

ITKP102 Ohjelmointi 1 (6 op), arvosteluraportti

Tentaattori: Antti-Jussi Lakanen

9. kesäkuuta 2023

Yleistä

Tentti¹ oli pistekeskiarvon XX,X (keskihajonta X,X) perusteella vaikeudeltaan XX. (Tietoja ei näytetä vähäisen osallistujamäärän vuoksi.) Huomaa, että demopisteet on laskettu tentin päälle, ja arvosana lasketaan vasta sen jälkeen. Opiskelijan omat tehtävät ovat nähtävissä TIMissä alkuperäisellä tenttisivulla. Uusintojen ajankohdat löydät Sisusta.

Tehtävä	Teki	Keskiarvo	Automaattinen arvostelu / Tarkastaja
T1	X	X	Antti-Jussi Lakanen
T2	X	X	Antti-Jussi Lakanen
T3	X	X	Antti-Jussi Lakanen
T4	X	X	Antti-Jussi Lakanen
Yht		XX,x	(ennen demopisteitä)

¹<http://users.jyu.fi/~anlakane/ohjelmointi1/tentit/2023-06-09-tentti3-arvosteluraportti.pdf>



JYVÄSKYLÄN YLIOPISTO

Arvosteluasteikko ja arvosanajakauma

Arvolause	Pistemäärä (alaraja on inklusiivinen, yläraja eksklusiivinen)
5	24
4	21
3	18
2	15
1	12

(Arvosanajakauma on piilotettu vähäisen osallistujamäärän vuoksi.)

Tehtävä 1 (6 p.)

(Arvosanjakauma on piilotettu vähäisen osallistujamäärän vuoksi.)

Tee funktio `MontakoVokaalia`, joka palauttaa merkkijonossa olevien vokaalien määrän. Funktio ottaa yhden parametrin (merkkijonon) ja palauttaa määrän kokonaislukuina.

Esimerkiksi kutsu `MontakoVokaalia("antti-jussi lakanen")` palauttaa 7.

Voit olettaa, että parametrina saatava merkkijono ei sisällä suuraakkosia.

Malliratkaisu

Dokumentaatiot on jätetty tässä pois tilan säästämiseksi.

Tapa 1:

```
public static int MontakoVokaalia(string jono)
{
    int montako = 0;
    string merkit = "aeiouyää";
    foreach (char c in jono)
    {
        if (merkit.Contains(c))
        {
            montako++;
        }
    }
    return montako;
}
```

Tapa 2:

```
public static int MontakoVokaalia(string jono)
{
    if (jono.Length == 0) return 0;
    int montako = 0;
    string merkit = "aeiouyää";
    for (int i = 0; i < jono.Length; i++)
    {
        char merkki = jono[i];
        if (merkit.IndexOf(merkki) >= 0)
        {
            montako++;
        }
    }
    return montako;
}
```

Tehtävä 1 b (3 p.)

Tee funktio `MontakoMerkkia`. Funktio ottaa merkkijonon (`jono`), sekä toisen merkkijonon (`etsittavat`). Funktio palauttaa, kuinka moni merkkijonossa `jono` olevista merkeistä löytyy `etsittavat`-jonosta.

Esimerkkejä

- `MontakoMerkkia("antti-jussi lakanen", "aeiouyää")` palauttaa 7, sillä ensimmäisessä jonossa olevat a, i, u, i, a, a, e löytyvät toisesta jonosta.
- `MontakoMerkkia("antti-jussi lakanen", "xyz")` palauttaa 0.
- `MontakoMerkkia("antti-jussi lakanen", "jjj")` palauttaa 1.

