

# ITKP102 Ohjelmointi 1 (6 op), arvosteluraportti

Tentaattori: Antti-Jussi Lakanen

24. toukokuuta 2023

## Yleistä

Tentti oli pistekeskiarvon 15,0 (keskihajonta 4,9) perusteella aavistuksen keskimääräistä helpompi. Demohyvitysten kanssa keskiarvo oli 18,4. 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.

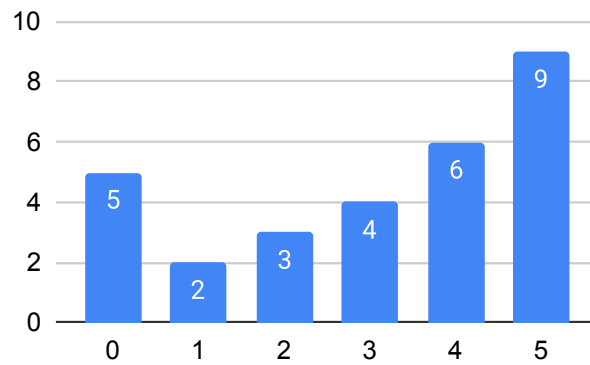
Tehtävä	Teki	Keskiarvo	Automaattinen arvostelu / Tarkastaja
T1	27	5,1	Automaattinen
T2	29	4,4	Automaattinen
T3	29	4,0	Automaattinen
T4	27	2,0	Antti-Jussi Lakanen
<b>Yht</b>		<b>15,0</b>	(ennen demopisteitä)



JYVÄSKYLÄN YLIOPISTO

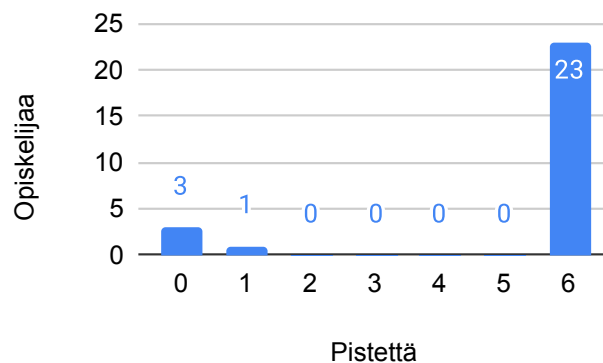
## Arvosteluasteikko ja arvosanajakauma

Arvolause	Pistemäärä (alaraja on inkluusiivinen, yläraja eksklusiivinen)
5	22
4	20
3	17
2	15
1	12



Tentin arvosanajakauma, kun demopisteet on laskettu tenttipisteisiin mukaan.

## Tehtävä 1 (6 p.)



Tehtävän 1 pistejakauma.

Tee funktio `SummaaTaulukko`, joka ottaa parametrina kokonaislukutaulukon ja palauttaa lukujen summan seuraavan algoritmin mukaisesti:

- taulukon parillisissa indekseissä olevat luvut lisätään summaan
- taulukon parittomissa indekseissä olevat luvut vähennetään summasta

Esimerkiksi syötteellä `[5, 3, 7, -2]` tulos olisi 11, koska  $5 - 3 + 7 - (-2) = 11$ .

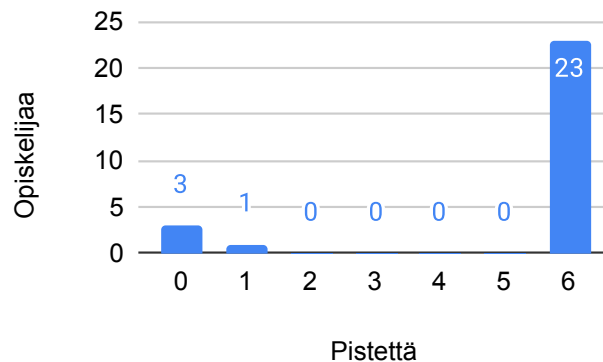
Tehtävässä on luokka ja pääohjelma valmiina, mutta pääohjelma sisältää tarkistuskoodin ja on piilotettu tarkoituksella. Kirjoita palautuslaatikkoon funktio esittelyriveineen ja dokumentaatiokommentteineen. Halutessasi voit kirjoittaa `ComTest`-testejä.

