

# Ohjelmointi 1 C#, kevät 2012, 16.4.2012

Tentti (yliopisto-opiskelijat)  
Toinen välikoe (lukiolaiset)

## Yliopisto-opiskelijoille

Vastaa kaikkiin kysymyksiin (bonuksiin ei ole pakko vastata). Kokeesta voi saada enintään 24 pistettä (+ mahdolliset bonuspisteet). Päästäkseen läpi on saatava 12 pistettä.

## Lukiolaisille

Vastaa kysymyksiin **3 ja 4** (bonuksiin ei ole pakko vastata). Tehtäviin 1 ja 2 ei tarvitse vastata, eikä niistä saa pisteitä. Kokeesta voi saada enintään 12 pistettä (+ mahdolliset bonuspisteet). Päästäkseen läpi välikokeesta on saatava kuusi pistettä.

## Kaikille

Demohyvitykset lasketaan tenttipisteiden "päälle", ja sitten katsotaan tuleeko tarvittavat pisteet täyteen. Tenttiaika on 4 tuntia.

Tentin lopussa on vinkkejä ja käyttöesimerkkejä C#-kielen metodeista. Voit soveltaa niitä vastauksiisi. Luokat ja aliohjelmat pitää dokumentoida, kuten kurssilla on opetettu. ComTest-testien kirjoittaminen ei ole pakollista, mutta siitä voi saada lisäpisteitä, mikäli tehtävässä on niin sanottu.

Jos vastauksessa tarvitset jotakin funktiota API-kirjastosta (C#:n kirjasto) ja et 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 olemassa (jollat itse myös toteuta niitä).

## Tehtävä 1 - Aliohjelmat, taulukot, silmukat ja ehtolauseet (6 p.)

Tehtävänäsi on toteuttaa kaksi (funktio-)aliohjelmaa.

- Ensimmäinen aliohjelma saa parametrinaan merkkijonotaulukon, josta aliohjelma etsii pisimmän merkkijonon, ja palauttaa sen pituuden. (3 p.)
- Toinen aliohjelma saa parametrinaan kolme kokonaislukua, ja palauttaa niistä suurimman. (3 p.)

Aliohjelmia voisi käyttää pääohjelmasta käsin esimerkiksi näin:

```
public static void Main(string[] args)
{
    String[] nimet = {"Juho", "Simo", "Ville", "Antti-Jussi"};
    int pisimmanPituus = PisimmanSananPituus(nimet); // pisimmanPituus == 11
}
```

```
    int suurin = Suurin(2, -6, 12); // suurin == 12
}
```

Muistathan kirjoittaa myös tarvittavat dokumentaatiot.

Bonus: Kirjoita molempiin aliohjelmiin ComTest-testit. (+ 1 p.)

## Tehtävä 2 - Muuttujat (6 p.)

(a) Kerro mitä alla oleva pääohjelma tulostaisi ja pohdi miksi. (3 p.)

```
public static void Main(string[] args)
{
    int a = 4;
    int b = 5;
    double c = 6.0;
    double d = 8.0;

    Console.WriteLine(b / a);
    Console.WriteLine(c / d);
    Console.WriteLine(b % a);
    Console.WriteLine(b / a - 1);
}
```

(b) Vertaile muuttujatyyppejä `string` ja `StringBuilder` alla olevan pääohjelman avulla. Kirjoita muuttujien `nimi` ja `nimi2` arvot kunkin rivin suorittamisen jälkeen. Mitkä ovat muuttujien arvot pääohjelman lopussa, miksi? (3 p.)

```
public static void Main(string[] args)
{
    string nimi = "";
    nimi += "Aku";
    nimi += " ";
    nimi += "Ankka";
    nimi.Replace("Ankka", "Sorsa");

    StringBuilder nimi2 = new StringBuilder();
    nimi2.Append("Roope");
    nimi2.Append(" ");
    nimi2.Append("Ankka");
    nimi2.Replace("Ankka", "Sorsa");
}
```

## Tehtävä 3 - Määritelmiä (6 p.)

Alla on kuusi kohtaa joista jokaisesta saa yhden pisteen. Vastaa kohtiin lyhyesti enintään muutamalla virkkeellä.

