

ITKP102 Ohjelmointi 1 (6 op)

Tentaattori: Antti-Jussi Lakanen

12. huhtikuuta 2019

Tee kukin tehtävä omalle konseptiarkille. Noudata ohjelmointitehtävissä kurssin koodauskäytänteitä. Yksi A4-kokoinen lunttilappu on sallittu tässä tentissä.

Jos tämä tentti on sinulle syksyn 2018 kurssi-instanssin uusintatentti, niin laita *jokaisen* vastauspaperin yläreunaan erillinen huomautus: "Syksyn 2018 uusinta".



JYVÄSKYLÄN YLIOPISTO

Tehtävä 1 (6 p.)

Valitse jokaisesta kohdasta (1–6) täsmälleen yksi vaihtoehdoista (a–d). Kukin kohta on yhden pisteen arvoinen. Perusteluita ei tarvitse eikä pidä kirjoittaa.

1. Mikä seuraavista väittämistä pitää paikkansa C#:ssa?
 - (a) Muuttujan näkyvyyttä aliohjelman sisällä voi rajoittaa.
 - (b) Samanniminen muuttuja kahdessa eri aliohjelmassa aiheuttaa käännösvirheen.
 - (c) Muuttujan arvon muuttaminen sen näkyvyysalueen sisällä aiheuttaa käännösvirheen.
 - (d) Mikäli `for`-silmukan määrittelyrivillä on alustettu `i`-niminen muuttuja, tulee seuraavassa `for`-silmukassa käyttää eri nimistä muuttujaa.

2. Mikä seuraavista väittämistä pitää paikkansa C#:ssa? Ohjelmassa voi olla...
 - (a) vain yksi saman niminen aliohjelma.
 - (b) useita samannimisiä aliohjelmia, joiden parametrien nimet eroavat toisistaan.
 - (c) useita samannimisiä aliohjelmia, joilla on sama määrä parametreja, mutta niiden tyypit eroavat toisistaan.
 - (d) useita samannimisiä aliohjelmia, joiden parametrit ovat täysin samat, mutta joiden paluarvojen tyypit eroavat toisistaan.

3. Mikä seuraavista väittämistä pitää paikkansa C#:ssa?
 - (a) On mahdollista, että aliohjelmakutsun argumentteina on pelkkiä aliohjelmakutsuja.
 - (b) Jos aliohjelmassa kutsutaan toista aliohjelmaa, tulee kutsussa antaa vähintään yksi argumentti.
 - (c) Jos aliohjelman paluarvon tyyppi on jotain muuta kuin `void`, niin sillä tulee olla vähintään yksi parametri.
 - (d) Aliohjelman parametrimuuttujan nimi täytyy olla sama kuin aliohjelmakutsun argumenttina olevan muuttujan nimi.

4. Mitä seuraava koodi tulostaa?

```
StringBuilder sb1 = new StringBuilder("Koira");
StringBuilder sb2;
sb2 = sb1;
sb1 = sb1.Append(" istuu");
Console.WriteLine(sb1 + " " + sb2);
```

 - (a) "Koira istuu Koira"
 - (b) "istuu istuu"
 - (c) "Koira istuu"
 - (d) "Koira istuu Koira istuu"

5. Mitä alla olevista vaihtoehdoista seuraa siitä, että aliohjelma on määritelty void-tyyppiseksi?

- (a) Aliohjelmaa *ei* voi kuormittaa.
 - (b) Aliohjelmalla *ei* ole parametreja.
 - (c) Aliohjelmalla *ei* voi tehdä rekursiota.
 - (d) Aliohjelmakutsu *ei* voi olla toisen aliohjelmakutsun argumenttina.
6. Mikäli funktion paluuarvon tyyppi on määritelty `double`, mikä seuraavista pitää paikkansa?
- (a) Paluuarvo voi sisältää listan `double`-lukuja.
 - (b) Paluuarvo voidaan sijoittaa `int`-tyyppiseen muuttujaan.
 - (c) Mikäli funktiossa on lause `return "2"`; ohjelma ei käänny.
 - (d) Funktiolle annettavan argumentin tai argumenttien tulee olla `double`-lukuja.

Tehtävä 2 (6 p.)

