

ITKP102 Ohjelmointi 1 (6 op), arvosteluraportti

Tentaattori: Antti-Jussi Lakanen

20. toukokuuta 2021

Yleistä

Tentti¹ oli pistekeskiarvon 13,6 (keskihajonta 5,8) perusteella vaikeudeltaan tyypillistä keskitasoa. Huomaa, että demopisteet on laskettu tentin päälle, ja arvosana lasketaan vasta sen jälkeen. Opiskelijan omat tehtävät ovat nähtävissä TIMissä. Viimeisen uusinnan ajankohdan löydät Sisusta.

Tehtävä	Teki	Keskiarvo	Tarkastaja
T1	40	4,0	Antti-Jussi Lakanen
T2	42	4,8	Antti-Jussi Lakanen
T3	42	3,8	Antti-Jussi Lakanen
T4	38	2,2	Antti-Jussi Lakanen
Yht		13,6	

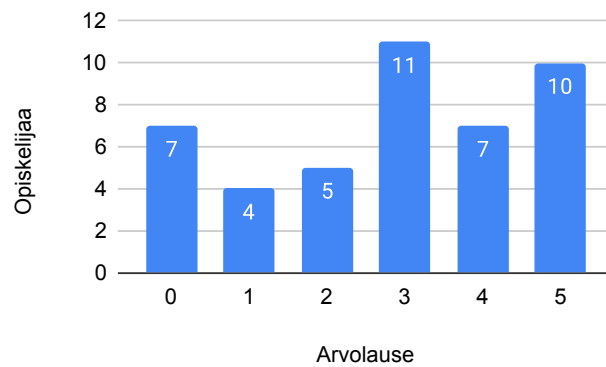
¹<http://users.jyu.fi/~anlakane/ohjelmointi1/tentit/2021-05-20-tentti2.pdf>



JYVÄSKYLÄN YLIOPISTO

Arvosteluasteikko ja arvosanajakauma

Arvosana	Pistemäärä (alaraja)
5	24
4	21
3	18
2	16
1	12



Kuva 1: Tentin arvosanajakauma.

Tehtävä 1 (6 p.)

Tee funktio Palindromi, joka palauttaa true tai false sen mukaan onko argumenttina annettu merkkijono sama etu- ja takaperin luettuna vai ei.

Esimerkkejä

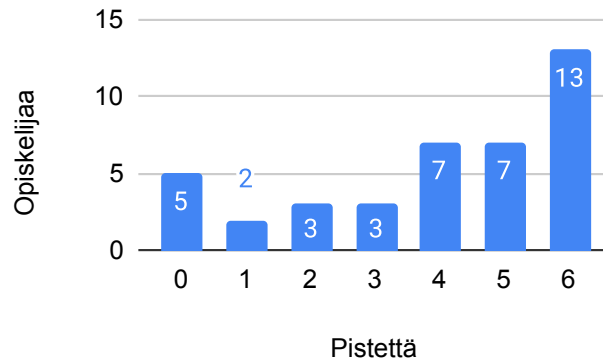
Palindromi("kissa") palauttaa false. Palindromi("Saippuakauppias") palauttaa true.

Vaatimukset

1. Kirjoita vastaukseksi kääntyvä ohjelma, joka sisältää tarvittavat using-lauseet, luokkamäärittelyn ja dokumentaatiokommentit.
2. Funktio Palindromi ei saa tulostaa mitään.
3. Mitään valmiita aliohjelmia/funktioita, kuten Array.Reverse, LINQ:ta, laajennusmetodeja tai vastaavia ei saa käyttää. Mikäli tätä kohtaa rikotaan, tulos tehtävästä on 0 pistettä.
4. Pääohjelmassa funktiota tulee kutsua tasan kaksi kertaa siten että ensimmäinen kutsu tuottaa arvon true ja toinen kutsu tuottaa false. Näiden kutsujen palauttama tulos tulee tulostaa ruudulle. Pääohjelma ei saa tulostaa mitään muuta.
5. Suuraakkosia ja pienaakkosia tulee käsitellä samalla tavalla. Niinpä sekä "Saippuakauppias" että "saippuakauppias" ovat palindromeja.
6. Välilyöntejä ei tarvitse huomioida. Niinpä "Saippua kauppias" ei ole palindromi.

Malliratkaisu

```
/// <summary>
/// Tutkii, onko merkkijono palindromi.
/// </summary>
/// <param name="jono">Tutkittava merkkijono</param>
/// <returns>Onko jono palindromi.</returns>
public static bool Palindromi(string jono)
{
    for (int i = 0; i < jono.Length / 2; i++)
    {
        if (Char.ToLower(jono[i]) != char.ToLower(jono[jono.Length - 1 - i]))
        {
            return false;
        }
    }
    return true;
}
```



Kuva 2: Tehtävän 1 pistejakauma.