### Malliratkaisu

```
/// <summary>
/// Laskee kokonaislukutaulukon luvuista summan siten, että parillisissa
/// indekseissä olevat luvut lisätään summaan ja parittomissa olevat
/// luvut vähennetään summasta.
/// </summary>
/// <param name="luvut">Kokonaislukutaulukko</param>
/// <returns>Summa edellä mainitulla tavalla</returns>
public static int SummaaTaulukko(int[] luvut)
{
    int summa = 0;
    for (int i = 0; i < luvut.Length; i++)
    {
        if (i % 2 == 0) summa += luvut[i];
        else summa -= luvut[i];
    }
    return summa;
}
```

Arvostelu pohjautui automaattiseen arvioon Liitteessä A Taulukossa 1 esitettyjen arvostelukohteiden/testitapausten kautta. Automaattisilla arvostelukohteilla on tärkeysjärjestys: Mikäli ratkaisu ei toteuta ylempänä olevaa kohdetta, ei sitä alempia arvostelukohteita oteta huomioon.

## Tehtävä 2 (6 p.)



Tehtävän 1 pistejakauma.

1. Mikä seuraavista väitteistä pitää paikkansa C#-kielen ohjelmointikäytänteiden perusteella?
  - (a) Muuttujan nimessä voi olla puolipiste.
  - (b) Muuttujan nimessä voi olla numeroita.
  - (c) Muuttujan nimi on oltava sama kuin aliohjelman nimi.
  - (d) Muuttujan nimessä toinen ja sitä seuraavat sanat aloitetaan pienaakkosella.
2. Mikä seuraavista väitteistä pitää paikkansa?
  - (a) Sijoituslause palauttaa aina bool-tyyppisen arvon.
  - (b) Ohjelmassa on oltava aina vähintään yksi sijoituslause.
  - (c) Muuttujan arvon muuttaminen tapahtuu aina sijoitusoperaattorilla '='.
  - (d) Muuttujaan sijoitettava lausekkeen arvo voi koostua vakioista, muuttujien arvoista tai laskutoimituksista.
3. Mikä seuraavista väitteistä pitää paikkansa C#-kielen ohjelmointikäytänteiden perusteella?
  - (a) Funktion versiohistoria kirjataan dokumentaatiokommentteihin.
  - (b) Jokaisen aliohjelman dokumentaatiokommentteihin tulee returns-kenttä.
  - (c) Dokumentaatiokommentit kirjoitetaan riveille joiden alussa on kolme kautaviivaa.
  - (d) Algoritmin toteutus kuvataan summary-kenttään mahdollisimman yksityiskohtaisesti.

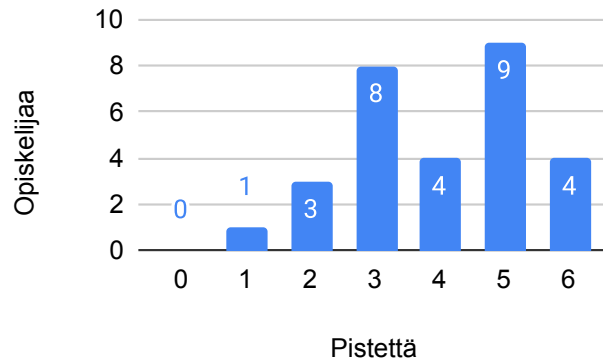
4. Oletetaan, että ohjelma on käynnistetty debuggaustilassa ja sen suoritus on keskeytetty sinun tekemäsi funktion sisälle asetettuun keskeytyskohtaan. Mikä seuraavista väittämistä pitää paikkansa? Huomautus: Ilmaisulla “debuggerin paikallisten muuttujien tarkastelunäkymä” tarkoitetaan Riderin debuggaustilan Threads & Variables -näköä, ja Visual Studion debuggaustilan Locals-näköä.
- (a) Muuttujan arvoksi voidaan debuggerin paikallisten muuttujien tarkastelunäkymässä sijoittaa funktiokutsu.
  - (b) Watches-paneelissa tehty muuttujan arvon muutos säilyy, kun ajo lopetetaan ja aloitetaan uudestaan.
  - (c) Muuttujien tyyppejä voidaan muuttaa debuggerin paikallisten muuttujien tarkastelunäkymästä käsin.
  - (d) Main-pääohjelmassa olevien muuttujien arvoja voidaan aina tarkastella debuggerissa.
5. Mikä seuraavista väitteistä pitää paikkansa?
- (a) Tyhjän taulukon pituutta voi muuttaa taulukon luomisen jälkeen.
  - (b) Taulukon alkiot sijaitsevat indekseissä 0...Length.
  - (c) Taulukko voi olla enintään 2-ulotteinen.
  - (d) Taulukon a pituus (a.Length) voidaan päätellä seuraavan lauseen suorittamisen jälkeen: `int[] a = new int[] { };`
6. Mikä seuraavista väitteistä pitää paikkansa? Aliohjelman esittelyriville kirjoitetaan...
- (a) paluuarvon tyyppi.
  - (b) aliohjelman dokumentaatiokomentit.
  - (c) tieto siitä, onko aliohjelma rekursiivinen.
  - (d) aliohjelman käyttämien attribuuttien nimet.

