ITKP102 Ohjelmointi 1 (6 op), arvosteluraportti

Tentaattori: Antti-Jussi Lakanen

18. toukokuuta 2018

Yleistä

Tentti[1] oli pistekeskiarvon (12.7) perusteella vaikeudeltaan kohtuullinen. Omasta tehtäväpaperista saa kopion Antti-Jussilta, huone C414.2. Jos en ole paikalla, laita sähköpostia tai soita 040 805 3276.

Uusintojen ajankohdat löydät kurssin Korppi-sivulta.

Tehtävä	Teki	Keskiarvo	Keskihajonta	Tarkastaja
T1	50	3.01	1.98	Oscar Calderón
T2	51	2.84	2.15	Mikko Häyrynen
T3	50	3.55	1.70	Mikko Häyrynen
T4	50	3.25	2.21	Oscar Calderón
Yht		**12.7**	6.4	

Arvosteluasteikko

```
Arvolause   Pistemäärä (alaraja)
5           24
4           21
3           18
2           15
1           12
```

[1] http://users.jyu.fi/~anlakane/ohjelmointi1/tentit/2018-05-18-tentti2.pdf

JYVÄSKYLÄN YLIOPISTO

Tehtävä 1 (6 p.)

Yleiset huomiot

Ensimmäisessä tehtävässä laskettiin kolmion pinta-ala. Pinta-alan laskemisen piti tapahtua omassa funktiossa, joka ei saanut tulostaa mitään. Kolmion pinta-ala piti laskea käyttäjän syötteillä ja vinkeiksi oli annettu `Console.Readline()`- ja `double.Parse()`-metodit.

Yleisimpiä virheitä olivat seuraavat:

- `KolmionAla`-funktiota kutsuttiin (muualla kuin `WriteLinen` sisällä), mutta sen arvoa ei tallennettu muuttujaan.
- `KolmionAla`-funktiossa tulostettiin tulos.
- `KolmionAla`-funktiossa ei ollut `return`-lausetta.
- `Parse()`- ja `ReadLine()` -metodeja ei osattu käyttää oikein.

Mallivastaus

```
using System;

/// <summary>
/// Ohjelma laskee kolmion alan
/// </summary>
public class KolmionAla
{
    /// <summary>
    /// Pyydetään käyttäjältä syötteet. Tehdään tulostukset.
    /// </summary>
    public static void Main()
    {
        Console.Write("Anna kolmion kanta >");
        string kantaTeksti = Console.ReadLine();
        double kanta = double.Parse(kantaTeksti);
        Console.Write("Anna kolmion korkeus >");
        string korkeusTeksti = Console.ReadLine();
        double korkeus = double.Parse(korkeusTeksti);
        double ala = KolmionAla(kanta, korkeus);
        Console.WriteLine("Kolmion pinta-ala on " + ala);
    }

    /// <summary>
    /// Laskee kolmion pinta-alan
    /// </summary>
    /// <param name="kanta">kolmion kanta</param>
    /// <param name="korkeus">kolmion korkeus</param>
    /// <returns>kolmion pinta-ala</returns>
```

```
    public static double KolmionAla(double kanta, double korkeus)
    {
        return kanta * korkeus / 2.0;
    }

}
```

Pisteytys

- Käyttäjän syöte luettiin oikein 1 p
- Parsittiin syötteet oikein 1 p
- `KolmionAla`-funktio toimi oikein 1 p
- Ohjelman rakenne oli oikeanlainen 1 p.
- Dokumentointi oli kurssikäytänteiden mukainen 1 p.
- `KolmionAla`-funktiota käytettiin oikein ja sen tulos tulostettiin pääohjelmassa 1 p.

Tehtävä 2 (6 p.)

Yleiset huomiot

Toisessa tehtävässä tuli kirjoittaa **yksi** pääohjelma, jonka sisällä on kaksi erillistä silmukkarakennetta. Kummassakin tulostetaan kakkosen potenssit väliltä 1–128. Tulostuksen muodolla ei ollut merkitystä. Yleisin ratkaisu oli käyttää `for`- ja `while`-silmukoita, mutta myös `do-while` oli käytössä muutamassa vastauksessa. Jos tulosteen luvut oli kovakoodattu ilman silmukkaa, pisteitä vähennettiin. Kommentointia ei tehtävässä poikkeuksellisesti vaadittu. `System`-nimiavaruuden sai olettaa olevan käytössä.

Tehtävä osattiin melko heikosti, ja keskihajonta oli korkea. Suuressa osassa vastauksia oli käytetty tarpeettoman paljon apumuuttujia, jotka monimutkaistavat ohjelmaa. Ylimääräisten muuttujien käytöstä ei kuitenkaan pääosin vähennetty pisteitä. Yleisimpiä virheitä olivat seuraavat:

- Silmukkamuuttujan arvoa oli kasvatettu, mutta muutettua arvoa ei oltu sijoitettu takaisin muuttujaan. Vertaa `i * 2` ja `i *= 2`.
- Tulostettiin kaikki luvut väliltä 1–128.
- Eri silmukoissa oli käytetty saman nimisiä muuttujia. Muuttujat ovat samalla näkyvyysalueella, joten tämä ei ole mahdollista C#:ssa.