Tehtävä 2 (6 p.)

Tee seuraava konsoliohjelma (luokka, pääohjelma, tarvittavat using-lauseet, tarvittavat funktiot, dokumentaatiokommentit) Visual Studiossa / Riderissa ja palauta valmis vastaus alla olevaan vastauslaatikkoon. Et voi testata ohjelman toimintaa Timissä.

Vaatimukset

Pääohjelmassa käyttäjältä luetaan näppäimistösyötteenä ympyrän säde. Kyseinen luku annetaan sitten argumenttina YmpyränAla-funktiolle, joka palauttaa säteen perusteella ympyrän pinta-alan. Lopuksi pääohjelma tulostaa pinta-alan. Alla esimerkki ohjelman toiminnasta.

```
Anna ympyrän säde > 5,1
Ympyrän pinta-ala on 81,713
```

Vinkki: Liukuluku luetaan merkkijonosta double.Parse-aliohjelmalla. Huomautus: Yllä olevan esimerkin tulostusta on lyhennetty luettavuuden parantamiseksi. Vastauksessa tulostusta ei tarvitse lyhentää tai pyöristää.

Malliratkaisu

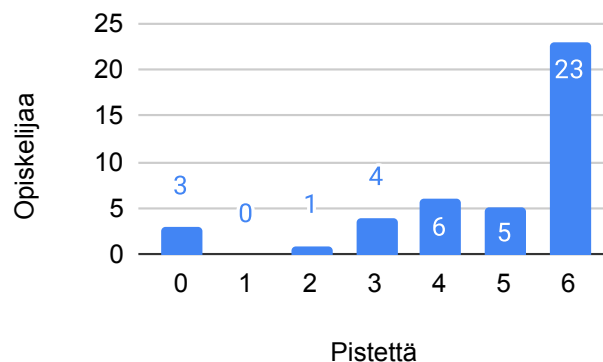
```
/// @author Antti-Jussi Lakanen
/// @version 20.5.2021
///
/// <summary>
/// Tehtävä 2. Ympyrän pinta-ala.
/// </summary>
public class Tehtava2
{
    /// <summary>
    /// Kysytään käyttäjältä ympyrän säde. Tulostetaan pinta-ala.
    /// </summary>
    public static void Main()
    {
```

```

        Console.WriteLine("Anna ympyrän säde > ");
        string luku = Console.ReadLine();
        double sade = double.Parse(luku);
        double ala = YmpyranAla(sade);
        Console.WriteLine("Ympyrän pinta-ala on " + Math.Round(ala, 3));
    }

    /// <summary>
    /// Funktio antaa ympyrän pinta-alan.
    /// </summary>
    /// <param name="sade">Säde</param>
    /// <returns>Pinta-ala</returns>
    public static double YmpyranAla(double sade)
    {
        return Math.PI * sade * sade;
    }
}

```



Kuva 3: Tehtävän 2 pistejakauma.

Tehtävä 3 (6 p.)

Tee konsoliohjelma (luokka, pääohjelma, tarvittavat using-lauseet, dokumentaatiokomentit), jossa

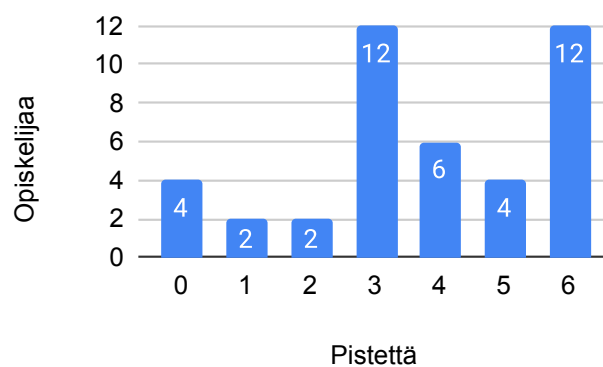
- for-silmukkarakennetta käyttäen tulostetaan luvut 1, 2, 4, 8, 16, ..., 128. Huom! Silmukkamuuttujaa (tyypillisesti i) ei saa muuttaa silmukan runko-osan sisällä, ja sitten perään
- while-silmukkarakennetta käyttäen tulostetaan luvut 1, -2, 3, -4, 5, -6, ..., 99. Tässä kohdassa silmukkamuuttujaa saa muuttaa silmukan runko-osan sisällä.