## Malliratkaisu

1B 2D 3C 4A 5D 6A

## Tehtävä 3 (6 p.)

1. Mikä seuraavista muuttujista sisältää suurimman arvon?
- (a) `int a = 4 % 2;`
  - (b) `int b = 5 / 3;`
  - (c) `double c = 0.75 + 0.75;`
  - (d) `int d = int.MaxValue / int.MaxValue;`



Tehtävän 3 pistejakauma.

2. Mikä seuraavista väitteistä pitää paikkansa? Huomautus: for-silmukan määrittelyrivi koostuu for-sanasta, kaarisulkeista sekä kaarisulkeiden sisään kirjoitettavista lauseista ja lausekkeista. Eräs for-silmukka voisi näyttää seuraavalta:

```
for (int i = 0; i < 4; i++) // tämä rivi on nimeltään määrittelyrivi
{
    // aaltosulkeiden sisäpuolinen alue on nimeltään runko-osa
}
```

Toistoehto on silmukan määrittelyrivin keskimmäinen lauseke, yllä esimerkissä  $i < 4$ .

- (a) for-silmukalla voidaan tehdä ns. ikuinen silmukka.
- (b) for-silmukassa tulee aina jossain kohdassa lukea  $i++$  tai  $i--$ .
- (c) for-silmukan runko-osaa toistetaan niin kauan kuin toistoehdon arvo on epätosi.
- (d) for-silmukalla tietorakenteen läpikäynti tulee aloittaa ensimmäisestä alkios-  
ta.

3. Oheinen koodi tulostaa luvut 1, 2 ja 3.

```
public static void Main()
{
    int[] luvut = { 1, 2, 3 };
    F(luvut);
    Console.WriteLine(luvut[0]);
    Console.WriteLine(luvut[1]);
    Console.WriteLine(luvut[2]);
}

public static void F(int[] taulukko)
{
    taulukko = new[] { 4, 5, 6, 7 };
}
```

Miksi koodi ei tulosta 4, 5, 6 ja 7? Koska...

- (a) parametrin (taulukko) nimi on eri kuin pääohjelman luvut.
- (b) luvut on paikallinen muuttuja Main-pääohjelmassa.
- (c) funktiossa F luodaan paikalliseen taulukko-muuttujaan uusi olio.
- (d) funktiossa F luodaan taulukko, joka on eri pituinen kuin Main-pääohjelman taulukko.

4. Olkoon meillä seuraava C#-ohjelma. Dokumentaatiokommentit on jätetty tarkoituksellisesti pois tilan säästämiseksi.

```
using System;

public class T3_4
{
    public static void Main()
    {
        int tulos = Summa(new int[] { 1, 3, 3, 7 });
        bool tulostetaanko = true;
        if (true)
        {
            Console.WriteLine(Summa(new int[] { tulos, tulos }));
        }
    }

    public static int Summa(int[] taulukko)
    {
        int summa = 0;
        foreach (int luku in taulukko)
        {
            summa += luku;
        }
        return summa;
    }
}
```

Mikä seuraavista väitteistä pitää paikkansa?

- (a) Ohjelma ei tulosta mitään.
- (b) Ohjelmassa on yksi aliohjelma.
- (c) Ohjelmassa on muuttujia, joita ei käytetä.
- (d) Summa-funktio laskee taulukon summan väärin.

5. Mikä seuraavista väitteistä pitää paikkansa?

- (a) Muuttujaan täytyy sijoittaa arvo määrittelyn yhteydessä.
- (b) Muuttujaan täytyy sijoittaa arvo ennen muuttujan käyttämistä.

- (c) Muuttujaan voi sijoittaa arvon vain käyttäjältä kysymisen seurauksena.
- (d) Jos pääohjelmassa olevia muuttujia \*ei\* käytä ohjelman aikana, siitä aiheutuu käännösvirhe.

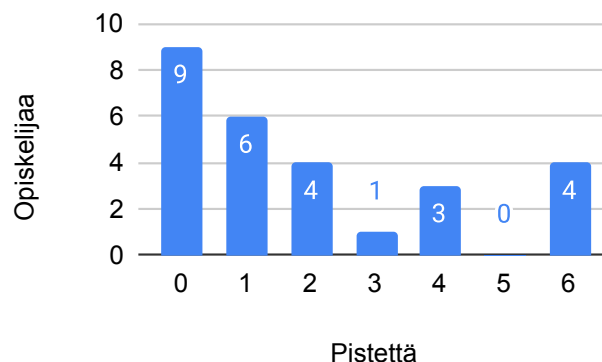
6. Mikä seuraavista väitteistä pitää paikkansa?