Mallivastaus

```
public static void Main()
{
    for (int i = 1; i <= 128; i *= 2)
    {
```

```
      Console.WriteLine(i);
    }

    int j = 1;
    while (j <= 128)
    {
        Console.WriteLine(j);
        j *= 2;
    }
}
```

Pisteytys

- Pääohjelman yleinen rakenne oikein 1 p.
- Osattu käyttää kahta erilaista silmukkaa, joiden rakenne on pääosin järkevä 1p.
- Ensimmäinen silmukka toimii oikein 2 p.
- Toinen silmukka toimii oikein 2 p.
- Syntaksivirheistä ja huonosta yleisestä ilmeestä (nimeämiskäytäntö, sulkuvirheet, kirjainkoko avainsanoissa jne.) vähennettiin enintään 2 p.

Tehtävässä ei pyydetty kirjoittamaan luokkaa tai testejä, eikä niiden sisällyttämisestä myönnetty tai vähennetty pisteitä.

Tehtävä 3 (6 p.)

Yleiset huomiot

Kolmas tehtävä koostui kolmesta osatehtävästä, joihin oli tarkoitus vastata lyhyesti ja kattavasti. Kommentointi ja operaattorien erot osattiin hyvin, mutta kuormittaminen oli monelle vieras asia.

Osatehtävien pistekeskiarvot

Osatehtävä	Keskiarvo	Keskihajonta
T3.1	0.63	0.84
T3.2	1.38	0.73
T3.3	1.55	0.69
Yht	**3.55**	**1.70**

Ratkaisut ja pisteytys

1. Aliohjelman kuormittamisella tarkoitetaan sitä, että saman niminen aliohjelma määritellään useaan kertaan siten, että parametrien määrä tai parametrien tyypit ovat erilaiset. Osapisteitä myönnettiin, jos vastauksessa mainittiin, että aliohjelmaa voidaan *kutsua* eri määrällä tai eri tyyppisillä parametreilla. Idean selittämisestä sai yhden pisteen ja esimerkistä toisen. Kokonaisia ohjelmia tai aliohjelmia täysiin pisteisiin ei tarvinnut kirjoittaa. Esimerkkikoodin pienistä virheistä ei vähennetty pisteitä, jos idea oli oikein.

   ```
   public override void Begin()
   {
       LuoPallo();
       LuoPallo(100, 100);
   }

   public static void LuoPallo() { ... }

   public static void LuoPallo(double x, double y) { ... }
   ```

2. Kaksiparametrisen funktion dokumentaatiokommentista tuli löytyä `summary`-osio, kaksi `param`-osiota ja `returns`-osio. Osapisteitä myönnettiin jokaisesta oikein kirjoitetusta osiosta. Alla esimerkkidokumentaatio kuvitteelliselle `Summa`-funktiolle.

   ```
   /// <summary>
   /// Aliohjelma laskee kahden kokonaisluvun summan.
   /// </summary>
   /// <param name="ekaluku">Ensimmäinen kokonaisluku.</param>
   /// <param name="tokaluku">Toinen kokonaisluku.</param>
   /// <returns>Kokonaislukujen summa.</returns>
   public static int Summa(int ekaluku, int tokaluku)
   ```

   `summary`-osiossa kerrotaan yleisesti, mitä funktio tekee, `param`-osioilla dokumentoidaan parametrit ja `return`-osiolla funktion paluuarvo.

3. `=` on sijoitusoperaattori, joka sijoittaa oikeanpuoleisen arvon vasemmanpuoleiseen muuttujaan. Esimerkki: `int i = 0;`

   `==` on vertailuoperaattori, joka vertaa operandejaan toisiinsa ja palauttaa totuusarvon. Esimerkki: `if (i == 0) break;`

   Pisteitä myönnettiin yksi kummastakin oikein selitetystä operaattorista.

Tehtävä 4 (6 p.)

Yleiset huomiot

Neljännessä tehtävässä tulostettiin luvut 1-100, joihin liitettiin välillä lisäsanoja. Dokumentteja ei poikkeuksellisesti tarvinut kirjoittaa. Tehtävän suhteellisen helppouden takia käytännössä kaikkiin virheisiin tartuttiin ja niistä sakotettiin. Tehtävän keskiarvo oli 3,25. Yleisimpiä virheitä olivat seuraavat:

- `i <= 100` kirjoitettiin matemaattisen notaation mukaisesti paperille
- Aloitettiin tulostus nollasta.
- Lopetettiin tulostus 99:än tai 101:en.
- `Hip` tai `Hei` tulostuivat seuraavalle riville
- Edellisen virheen myötä seuraava luku tulostettiin `Hip` tai `Hei` kanssa samalle riville.

Mallivastaus

```
using System;

class HipHei
{
    private static void Main()
    {
        for (int i = 1; i <= 100; i++)
        {
            Console.Write(i + " ");
            if (i % 3 == 0)
            {
              Console.Write("Hip");
            }
            if (i % 5 == 0)
            {
              Console.Write("Hei");
            }
            Console.WriteLine();
        }
    }
}
```

Pisteytys

Tehtävän maksimipisteisiin tehtiin vähennyksiä seuraavasti.

- Lukua ei koskaan tulostettu -3 p. (Tulostettiin siis vain Hippiä ja Heitä)
- `Hip` tai `Hei` tuli lukua seuraavalle riville -1 p.
- Seuraava luku tuli sanan `Hip` tai `Hei` kanssa samalle riville -1 p.
- Viidellätoista jaollisella rivillä tulostettiin myös kolmella ja viidellä jaolliset tulokset. Esimerkiksi `15 Hip 15 Hei 15 HipHei`. -1 p.
- Syntaksivirheistä vähennettiin enintään 2 p.

Tehtävän minipistemäärä oli tietenkin nolla.