

Ohjelmointi 1 C#, kevät 2014, tentti

Tentaattori Antti-Jussi Lakanen

Tässä tentissä saa olla mukana omia muistiinpanoja yksi A4-arkillinen. Tentin valvojalla on oikeus tarvittaessa tarkastaa muistiinpanot (esimerkiksi tilanteessa, jossa tenttijä tekee kerralla enemmän kuin yhden tentin).

Valitse **neljä tehtävää** ja vastaa niihin (bonuksiin ei ole pakko vastata). Kokeesta voi saada enintään 24 pistettä (+ mahdolliset bonuspisteet). Keväällä 2014 kurssin tehneille lasketaan demohyvitykset tenttipisteisiin mukaan.

Tee jokainen tehtävä omalle konseptiarkille! (Tämä tarkistuksen nopeuttamiseksi.)

Kirjoittamasi luokat ja aliohjelmat pitää dokumentoida, kuten kurssilla on opetettu. Jos vastauksessa tarvitset jotakin funktiota .NET-APIsta (.NET-kirjasto) etkä muista tarkkaan mikä oli funktion/metodin nimi, niin kirjoita funktion esittely parametreineen ja kommentti siitä, mitä sen pitäisi tehdä. Luonnollisesti et saa itse keksiä funktiota, joita ei ole .NETissä (jos et sitten itse toteuta niitä). Vinkkejä, joita saatat tarvita tehtävissä, löytyy kokeen loppuosasta.

Tehtävä 1 (6 p.)

Vastaa kaikkiin kohtiin.

a) Mitä on kommentointi ja mitä eri tapoja kommentoinnille on? Miksi koodia kannattaa kommentoida? (2 p.)

b) Kirjoita dokumentaatio seuraavalle funktiolle. Asiaankuuluvien ComTest-testien kirjoittamisesta saa yhden bonuspisteen. (2 p. + 1 bonusp.)

```
public static int SummaaPituudet(String[] sanat)
{
    int summa = 0;
    for (int i = 0; i < sanat.Length; i++)
    {
        int pituus = sanat[i].Length;
        summa += pituus;
    }
    return summa;
}
```

c) Esittele vähintään kaksi loogista operaattoria ja kerro mitä kukin tekee. (1 p.)

d) Luku 23 binäärilukuna? Entä 00101010 muutettuna 10-järjestelmään? (1 p.)

Palaute tästä tehtävästä (valinnainen, tutkimustarkoituksiin & opetuksen kehittämiseen)

asteikko 1 = täysin eri mieltä, ... , 5 = täysin samaa mieltä

Tehtävä mittaa kurssin keskeisiä oppimistavoitteita	1	2	3	4	5
---	---	---	---	---	---

Tehtävä oli vaikea	1	2	3	4	5
--------------------	---	---	---	---	---

arvio omasta pistemäärästä (1-6 + bonus): _____

käytetty aika (minuuttia): _____

Tehtävä 2 (6 p.)

Jos listaamme kaikki ne lukua 10 pienemmät (nollaa suuremmat) luonnolliset luvut, jotka ovat jaollisia 3:lla tai 5:llä, saamme luvut 3, 5, 6, ja 9. Näiden lukujen summa on 23.

Tee ohjelma (siis luokka ja pääohjelma, aliohjelmaa/funktioita ei tässä tarvitse tehdä), joka laskee kaikkien 1000:a pienempien, 3:lla tai 5:llä jaollisten lukujen summan. Ohjelma tulostaa lopuksi tämän summan. Muista dokumentaatio.

Palaute tästä tehtävästä (valinnainen, tutkimustarkoituksiin & opetuksen kehittämiseen)

asteikko 1 = täysin eri mieltä, ... , 5 = täysin samaa mieltä

Tehtävä mittaa kurssin keskeisiä oppimistavoitteita 1 2 3 4 5

Tehtävä oli vaikea 1 2 3 4 5

arvio omasta pistemäärästä (1-6): _____

käytetty aika (minuuttia): _____

Tehtävä 3 (6 p.)

