
T-106.1250 Informaatioverkostojen ohjelmointikurssi

T-106.2001 Informaatioverkostot: Studio 1

Tentti 10.01.2011

Kirjoita jokaiseen vastauspaperiin vähintään päivämäärä, kurssin nimi, nimesi, opiskelijanumerosi ja allekirjoituksesi.

1. Kysymykset (8 pistettä)

Vastaa seuraaviin kysymyksiin.

- a) Määrittele lyhyesti *olion* ja *luokan* käsitteet Java-ohjelmoinnissa. (2 p)
- b) Mitä ovat abstraktit luokat ja metodit Javassa? (2 p)
- c) Mikä on *parametri* ja mihin sitä käytetään Java-ohjelmoinnissa? (1 p)

Selitä tarkasti mitä pääohjelmametodin (*engl. main method*) otsikon osat tarkoittavat `public static void main(String[] args)`. (à 0,5 p kohta, i kohdasta 1p)

- d) `public`
- e) `static`
- f) `void`
- g) `main`
- h) `(String[] args)`

2. Väittämät (8 pistettä)

Perustele ovatko seuraavat väittämät totta vai eivät. Pisteet saa perustelusta.

- a) Javassa kaikkia kokoelmia (*engl. Collections*) voi käydä läpi `Iterator`-luokan ilmentymän avulla ja tavalliset `for`-silmukat ovat oikeastaan turhia.
- b) Kaikki virheet on pakko käsitellä.
- c) Oletetaan, että olemme laatimassa korttipeliä Java-kielellä. Paras tapa toteuttaa pelikorttien maat olisi luoda seuraava luokka.

```
1
2 public class Maa {
3     public static final String HERTTA = "HERTTA";
4     public static final String PATA = "PATA";
5     public static final String RISTI = "RISTI";
6     public static final String RUUTU = "RUUTU";
7 }
8
```

- d) Seuraavan luokan kääntäminen aiheuttaa kääntövirheen.

```
1
2 public class Tentti {
3
4     private int luku;
5
6     public Tentti(int luku) {
7         this.luku = luku;
8     }
9
10    public static int onkoIso() {
11        if (this.luku > 10) {
12            return this.luku;
13        } else (this.luku <= 10) {
14            return 0;
15        }
16    }
17
18    public static void main(String[] args) {
19        System.out.println(Tentti.onkoIso());
20    }
21 }
22
```

3. Essee (8 pistettä)

"Java-perintä? Niin, mäki sain isoäidiltä ennakkoperintönä kahvikuppeja..."

Ystäväsi on kuunnellut juttujasi Studio 1:ltä eikä ole aivan ymmärtänyt, mitä perintä (*engl. inheritance*) on ja mihin sitä voisi koodatessa käyttää. Myös rajapintaluokan (liittymä, *engl. interface*) idean hän pyytää selittämään. Joitain olio-ohjelmoinnin peruskäsitteitä kuten luokka, olio, attribuutti ja metodi kaverisi tuntee jo.

Kirjoita siis lyhyt (n. 2 sivun) kirjallinen selitys ystävällesi aiheesta.

4. Mitä tämä koodi tulostaa? (8 pistettä)

Kirjoita näkyviin tuloste, jonka Tulostaja-luokan ajaminen aiheuttaa.

```
1 public abstract class Tulostaja {
2     protected boolean b;
3
4     public Tulostaja() {
5         System.out.println("Tulostajan konstruktori");
6     }
7
8     public Tulostaja(boolean b) {
9         this.b = b;
10    }
11
12    public abstract void laske(int alku);
13
14    public static void main(String[] args) {
15        Tulostaja t1 = new Oletustulostaja();
16        System.out.println("-----");
17        Tulostaja t2 = new Erikoistulostaja();
18        System.out.println("-----");
19        Tulostaja t3 = new Virhetulostaja();
20        System.out.println("-----");
21
22        t1.laske(3);
23        System.out.println("-----");
24        t2.laske(2);
25        System.out.println("-----");
26        t3.laske(0);
27    }
28 }
29
30
31
32 public class Oletustulostaja extends Tulostaja {
33
34     public Oletustulostaja() {
35         super();
36         System.out.println("Oletus-luokan konstruktori");
37     }
38
39     public void laske(int alku){
40         int tulos = 0;
41
42         for (int i = 3; i >= 0; i--) {
43             System.out.println(tulos);
44             tulos += alku * i * (this.b ? 1 : -1);
45         }
46
47         System.out.println(tulos);
48     }
49 }
50 }
51 // jatkuu seuraavalla sivulla
```

```

52 public class Erikoistulostaja extends Tulostaja {
53
54     public void laske(int alku) {
55         System.out.println(alku);
56
57         if (alku >= 0) {
58             switch (alku) {
59                 case 0: this.laske(--alku); break;
60                 case 1: this.laske(--alku);
61                 case 2: this.laske(--alku);
62                 case 3: this.laske(--alku);
63                 default: System.out.println("default");
64             }
65         }
66     }
67
68 }
69
70
71 public class Virhetulostaja extends Tulostaja{
72
73     public Virhetulostaja() {
74         super(true);
75     }
76
77     public void laske(int alku) {
78
79         try {
80             if (this.b) {
81                 throw new Exception("virhe");
82             }
83         } catch (RuntimeException e) {
84             System.out.println("aika juosta!");
85         } catch (Exception e) {
86             System.out.println("poikkeus!");
87         } finally {
88             System.out.println("vihdoin!");
89         }
90     }
91 }
92
93 }
94

```