- (a) Taulukossa voi olla enintään kaksi ulottuvuutta.
- (b) StringBuilder-tilaukion alkioiksi voi sijoittaa myös string-tilion.
- (c) Kaksiulotteinen string-tiluukko voidaan käydä läpi foreach-tilmukalla.
- (d) string-tiluukon alkion sisällön muuttaminen aiheuttaa ajonaikaisen virheen.

## Malliratkaisu

1C 2A 3C 4C 5B 6C

## Tehtävä 4 (6 p.)



Tehtävän 4 pistejakauma.

Tee funktio Tunnusgeneraattori, joka luo henkilölle käyttäjätunnuksen. Funktio ottaa kaksi parametria:

1. henkilön koko nimen (string), sekä
2. lista olemassa olevista käyttäjätunnuksista (List<string>).

Voit olettaa, että henkilön nimi on aina muodossa "Etunimi (tai -nimet) Sukunimi", ja nimien välissä on yksi välilyönti. Funktio palauttaa näiden tietojen perusteella käyttäjätunnuksen (string) seuraavasti:

1. Ensimmäinen merkki on ensimmäisen etunimen ensimmäinen kirjain pientäakkosina.
2. Tämän jälkeen sukunimestä 7 ensimmäistä merkkiä ilman välimerkkejä, pientäakkosina.



3. Jos vaiheiden 1 ja 2 seurauksena syntynyt tunnus on jo olemassa, poistetaan tunnuksen viimeinen merkki, ja lisätään ensimmäinen vapaana oleva numero: 1, 2 jne.

**Esimerkki:** Olkoon meillä seuraava lista käyttäjätunnuksia.

```
List<string> tunnuksset =  
    new List<string>() { "alakanen", "vlappala", "j itkone" };
```

Tällöin alla olevat ComTest-syntaksilla kirjoitetut testitapaukset olisivat totta:

```
Tunnusgeneraattori("Irmeli Möttönen", tunnuksset) === "imottone";  
Tunnusgeneraattori("Alma Lakanen", tunnuksset) === "alakane1";  
Tunnusgeneraattori("Åke Päiviö Jarmo Ylä-Myyräkki", tunnuksset) === "aylamyyr";
```

Huomaa, että Tunnusgeneraattori-funktiossa *\*ei\** tule luoda listaa käyttäjätunnuksista, eikä siihen lisätä uutta tunnusta. Lista annetaan parametrina, eikä sitä tule muuttaa funktiossa.

**Vinkki:** Saatat tarvita (mutta et välttämättä tarvitse) alla olevia funktioita:

- String.ToLower
- String.Replace
- String.Split
- String.Substring
- List.Contains