a) Alla on pääohjelma, jossa lasketaan taulukon ei-negatiivisten lukujen summa, ja jos luku sattuu olemaan negatiivinen, niin summaan lisätään jotain muuta - päättelä alla olevasta koodista, mitä. Ohjelmassa on turhaa toistoa. Poista turha toisto käyttäen silmukkaa. (3 p.)

```
public static void Main()
{
    int[] luvut = { 2, 4, -6, 3, 1 };
    int summa = 0;
    if (luvut[0] >= 0) summa += luvut[0];
    else summa += 10;
    if (luvut[1] >= 0) summa += luvut[1];
    else summa += 11;
    if (luvut[2] >= 0) summa += luvut[2];
    else summa += 12;
    if (luvut[3] >= 0) summa += luvut[3];
    else summa += 13;
    if (luvut[4] >= 0) summa += luvut[4];
    else summa += 14;

    Console.WriteLine(summa);
}
```

b) Refaktoroi alla oleva koodi siten, että arvotun kentän tiedot tulostetaan omassa aliohjelmassa, jolle annetaan parametrina arvottu kenttännumero (kokonaisluku), ja toinen kokonaisluku sekä siihen liittyvä kentän "esittelyteksti" merkkijonona. Kutsu sitten tekemääsi aliohjelmaa pääohjelmasta käsin.

```
public static void Main()
{
    Random r = new Random();
    int arvottuKentta = r.Next(1, 4); // antaa satunnaisluvun väliltä 1-3.

    if (arvottuKentta == 1)
    {
        Console.WriteLine("Tervetuloa peliin, olet kentässä numero 1.");
        Console.WriteLine("Edessäsi on ovi.");
    }
}
```

```
if (arvottuKentta == 2)
{
    Console.WriteLine("Tervetuloa peliin, olet kentässä numero 2.");
    Console.WriteLine("Edessäsi on järvi.");
}
if (arvottuKentta == 3)
{
    Console.WriteLine("Tervetuloa peliin, olet kentässä numero 3.");
    Console.WriteLine("Olet metsässä. On pimeää.");
}
}
```

Main-pääohjelmaan ei saa muutosten jälkeen jäädä yhtään if-lausetta.

Palaute tästä tehtävästä (valinnainen, tutkimustarkoituksiin & opetuksen kehittämiseen)

asteikko 1 = täysin eri mieltä, ... , 5 = täysin samaa mieltä

Tehtävä mittaa kurssin keskeisiä oppimistavoitteita 1 2 3 4 5

Tehtävä oli vaikea 1 2 3 4 5

arvio omasta pistemäärästä (1-6): _____

käytetty aika (minuuttia): _____

Tehtävä 4 (6 p.)

Tee funktio, joka ottaa parametrina kokonaislukutaulukon. Funktio palauttaa true, jos taulukosta löytyy sama alkio kaksi kertaa tai useammin. Vastaavasti false, jos kukin alkio esiintyy vain kerran.

Ohje/vinkki: Voit muuttaa taulukossa olevien alkoiden järjestystä.

Bonus (+ 2 p.) Tee toinen funktio, jossa teet saman asian koskematta alkoiden järjestykseen.

Kirjoita lyhyt analyysi siitä, kumpi funktioista on nopeampi ja miksi.

Palaute tästä tehtävästä (valinnainen, tutkimustarkoituksiin & opetuksen kehittämiseen)

asteikko 1 = täysin eri mieltä, ... , 5 = täysin samaa mieltä

Tehtävä mittaa kurssin keskeisiä oppimistavoitteita 1 2 3 4 5

Tehtävä oli vaikea 1 2 3 4 5

arvio omasta pistemäärästä (1-6 + bonus): _____

käytetty aika (minuuttia): _____

Tehtävä 5 (6 p.)

Tee funktio `SuurinViidenOsajono`, joka antaa merkkijonon suurimman viiden perättäisen numeron (välillä 0-9) summan. Esimerkiksi alla oleva pääohjelma ...

```
public static void Main()
{
    String jono = "7316717653";
    int suurinSumma = SuurinViidenOsajono(jono);
    Console.WriteLine(suurinSumma);
}
```

... tulostaisi luvun 27, sillä $6 + 7 + 1 + 7 + 6 = 27$. Oletetaan, että merkkijono sisältää vain numeroita, joten mitään poikkeuksia ei tarvitse ottaa huomioon. Alla ComTest-testit, jotka funktion pitää läpäistä. **BONUS (+ 1 p.)** Jos kaksi alintakin testiä menee läpi, niin saat yhden lisäpisteen.

```
/// Teht1.SuurinViidenOsajono("7316717653") === 27;
/// Teht1.SuurinViidenOsajono("012345") === 15;
/// Teht1.SuurinViidenOsajono("43210123") === 10;
/// Teht1.SuurinViidenOsajono("43210") === 10;
/// Teht1.SuurinViidenOsajono("4321") === 10; // BONUS
/// Teht1.SuurinViidenOsajono("") === 0; // BONUS
```

