T-106.1250 Informaatioverkostojen ohjelmointikurssi

T-106.2001 Informaatioverkostot: Studio 1

Tentti 09.01.2012

Kirjoita jokaiseen vastauspaperiin vähintään päivämäärä, kurssin nimi, nimesi, opiskelijanumerosi ja allekirjoituksesi. Apuvälineinä ei saa käyttää kuin muistiinpanovälineitä.

1. Kysymykset (8 pistettä)

Vastaa seuraaviin kysymyksiin.

a) Mikä on *listan* (engl. list) ja *joukon* (engl. set) oleellinen ero?

b) Mikä on *MVC-malli* ja mihin sitä käytetään? (2 pistettä)

c) Mikä on *tarkistettujen poikkeusten* (engl. checked exceptions) ja *tarkistamattomien poikkeusten* (engl. unchecked exceptions) ero?

d) Mitä on *rekursio*? Anna esimerkki ongelmasta, jonka ratkaiseminen rekursion avulla on järkevää. Itse ratkaisua ei tarvitse esittää. (2 pistettä)

2. Väittämät (8 pistettä)

Perustele ovatko seuraavat väittämät totta vai eivät. Pisteet saa perustelusta. Jos väittämä on neljän pisteen arvoinen, vaaditaan neljän pisteen saamiseen esimerkkejä.

a) `Switch`-lohkon `default`-lause suoritetaan jokaisella `switch`-lohkon ajokerralla. (2 pistettä)

b) Hinnoittelu, vaatimusmäärittely ja toteutusmetodi eivät liity mitenkään toisiinsa. (4 pistettä)

c) Sijoituslauseen `a = b;` jälkeen muuttujassa `a` on kopio muuttujasta `b`. (2 pistettä)

3. Määrittelyt (8 pistettä)

Vastaa kumpaankin kohtaan esseemuotoisella lyhyellä (n. 1 sivu) vastauksella.

a) <u>Dokumentointi ja kommentointi.</u>
Javan valmiit pakkaukset ja luokat metodeineen on dokumentoitu ja julkaistu JavaDoc-muotoisina Internetissä API-dokumentaationa. Mitä hyötyä dokumentoinnista on? Miksi ohjelmakoodia kuuluu dokumentoida? Miksi ei sovi unohtaa myöskään kommentointia? (4 pistettä)

b) <u>Luokat.</u>
Java on olio-ohjelmointikieli ja oliot luodaan luokista. Mitä luokat sisältävät? Mitä eri avainsanoja voidaan luokkaa tai luokan osia määritellessä käyttää? Minkälaisia erilaisia luokkia on? (4 pistettä)

4. Virheet (8 pistettä)

Alla on ohjelma, joka sisältää luokat `Pokemon` ja `Pokemonmestari`. Pelkästä virheellisen kohdan löytämisestä ei saa pistettä. Tehtävästä voi saada vain kokonaisia pisteitä ja pyöristys tapahtuu alaspäin.

a) Koodi ei mene kääntäjästä läpi. Etsi ja korjaa syntaksivirheet ja semantiikkavirheet, joista kääntäjä huomauttaa (5kpl). (2,5 pistettä)

b) Koodi menee kääntäjästä läpi, mutta kaatuu. Etsi, nimeä ja korjaa semantiikkavirhe (1kpl). (1 piste)