## Malliratkaisu

```
/// <summary>  
/// Käyttäjätunnusgeneraattori luo henkilölle käyttäjätunnuksen:  
/// etunimen ensimmäinen merkki + sukunimestä enintään seitsemän merkkiä.  
/// </summary>  
/// <param name="kokonimi">Koko nimi</param>  
/// <param name="tunnukset">Olemassa olevat tunnukset</param>  
/// <returns>Käyttäjätunnus</returns>  
/// <example>  
/// <pre name="test">  
/// List<string> tunnuksset =  
/// new List<string>() { "alakanen", "vlappala", "j itkone" };  
/// Tunnusgeneraattori("Irmeli Möttönen", tunnuksset) === "imottone";  
/// tunnuksset.Add("imottone");  
/// Tunnusgeneraattori("Alma Lakanen", tunnuksset) === "alakane1";  
/// tunnuksset.Add("alakane1");  
/// Tunnusgeneraattori("Antero von Lakanen", tunnuksset) === "alakane2";  
/// Tunnusgeneraattori("Åke Päiviö Jarmo Ylä-Myyräkki", tunnuksset) === "aylamyyr";  
/// Tunnusgeneraattori("Åke Päiviö Jarmo Malm", tunnuksset) === "amalm";  
/// tunnuksset.Add("amalm");
```

```

/// // Tämän toteutumista ei vaadittu täysiin pisteisiin
/// Tunnusgeneraattori("Antti Malm", tunnuksset) === "amalm1";
/// </pre>
/// </example>
public static string Tunnusgeneraattori(string kokonimi, List<string> tunnuksset)
{
    // Muutetaan koko merkkijono pienaakkosiksi, niin jatkossa on
    // vähän helpompi käsitellä
    kokonimi = kokonimi.ToLower();

    // Napataan väliviiva pois ja muutetaan skandit tavallisiksi aakkosiksi.
    kokonimi = kokonimi.Replace("-", "");
    kokonimi = kokonimi.Replace("ä", "a");
    kokonimi = kokonimi.Replace("ö", "o");
    kokonimi = kokonimi.Replace("å", "a");

    // Otetaan etunimen ensimmäinen merkki
    string etunimi = kokonimi[0].ToString();

    // Paloitellaan nimi merkkijonotaulukoksi
    string[] nimenPalat = kokonimi.Split(" ");

    // Otetaan sukunimi (paloitellun taulukon viimeinen alkio)
    string sukunimi = nimenPalat[nimenPalat.Length - 1];

    // Kuinka monta merkkiä sukunimestä otetaan tunnukseen?
    // Jos sukunimi on alle 7 merkkiä pitkä, se otetaan kokonaan
    int merkkejaSukunimesta = Math.Min(sukunimi.Length, 7);
    string sukunimenPala = sukunimi.Substring(0, merkkejaSukunimesta);
    string tunnus = etunimi + sukunimenPala;

    int numero = 1;
    while (tunnuksset.Contains(tunnus))
    {
        // Tunnusta lyhennetään numeroa varten vain jos se on tarpeen, ts.
        // jos sukunimen pituus on alle 7 merkkiä.
        int lyhennetytTunnuksenPituus = Math.Min(tunnus.Length, 7);
        tunnus = tunnus.Substring(0, lyhennetytTunnuksenPituus) + numero;
        numero++;
    }
    return tunnus;
}

```

## A Arviointikohteet

Syöte	Tulos	Selitys	Virhe	Pistemäärä
{ 8, 0, 3, 0, 4, 0, 10 }	25	Funktio osaa tavallisen summan laskemisen.	Funktio ei osaa laskea tavallista summaa.	1
{ 5, 3, 7, -2 }	11	Funktio toimii tehtävänannossa annetulla taulukolla	Funktio ei toimi oikein tehtävänannossa annetulla taulukolla.	1
{ -8, 4, 3, 2, 6, 10, 1 }	-14	Funktio toimii parittoman pituisella taulukolla.	Funktio ei toimi oikein parittoman pituisella taulukolla.	1
{ 5 }	5	Funktio toimii yhden alkion mittaisella taulukolla.	Funktio ei toimi oikein yhden alkion mittaisella taulukolla.	1
{ 0, 0, 0, 0, 0 }	0	Funktio toimii pelkkiä nollia sisältävällä taulukolla.	Funktio ei toimi oikein pelkkiä nollia sisältävällä taulukolla.	1
{ }	0	Funktio toimii tyhjällä taulukolla.	Funktio ei toimi oikein tyhjällä taulukolla.	1

Taulukko 1: Tehtävän 1 arviointikohteet

Syöte1	Syöte2	Tulos	Selitys	Virhe	Pistemäärä
“Irmeli Korhonen”	{ “alakanen”, “vlappala”, “j itkone” }	“ikorhone”	Tunnus on luotu oikein. Vastaavaa tunnusta ei löydy aiemmista tunnuksista.	Luotu tunnus on virheellinen.	1
“Alma Lakenen”	{ “alakanen”, “vlappala”, “j itkone” }	“alakane1”	alakanen löytyy jo aiemmista tunnuksista, oikea tunnus on alakane1.	Luotu tunnus on virheellinen.	1
“Irmeli Möttönen”	{ “alakanen”, “vlappala”, “j itkone” }	“imottone”	Skandit on huomioitu tunnusta luodessa.	Skandeja ei ole huomioitu tunnuksessa.	1
“Åke Jarmo Päiviö Ylä-Myyräkki”	{ “alakanen”, “vlappala”, “j itkone” }	“aylamyyr”	Sekä ääkköset että välimerkit on huomioitu.	Ääkkösiä ja/tai välimerkkejä ei ole huomioitu.	1
“Alma Lakenen”	{ “alakanen”, “vlappala”, “j itkone”, “alakane1” }	“alakane2”	alakane1 löytyy jo aiemmista tunnuksista, oikea tunnus on alakane2.	Luotu tunnus on virheellinen.	1
“Alma Lakenen”	{ “alakanen”, “vlappala”, “j itkone”, “alakane2” }	“alakane1”	alakane1 ei löydy aiemmista tunnuksista joten valitaan se.	Luotu tunnus on virheellinen.	1

Taulukko 2: Tehtävän 4 arviointikohteet