Palaute tästä tehtävästä (valinnainen, tutkimustarkoituksiin & opetuksen kehittämiseen)

asteikko 1 = täysin eri mieltä, ... , 5 = täysin samaa mieltä

Tehtävä mittaa kurssin keskeisiä oppimistavoitteita	1	2	3	4	5
---	---	---	---	---	---

Tehtävä oli vaikea	1	2	3	4	5
--------------------	---	---	---	---	---

arvio omasta pistemäärästä (1-6 + bonus): _____

käytetty aika (minuuttia): _____

Huomautus

Muistithan, että tässä tentissä piti tehdä **enintään neljä tehtävää**. Jos jostain syystä kuitenkin palautat viisi tehtävää, niin tarkastajalla ei ole muuta vaihtoehtoa kuin hylätä se tehtävä, josta olet saanut korkeimman pistemäärän.

Palauta lunttilappu tutkimustarkoituksiin

Ystävällisesti pyydän, että palauttaisit valvojalle myös lunttilapun jättäessäsi tenttipaperit. Jos palautat lunttilapun, niin suostut että sitä voidaan käyttää tutkimustarkoituksiin -- luonnollisesti nimettömänä ja ilman tunnistetietoja. Voit jättää myös lapun palauttamatta ja ottaa sen mukaasi tenttialista.

Vinkit

Merkkijonosta yhden merkin ottaminen

Voit lukea String-merkkijonosta yhden merkin, esimerkiksi ensimmäisen, seuraavasti.

```
String jono = "Ville";  
char ekaKirjain = jono[0];
```

Samasta jonosta viimeisen merkin saat näin.

```
char vikaKirjain = jono[jono.Length-1];
```

Substring

Substring antaa merkkijonon osan. Metodista on mm. kaksi seuraavaa versiota.

Substring(Int32): Palauttaa jonon alkaen annetusta indeksistä.

```
String henkilo = "Ville Virtanen";  
String sukunimi = henkilo.Substring(6);  
Console.WriteLine(sukunimi); // "Virtanen"
```

Substring(Int32, Int32): Ensimmäinen parametri on palautettavan merkkijonon ensimmäisen merkin indeksi ja toinen parametri palautettavien merkkien määrä.

```
String henkilo = "Ville Virtanen";  
String etunimi = henkilo.Substring(0, 5); // first name  
String sukunimenOsa = henkilo.Substring(6, 5); // part of the last name  
Console.WriteLine(etunimi + sukunimenOsa); // "VilleVirta"
```

Numero kirjaimeksi

Yhden numeron (joka on char-merkki) saa muutettua kokonaisluvuksi joko seuraavasti (helpompi ja koneen näkökulmasta tehokkaampi tapa).

```
char merkki = '9';  
int luku = merkki - '0';
```

tai int.Parse-metodin avulla.

```
String k1 = "9";  
int luku = int.Parse(k1);    // nyt: luku == 9
```

Huom! `int.Parse` ottaa parametrinaan `String`-merkkijonon. Jos haluat muuttaa yhden `char`-merkin, esimerkiksi `'9'`, merkkijonoksi, voit tehdä sen kirjoittamalla esimerkiksi näin.

```
char merkki = '9';  
int luku = int.Parse("" + merkki);
```

Array.Sort

Järjestää taulukon. Alla esimerkki.

```
int[] luvut = { 5, 3, 4, 1 };  
Array.Sort(luvut); // luvut-tilukko nyt: 1, 3, 4, 5.
```

% (jakojäännös) -operaattori

Jakojäännösoperaattori `%` laskee jakojäännöksen, joka saadaan kun merkin vasemmalla puolella oleva luku jaetaan oikealla puolella olevalla luvulla. Alla esimerkki.

```
public class Tentti  
{  
    public static void Main(string[] args)  
    {  
        Console.WriteLine(5 % 2); // tulostaa: 1, koska 5/2 on 2, jää 1  
        Console.WriteLine(4 % 2); // tulostaa: 0  
        Console.WriteLine(11 % 7); // tulostaa: 4  
        Console.WriteLine(11 % 11); // tulostaa: 0  
    }  
}
```