Malliratkaisu

Tapa 1:

```
/// <summary>
/// Kuinka moni merkkijonossa olevista merkeistä löytyy toisesta jonosta.
/// </summary>
/// <param name="jono">Tutkittava merkkijono.</param>
/// <param name="etsittavat">Merkit, joiden joukosta etsitään.</param>
/// <returns>Kuinka monta merkkiä löytyi.</returns>
public static int MontakoMerkkia(string jono, string etsittavat)
{
    int montako = 0;
    foreach (char c in jono)
    {
        if (etsittavat.Contains(c))
            montako++;
    }
    return montako;
}
```

Tapa 2:

```
public static int MontakoMerkkia(string jono, string etsittavat)
{
    int montako = 0;

    for (int i = 0; i < jono.Length; i++)
    {
        for (int j = 0; j < etsittavat.Length; j++)
        {
            if (jono[i] == etsittavat[j])
            {
                montako++;
            }
        }
    }
}
```

```

    }
}
return maara;
}

```

Tehtävä 2 (6 p.)

(Arvosanjakauma on piilotettu vähäisen osallistujamäärän vuoksi.)

Tee funktio `PositiivistenSumma`, joka palauttaa kokonaislukutaulukon positiivisten lukujen summan. Esimerkki:

```

int[] luvut = { 1, -4, 7, 12 };
int summa = PositiivistenSumma(luvut);

```

Näiden lauseiden suorittamisen jälkeen `summa`-muuttujan arvo on 20, koska $1 + 7 + 12 = 20$.

Vaativuudet: Jos mitään summattavaa ei ole, tulos on 0.

Malliratkaisu

Dokumentaatiot on jätetty tässä pois tilan säästämiseksi.

```

public static int PositiivistenSumma(int[] taulukko)
{
    int summa = 0;
    for (int i = 0; i < taulukko.Length; i++)
    {
        if (taulukko[i] > 0)
            summa = summa + taulukko[i];
    }
    return summa;
}

```

Tehtävä 3 (6 p.)

(Arvosanjakauma on piilotettu vähäisen osallistujamäärän vuoksi.)

Tee funktio `Toista`. Funktio ottaa kaksi parametria: toistettavan merkkijonon (tyypiltään `string`), sekä kokonaisluvun joka kertoo kuinka monta kertaa peräkkäin merkkijono toistetaan. Funktio palauttaa merkkijonon (tyypiltään `string`).

Esimerkkejä

- `Toista("A", 3)` palauttaa `"AAA"`
- `Toista("A bc", 4)` palauttaa `"A bcA bcA bcA bc"`
- `Toista("", 9)` palauttaa `""`
- `Toista("Abc", 0)` palauttaa `""`

Vaatimukset

- Tulosmerkkijonon rakentelu tulee tehdä käyttäen StringBuilder-oliota. Mikäli tulostusmerkkijono rakennellaan käyttäen string-olioita, vähennetään pisteitä.
- Funktio tulee toimia kaikenlaisilla merkkijonoilla (ensimmäinen parametri).
- Voit olettaa että kokonaisluku (toinen parametri) on aina vähintään 0.

Malliratkaisu

```
public static string Toista(string jono, int kertoja)
{
    StringBuilder lopullinen = new StringBuilder();
    for (int i = 0; i < kertoja; i++)
    {
        lopullinen.Append(jono);
    }
    return lopullinen.ToString();
}
```

Tehtävä 4 (6 p.)

(Arvosanajakauma on piilotettu vähäisen osallistujamäärän vuoksi.)

Tee aliohjelma TulostaLuvut, joka ottaa parametreina kaksi kokonaislukua. Ensimmäinen parametri on pienin tulostettava luku ja toinen parametri on suurin tulostettava luku. Aliohjelma ei palauta arvoa.

Esimerkkejä:

Kutsuttaessa TulostaLuvut(1, 5) tuloste on

```
1 2 3 4 5
2 3 4 5
3 4 5
4 5
5
```

Kutsuttaessa TulostaLuvut(-2, 5) tuloste on:

```
-2 -1 0 1 2 3 4 5
-1 0 1 2 3 4 5
0 1 2 3 4 5
1 2 3 4 5
2 3 4 5
3 4 5
4 5
5
```

Malliratkaisu

```
public static void Tulosta(int alku, int loppu)
{
    for (int i = alku; i <= loppu; i++)
    {
        for (int j = i; j <= loppu; j++)
        {
            Console.Write(j + " ");
        }
        Console.WriteLine();
    }
}
```