c) Koodi ei kaadu, mutta luokan `Pokemonmestari` `main`-lohkoa ajettaessa ohjelma ei toimi kuten kommenteissa ilmoitetaan. Etsi, kuvaile ja korjaa logiikkavirheet (3kpl). (4,5 pistettä)

```
1
2     public class Pokemon implements Comparable {
3           private int taso, nopeus, voima;
4           private String nimi;
5
6           public Pokemon(int taso, int voima, int nopeus) {
7                 this.taso = (taso > 0 ? taso : 1);
8                 this.nopeus = nopeus;
9                 this.voima = voima;
10                if (this.nopeus >= 10) {
11                      if (this.voima >= 40) {
12                            if (this.nopeus >= 40) {
13                                  this.nimi = "Charizard";
14                            } else if (this.nopeus >= 30){
15                                  this.nimi = "Gyarados";
16                            } else {
17                                  this.nimi = "Onix";
18                            }
19                      } else {
20                            if (this.nopeus < 50) {
21                                  if (this.voima < 20) {
22                                        this.nimi = "Caterpee";
23                                  } else if (this.nopeus >= 30) {
24                                        this.nimi = "Bulbasaur";
25                                  } else {
26                                        this.nimi = "Psyduck";
27                                  }
28                            } else {
29                                  this.nimi = "Pikachu";
30                            }
31                } else {
32                      this.nimi = "Slowpoke";
33                }
34          }
35
36          public String annaNimi() {
37                return this.nimi;
38          }
39
40          public int annaNopeus() {
41                return this.nopeus;
42          }
43
44          public int annaTaso() {
45                return this.taso;
46          }
```

```
47
48          public int annaVoima() {
49                return this.voima;
50          }
51
52          @Override
53          public int compareTo(Object o) {
54                if (o instanceof Pokemon) {
55                      Pokemon pokemon = o;
56                      int tasoero = this.annaTaso()
57                      - pokemon.annaTaso();
58                      int voimaero = this.annaVoima()
59                      - pokemon.annaVoima();
60                      int nopeusero = this.annaNopeus()
61                      - pokemon.annaNopeus();
62                      if (Math.abs(tasoero) < 10) {
63                            if (Math.abs(voimaero) < 10) {
64                                  if (Math.abs(nopeusero) < 10) {
65                                        return Integer.signum
66                                        (nopeusero);
67                                  } else if (Math.abs(tasoero) <5){
68                                        return Integer.signum
69                                        (voimaero+nopeusero);
70                                  } else return Integer.signum
71                                  (tasoero+voimaero);
72                            } else return Integer.signum(voimaero);
73                      } else return Integer.signum(tasoero);
74                      //    static int signum(int i)
75                  //  Returns the signum function of the int value.
76                }
77                return 0;
78          }
79
80          @Override
81          public String toString() {
82                return "Tason " + this.annaTaso()
83                + " " + this.annaNimi();
84          }
85
86    }
```

```
1     import java.util.ArrayList;
2     import java.util.Collections;
3
4
5     public class Pokemonmestari {
6           private String nimi;
7           private ArrayList<Pokemon> pokemonit;
8           private static int pokemonMaksimi = 6;
9
10          public Pokemonmestari(String nimi) {
11                this.nimi = nimi;
12                this.pokemonit = new ArrayList<Pokemon>;
13          }
14
15          public void nappaaPokemon(Pokemon pokemon) {
16                this.pokemonit.add(pokemon);
17                Collections.sort(this.pokemonit);
18                //    static void sort(List list)
```

```
19          // Sorts the specified list into ascending order,
20              // according to the natural ordering
21              // (Comparable) of its elements.
22
23              Collections.reverse(this.pokemonit);
24              //    static void reverse(List list)
25          // Reverses the order of the elements in the list.
26
27              while (this.pokemonit.size()
28                          > Pokemonmestari.pokemonMaksimi){
29                  this.pokemonit.remove(this.pokemonit.size()-1);
30              }
31      }
32
33      public String toString() {
34              String string = "**Pokemonmestari "
35                      + this.nimi + "** ";
36              string = string.concat(this.pokemonit.isEmpty()
37                          ? "ei Pokemoneja" : "\nPokemonit ("
38                                  + this.pokemonit.size()+"kpl):";
39
40              //            String concat(String str)
41          // Concatenates the specified string
42              // to the end of this string.
43
44              for (a = 1; a < this.pokemonit.size(); a++) {
45                      string = string.concat("\n"
46                              + this.pokemonit.get(a).toString());
47              }
48
49              string = string.concat("\n---");
50              return string;
51      }
52
53      public static void main(String[] args) {
54              Pokemonmestari mestari;
55              if (args[0] != null) {
56                      mestari = new Pokemonmestari(args[0]);
57              } else {
58                      mestari = new Pokemonmestari("Mestari");
59              }
60
61              System.out.println(mestari);
62              /* printtaa:
63               * **Pokemonmestari Mestari** ei Pokemoneja
64               * ---
65               * */
66
67              /* Luodaan Pikachu-niminen Pokemon */
68              Pokemon pika = new Pokemon(25,25,55);
69              mestari.nappaaPokemon(pika);
70              /* Luodaan Caterpee-niminen Pokemon */
71              Pokemon toukka = new Pokemon(10,10,40);
72              mestari.nappaaPokemon(toukka);
73              /* Luodaan Onix-niminen Pokemon */
74              Pokemon kivi = new Pokemon(20,40,30);
75              mestari.nappaaPokemon(kivi);
76              /* Luodaan Bulbasaur-niminen Pokemon */
77              Pokemon bulba = new Pokemon(15,25,35);
78              mestari.nappaaPokemon(bulba);
79              /* Luodaan Psyduck-niminen Pokemon */
```

```
80                  Pokemon ankka = new Pokemon(20,35,15);
81                  mestari.nappaaPokemon(ankka);
82
83                  System.out.println(mestari);
84                  /* printtaa:
85                   * **Pokemonmestari Mestari**
86                   * Pokemonit (5kpl):
87                   * Tason 20 Onix
88                   * Tason 20 Psyduck
89                   * Tason 25 Pikachu
90                   * Tason 15 Bulbasaur
91                   * Tason 10 Caterpee
92                   * ---
93                   * */
94
95                  /* Luodaan Slowpoke-niminen Pokemon*/
96                  Pokemon hidas = new Pokemon(25,40,5);
97                  mestari.nappaaPokemon(hidas);
98                  /* Luodaan Gyarados-niminen Pokemon */
99                  Pokemon fisu = new Pokemon(30,40,30);
100                 mestari.nappaaPokemon(fisu);
101                 /* Luodaan Caterpee-niminen Pokemon */
102                 Pokemon otokka = new Pokemon(30,15,45);
103                 mestari.nappaaPokemon(otokka);
104                 /* Luodaan Charizard-niminen Pokemon */
105                 Pokemon lisko = new Pokemon(30,50,40);
106                 mestari.nappaaPokemon(lisko);
107
108                 System.out.println(mestari);
109                 /* printtaa:
110                  * **Pokemonmestari Mestari**
111                  * Pokemonit (6kpl):
112                  * Tason 30 Charizard
113                  * Tason 30 Gyarados
114                  * Tason 20 Onix
115                  * Tason 25 Slowpoke
116                  * Tason 25 Pikachu
117                  * Tason 30 Caterpee
118                  * ---
119                  * */
120         }
121
122 }
```