

**T-106.1250 Informaatioverkostojen ohjelmointikurssi**  
**T-106.2001 Informaatioverkostojen studio 1**

Tentti 18. joulukuuta 2008

Vastaa kaikkiin kysymyksiin. Kirjoita jokaiseen vastauspaperiin vähintään päivämäärä, nimesi, opiskelijanumerosi ja allekirjoituksesi.

**1. Kysymykset (8p)**

Vastaa seuraaviin kysymyksiin. Sopiva vastauksen pituus on noin yksi kappale.

- a) Määrittele lyhyesti *olion* ja *luokan* käsitteet Java-ohjelmoinnissa.
- b) Mitkä ovat *listan* (engl. *list*) ja *joukon* (engl. *set*) oleelliset ero? Anna esimerkki tilanteesta, jossa käyttäisit listaa, ja toinen tilanteesta, jossa käyttäisit joukkoa.
- c) Mitä on *metodin ylikirjoittaminen* (engl. *method overriding*)? Anna esimerkki tilanteesta, jossa ylikirjoittamisesta on hyötyä.
- d) Java-ohjelmoinnissa *vakio* (engl. *constant*) ja *attribuutti* ovat eräänlaisia muuttujia. Mainitse kaksi muunlaista muuttujaa ja kerro, miten ne eroavat käyttötarkoituksiltaan.

## 2. Määrittelyt (8p)

Olkoon seuraava ohjelma:

```
1 public class HeiMaailma {
2
3     public static void main(String[] args) {
4
5         System.out.println("Hei maailma!");
6
7     }
8
9 }
```

Määrittele tarkasti seuraavien sanojen merkitys edellisessä ohjelmassa.

- a) static
- b) void
- c) main
- d) String[]
- e) args
- f) System
- g) out
- h) "Hei maailma!"

# 1. Mitä tämä koodi tulostaa? (8p)

Kirjoita näkyviin tuloste, jonka Tulostaja-luokan ajaminen aiheuttaa.

```
1 import java.util.Random;
2
3 public class Tulostaja {
4     protected boolean b;
5
6     public Tulostaja() {
7         this(false);
8         System.out.println("Tulostaja()");
9     }
10
11     public Tulostaja(boolean b) {
12         System.out.println("Tulostaja(boolean)");
13         this.b = b;
14     }
15
16     public void tulosta() {
17         System.out.println("Tulostaja.tulosta()");
18     }
19
20     public static void main(String[] args) {
21
22         Tulostaja t1 = new Oletustulostaja();
23         Tulostaja t2 = new Erikoistulostaja();
24
25         System.out.println("-----");
26
27         t1.tulosta();
28         t2.tulosta();
29
30     }
31 }
32
33 // jatkuu seuraavalla sivulla
```

```
34
35 private static class Oletustulostaja extends Tulostaja {
36
37     public void tulosta() {
38
39         Random rand = new Random();
40         int x = rand.nextInt(1) + 3;
41         int y = 8;
42
43         while (!(x < 1 || y > 12)) {
44
45             System.out.println("x: " + x);
46             System.out.println("y: " + y);
47
48             x--;
49             y++;
50
51         }
52     }
53 }
54
55 }
56
57 private static class Erikoistulostaja extends Tulostaja {
58
59     public Erikoistulostaja() {
60         super(true);
61         System.out.println("Erikoistulostaja()");
62     }
63
64     public void tulosta() {
65         try {
66             System.out.println("try");
67             if (this.b) {
68                 throw new RuntimeException("exception");
69             }
70         } catch (RuntimeException e) {
71             System.out.println("catch b");
72             return;
73         } catch (Exception e) {
74             System.out.println("catch a");
75         } finally {
76             System.out.println("finally");
77         }
78     }
79 }
80
81 }
82 }
```

