Dani sta dve implementaciji manjkajoče kode

I1:
```
  sum[0] = arr[0];
  for (int j = 1; j < arr.length; j++) sum[j] = sum[j - 1] + arr[j];
```

I2:
```
  for (int j = 0; j < arr.length; j++)
    for (int k = 0; k <= j; k++)
       sum[j] = sum[j] + arr[k];
```

[5T] Katere od trditev držijo?

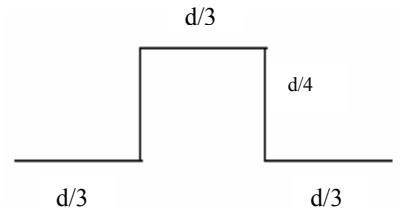
- ☻ Obe implementaciji delata pravilno.
- (b) I1 deluje, a nepravilno (ne da pravilen rezultat).
- (c) I1 deluje, a nepravilno (ne da pravilen rezultat).
- (d) I1 ne dela zaradi `ArrayIndexOutOfBoundsException`.
- (e) I2 ne dela zaradi `ArrayIndexOutOfBoundsException`.
- (f) Nobena od gornjih trditev.

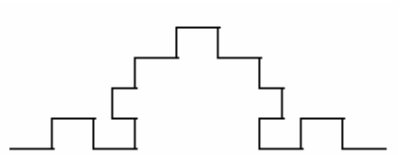
5.) **[25T]** Sestavite metodo, ki ima dva parametra. Prvi predstavlja število, drugi pa faktor. Metoda naj število razstavi na produkt potence faktorja in preostanka in v nizu rezultat. Če metodo pokličemo z Razstavi(10,2) dobimo niz "10 = 2^1 * 5".

```
public static String razstavi (int stevilo, int faktor) {
    // stevilo razstavi tako, da
    // velja stevilo = faktor^ekspon * preostanek
    int ekspon = 0; // kolikokrat nastopa faktor v stevilu
    String rezul = stevilo + " = "; // prvi del odgovora
    while (stevilo % faktor == 0)
    { // ce faktor se deli stevilo, ga delimo (in povecamo eksponent
      ekspon = ekspon + 1;
      stevilo = stevilo / faktor;
    }
    // v stevilu je sedaj preostanek
    // "zlozimo" rezultat
    rezul = rezul + faktor + "^" + ekspon + " * " + stevilo;
    return rezul;
}
```

Čas reševanja: 90 minut Ime in priimek: _____

6.) **[25T]** Dan je razred LogoZelva z metodami `fd(int)`, `bk(int)`, `right(int)`, `left(int)`, ki premaknejo želvo naprej, nazaj, oz. jo zasukajo desno, oz. levo za ustrezni parameter. S pomočjo objekta iz tega razreda sestavi **metodo** `public static void crta(LogoZelva ena, int n, int d)`, ki z želvo ena nariše črto lomljenko stopnje `n` in dolžine `d`. Črta stopnje 0 je kar črta v smeri želvinega gibanja dolžine d. Črto stopnje 1 prikazuje prva, črto stopnje 2 pa druga slika.





```
public static void crta(LogoZelva ena, int n, int dol)
{
      if (n == 0)
      { // crta stopnje n je kar daljicka dolzine dol
         ena.fd(dol);
      }
      else
      {
            crta(ena, n - 1, dol/3); // prvi vodoravni segment je
                                     // crta stopnje n – 1 in dolga
                                     // za tretjino prvotne
            ena.left(90);
            crta(ena, n - 1, dol/4); // prvi navpicni segment
            ena.right(90);
            crta(ena, n - 1, dol/3); // drugi vodoravni del
            ena.right(90);
            crta(ena, n - 1, dol/4); // drugi navpicni del
            ena.left(90);
            crta(ena, n - 1, dol/3); // tretji vodoravni del
      }
}
```

Čas reševanja: 90 minut Ime in priimek: _____

7.) **[25T]** Metka je vodja izmene v robotsko vodenem skladišču. Njeno delo je, da nadzira programerje, ki upravljajo robote-viličarje, ki skladajo zaboje v skladovnice. Janko jo je dolgo prosil, naj mu preskrbi delo programerja v tem skladišču. Končno je Metka le popustila in mu pri direktorju izposlovala zaposlitev. A že takoj prvi dan ga je Janko polomil. Robota je narobe sprogramiral. Tako so zabojniki v skladovnicah, ki jih nadzira Janko, zloženi (levo je naveden ZGORNJI zaboj skladovnice, v TEJ skladovnici so 4 zaboji) 1, 2, 3, 4; namesto 2, 1, 4, 3; ali pa (za 7 zabojev) 1, 2, 3, 4, 5, 6, 7 namesto 2, 1, 4, 3, 6, 5, 7 (po dva in dva zaboja sta torej v napačnem vrstnem redu).
Metka mora hitro ukrepati in napisati ustrezen program, ki bo vodil robota, da preložil zaboje v vseh skladovnicah. Robot zna izvajati tiste operacije, ki so osnovne operacije nad skladom. Pomagaj Metki in napiši ALGORITEM za vodenje robota, ki v danem skladu preloži elemente tako, da zamenja po dva in dva elementa.

OPOMBA: Pri testu za DIRI 2003 ta naloga ne pride v poštev, ker je objavljena pomotoma – tu bi morala biti druga naloga !

Ce pa vas ze zanima ;-)

Sklad se imenuje s:

```
pom = pripravi();
while (!s.prazen() )
{
   x = s.vrh();
   s.odstrani();
   if (!s.prazen()) // ce imamo se elemente – zaradi mozne lihosti stevila
   {
      y = s.vrh(); // se drugi element para
      s.odstrani();
      pom.vstavi(y);
      pom.vstavi(x); // obrnemo par
   }
   else // to bio le pri zadnjem elementu v primeru lihosti
   {
      pom.vstavi(x);
   }
}

// le se prelozimo nazaj

while (!.pom.prazen() )
{
   s.vstavi(pom.vrh());
   pom.odstrani();
}
```