Yhtään if-lausetta ei saa käyttää koko ohjelmassa. Silmukan toistoehdot ei luonnollisesti lasketa tässä kielletyksi if-lauseeksi. Ohjelma ei saa tulostaa mitään muuta.

Malliratkaisu

```
/// @author Antti-Jussi Lakanen
/// @version 20.5.2021
///
/// <summary>
/// Tehtävä 3. Tulosteet.
/// </summary>
public class T3_Tulosteet
{
    /// <summary>
    /// Tulostetaan kaksi lukujonoa
    /// </summary>
    public static void Main()
    {
        for (int i = 1; i <= 128; i *= 2)
        {
            Console.WriteLine(i);
        }

        int j = 1;
        int kerroin = 1;
        while (j <= 99)
        {
            Console.WriteLine(j * kerroin);
            j++;
            kerroin *= -1;
        }
    }
}
```



Kuva 4: Tehtävän 3 pistejakauma.

Tehtävä 4 (6 p.)

Toteuta seuraava ohjelma (luokka, pääohjelma, tarvittavat using-lauseet, tarvittavat funktiot, dokumentaatiokommentit).

Vaatimukset

1. Luo pääohjelmassa taulukko, jonka pituus on satunnainen väliltä 10..20. Taulukko sisältää satunnaisia kokonaislukuja väliltä -10..10.
2. Tee funktio Jarjesta joka ottaa kyseisen taulukon ja järjestää sen laskevaan järjestykseen (suuremmat luvut ennen pienempiä lukuja) alla olevan algoritmin mukaisesti.

Algoritmi funktiolle Jarjesta

1. Käydään taulukko läpi alusta loppuun.
2. Mikäli kaksi perättäistä alkioita eivät ole järjestyksessä (vasemmalla oleva alkio on pienempi kuin oikealla), vaihdetaan näiden alkioden paikkoja keskenään.
3. Käsitellään jokainen perättäisten alkioden pari kohdassa 2 kuvatulla tavalla.
4. Jos tehtiin alkioden vaihtoja (ks. kohta 2) aloitetaan uudestaan kohdasta 1.
5. Kun järjestäminen on tehty, funktio palauttaa järjestetyn taulukon.
6. Huom! Parametrina saatua taulukkoa EI saa muuttaa vaan funktion tulee tehdä uusi taulukko.

Malliratkaisu

```
/// @author Antti-Jussi Lakanen
/// @version 20.5.2021
///
/// <summary>
/// Tehtävä 4. Lukujen järjestäminen.
/// </summary>
public class T4_BubbleSort
{
    /// <summary>
    /// Luodaan taulukko ja kutsutaan funktiota.
    /// </summary>
    public static void Main()
    {
        Random r = new Random();
        int[] luvut = new int[r.Next(10, 21)];
        for (int i = 0; i < luvut.Length; i++)
        {
            luvut[i] = r.Next(-10, 11);
        }
    }
}
```

```

    }

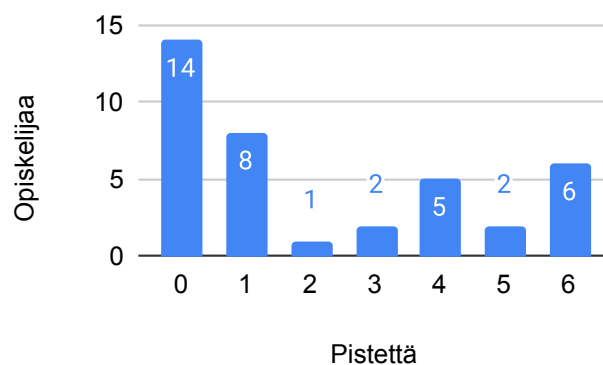
    // Alla olevaa lausetta ei kirjaimellisesti ottaen
    // tehtävässä vaadittu.
    int[] jarjestetty = Jarjesta(luvut);
}

public static int[] Jarjesta(int[] luvut)
{
    int[] jarjestetty = (int[])luvut.Clone();

    // Tai seuraavasti:
    // int[] jarjestetty = new int[luvut.Length];
    // for (int i = 0; i < luvut.Length; i++)
    //     jarjestetty[i] = luvut[i]

    int i = 1;
    while (i < jarjestetty.Length)
    {
        if(jarjestetty[i-1] < jarjestetty[i])
        {
            int temp = jarjestetty[i];
            jarjestetty[i] = jarjestetty[i - 1];
            jarjestetty[i - 1] = temp;
            i = 1;
        } else
        {
            i++;
        }
    }
    return jarjestetty;
}
}

```



Kuva 5: Tehtävän 4 pistejakauma.