## 1. Virheet (8p)

Alla on pelikorttia kuvaavan luokan koodi. Koodi kääntyy, mutta siinä on virheitä, eikä se toimi JavaDoc-kommenteissa kuvatulla tavalla. Luettele ja perustele virheelliset kohdat.

```
1 import java.util.*
2
3 /**
4  * Luokka kuvaa korttipelin korttia. Pelikorteilla on neljä erilaista maata
5  * ja arvot yhdestä kolmeentoista.
6  */
7 public class Kortti {
8
9     private Maa maa;
10
11     private int arvo;
12
13     /**
14     * Enumeraatio korttipelin maille.
15     */
16     private enum Maa {
17         HERTTA, PATA, RISTI, RUUTU
18     }
19
20     private static final String[] NIMET =
21         ("ässä kakkonen kolmonen nelonen vitonen kutonen"+
22         "seiska kasi yksi kymppi jätkä rouva kuningas").split(" ");
23     // split-metodi tekee merkkijonosta taulukon katkomalla sitä annetun
24     // erottimen avulla
25
26     /**
27     * Luo uuden pelikortin.
28     *
29     * @param maa kortin maa
30     * @param arvo kortin arvo
31     * @throws NullPointerException jos maa on null
32     * @throws IllegalArgumentException jos arvo on laiton
33     */
34     public Kortti(Maa maa, int arvo) {
35         if (maa == null) {
36             new NullPointerException("Maa ei voi olla null");
37         }
38         if (arvo < 1 && arvo > 13) {
39             new IllegalArgumentException("Arvo ei voi olla " + arvo);
40         }
41         maa = maa;
42         arvo = arvo;
43     }
44
45     /**
46     * Luo uuden korttipakan. Korttipakka on järjestetty maan ja arvon mukaan.
47     * Korttien arvot ovat nousevassa järjestyksessä (ässä ensin ja kuningas
48     * viimeisenä). Maat eivät ole missään tietyssä järjestyksessä.
49     *
50     * @return korttipakka
```

```

51  */
52  private static List<Kortti> luoPakka() {
53
54      List<Kortti> pakka = new ArrayList<Kortti>();
55
56      for (Maa m : Maa.values()) {
57          for (int i = 1; i < 13; i++) {
58              pakka.add(new Kortti(m, i));
59          }
60      }
61
62      return pakka;
63
64  }
65
66  /**
67   * Tekee jaon. Kullekin pelaajalle jaetaan pakan päältä vuorotellen
68   * haluttu määrä kortteja siten, että
69   * ensimmäinen pelaaja saa ensimmäinen kortin, toinen pelaaja toisen jne.
70   *
71   * @param pakka käytettävä korttipakka
72   * @param pelaajat pelaajien määrä
73   * @param korttejaKadessa jaettava korttien määrä
74   * @return tehty jako
75   */
76  public static Kortti[][] teeJako(List<Kortti> pakka, int pelaajat,
77      int korttejaKadessa) {
78
79      Kortti[][] jako = new Kortti[pelaajat][korttejaKadessa];
80
81      for (int pelaaja = 0; pelaaja < pelaajat; ++pelaaja) {
82          for (int kortti = 0; kortti < korttejaKadessa; ++kortti) {
83              jako[kortti][pelaaja] = pakka.iterator().next();
84          }
85      }
86
87      return jako;
88
89  }
90

```

```
91  /**
92   * Testaa Kortti-luokan toimintaa. Luo ja tulostaa korttipakan sekä
93   * yhden jaon.
94   *
95   * @param args ei käytössä
96   */
97  public static void main(String[] args) {
98
99      List<Kortti> pakka = luoPakka();
100
101      for (Kortti k : pakka) {
102          System.out.println(k);
103      }
104
105      // shuffle sekoittaa annetun listan
106      Collections.shuffle(pakka);
107
108      System.out.println("---");
109
110      Kortti[][] jako = teeJako(luoPakka(), 4, 5);
111      for (Kortti[] kasi : jako) {
112          for (Kortti k : kasi) {
113              System.out.println(k);
114          }
115          System.out.println("---");
116      }
117  }
118
119 }
```