1. Määrittele rekursio.
2. Esittele vähintään kolme lukujärjestelmää ja anna kustakin esimerkkiluku.
3. Mitä eroa on `while` ja `do-while` -silmukalla?
4. Mikä on tietotyyppi ja mihin sitä käytetään?
5. Esittele vähintään kolme loogista operaattoria ja kerro mitä kukin tekee.
6. Mitä eroa ja mitä yhteistä on dynaamisella tietorakenteella (lista) ja staattisella tietorakenteella (taulukko)?

## Tehtävä 4 (6 p.)

Tee allaolevaa dokumentaatiota vastaava ja ComTest-testitapaukset täyttävä funktio. Bonus: Täydennä funktiota siten (ks. ComTest-rivit, joissa lukee BONUS), että jos parametrina annetaan väärä päivämäärä (esim. 0.1. tai 31.2.) funktio palauttaa aina 0. (+ 1 p.)

```

/// <summary>
/// Palauttaa kahden päivämäärän väliin jäävien päivien lukumäärän.
/// Esimerkiksi päivien 1.1. (1. tammikuuta) ja 2.1. (2. tammikuuta) väliin jää yksi
/// päivä, 1.1. ja 1.2. väliin jää 31 päivää. Funktio ei ota huomioon vuosia, eli
/// esimerkiksi 31.12. ja 1.1. väliin jää yksi päivä.
/// </summary>
/// <param name="kkPituudet">Kuukausien päivien lukumäärä</param>
/// <param name="paiva1">Ensimmäisen päivämäärän päivännumero</param>
/// <param name="kk1">Ensimmäisen päivämäärän kuukauden numero</param>
/// <param name="paiva2">Jälkimmäisen päivämäärän päivännumero</param>
/// <param name="kk2">Jälkimmäisen päivämäärän kuukauden numero</param>
/// <returns>Päivämäärien väliin jäävien päivien lukumäärä.</returns>
/// <example>
/// <pre name="test">
/// int[] kkPituudet = new int[] { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };
/// Paivat.MontakoPaivaaValissa(kkPituudet, 1, 1, 2, 1) === 1;
/// Paivat.MontakoPaivaaValissa(kkPituudet, 1, 1, 1, 2) === 31;
/// Paivat.MontakoPaivaaValissa(kkPituudet, 31, 1, 1, 2) === 1;
/// Paivat.MontakoPaivaaValissa(kkPituudet, 1, 1, 31, 12) === 364;
/// Paivat.MontakoPaivaaValissa(kkPituudet, 31, 12, 1, 1) === 1;
/// Paivat.MontakoPaivaaValissa(kkPituudet, 31, 12, 31, 12) === 0;
/// Paivat.MontakoPaivaaValissa(kkPituudet, 1, 2, 31, 2) === 0; // (BONUS)
/// Paivat.MontakoPaivaaValissa(kkPituudet, 0, 1, 31, 1) === 0; // (BONUS)
/// </pre>
/// </example>

```

## Vinkit

### **String.Replace(String oldValue, String newValue)**

Returns a new string in which all occurrences of a specified string in the current instance are replaced with another specified string. If *newValue* is *Nothing*, all occurrences of *oldValue* are removed. This method does not modify the value of the current instance. Instead, it returns a new string in which all occurrences of *oldValue* are replaced by *newValue*.

## Parameters

### oldValue

Type: System.String  
The string to be replaced.

### newValue

Type: System.String  
The string to replace all occurrences of oldValue.

### Return Value

Type: System.String  
A string that is equivalent to the current string except that all instances of oldValue are replaced with newValue.

## **StringBuilder.Replace(String oldValue, String newValue)**

Replaces all occurrences of a specified string in this instance with another specified string. If newValue is *Nothing*, all occurrences of oldValue are removed.

## Parameters

### oldValue

Type: System.String  
The string to replace.

### newValue

Type: System.String  
The string that replaces oldValue, or Nothing.

### Return Value

Type: System.Text.StringBuilder  
A reference to this instance with all instances of oldValue replaced by newValue.