1. Anna lyhyet esimerkit kolmesta erilaisesta C#:n toistorakenteesta. (1 p.)
2. Laske yhteen 8-bittiset binääriluvut 1011 1011 ja 0100 1101. Ilmoita vastaus 8-bittisenä binäärilukuna ja kymmenjärjestelmän lukuna. (1 p.)
3. Oheisen funktion pitäisi palauttaa parametrina saatujen lukujen keskiarvo. Funktiossa on kuitenkin joitain vikoja. Mainitse ainakin neljä eri asiaa, jotka vaativat korjausta, ja kerro miten ne tulee korjata. (2 p.)

```
public static void keskiarvo(List luvut)
{
    int summa = 0;
    for (int i = 0; i <= luvut.Length; i++)
    {
        summa += luvut(i);
    }
    return summa / luvut.Length;
}
```

4. Tutki koodia Liitteessä A. Mitä pääohjelma tulostaa? (2 p.)
- (a) 6
 - (b) 14
 - (c) 24
 - (d) 36

Tehtävä 3 (6 p.)

Valitse jokaisesta kohdasta (1–6) täsmälleen yksi vaihtoehdoista (a–d). Kukin kohta on yhden pisteen arvoinen. Perusteluita ei tarvitse eikä pidä kirjoittaa.

1. Tutki koodia Liitteessä B. Mikä väittämistä pitää paikkansa?
 - (a) Jos toinen jonoista on tyhjä merkkijono (`string`-olio, jossa ei ole yhtään merkkiä), ohjelma kaatuu.
 - (b) Jos toinen jonoista on tyhjä merkkijono, tulosjonon pituus on 0.
 - (c) Jos jonot ovat eri pituisia, tulee tulosjonon pituudeksi sama kuin lyhyemmän jonon pituus.
 - (d) Jos jonot ovat eri pituisia, tulee tulosjonon pituudeksi `jono1` ja `jono2` yhteenlaskettu pituus.
2. Tutki koodia Liitteessä B. Minkä merkkijonon funktio palauttaa kutsuttaessa `UusiJono("OHJ", "1");`?
 - (a) 1
 - (b) 01
 - (c) OHJ1
 - (d) 01H1J1
3. Mikä seuraavista väittämistä pitää paikkansa C#:ssa?
 - (a) Aliohjelmassa voi olla enintään yksi `return`-lause.
 - (b) Lause `if ('c' == 89)` on syntaktisesti oikein.
 - (c) Taulukon koko voi muuttua sen esittelyn jälkeen.
 - (d) Luvun `int i = 4;` toinen potenssi saadaan kirjoittamalla `i^2`.
4. Mitä tulostaa `Console.WriteLine((true || false) && !true || 2 < 3);`?
 - (a) False
 - (b) True
 - (c) `(true || false) && !true || 2 < 3`
 - (d) Ei mitään, ohjelma ei käänny.
5. Mikä seuraavista väittämistä pitää paikkansa C#:ssa?
 - (a) Jos meillä on funktiokutsu `F(5.2)`, tulee funktion `F` paluarvon olla tyyppiä `double`.
 - (b) Funktion paluarvo täytyy aina sijoittaa johonkin muuttujaan.
 - (c) Funktion paluarvo voi olla eri tyyppiä kuin sen parametrit.
 - (d) Funktion paluarvon tyypin määrittämisen voi halutessaan jättää kirjoittamatta.
6. Mitä tulostaa `Console.WriteLine("1" + 5 + 6);`?
 - (a) 156
 - (b) 111
 - (c) 12
 - (d) Ei mitään, ohjelma kaatuu.

Valitse toinen seuraavista tehtävistä

Valitse ja tee (vain) toinen alla olevista tehtävistä a tai b. Jos kuitenkin teet molemmat tehtävät, tenttitulokseesi huomioidaan se tehtävä josta sait vähemmän pisteitä. Dokumentaatiokommentit on näissä tehtävissä kirjoitettava XML-tageilla kurssin käytännön mukaisesti.

Tehtävä 4, vaihtoehto a (6 p.)

Collatzin konjektuuri on Lothar Collatzin vuonna 1937 esittämä ratkaisematon matemaattinen ongelma joka on määritelty seuraavasti:

$$f(n) = \begin{cases} n/2, & \text{jos } n \text{ on parillinen} \\ 3n + 1, & \text{jos } n \text{ on pariton.} \end{cases}$$

Eli jos funktiolle annetaan mikä tahansa parillinen luku, se jaetaan kahdella. Mikäli luku on pariton, kerrotaan se kolmella ja lisätään saatuun lukuun yksi. Tuloksena saadun luvun perusteella lasketaan lukujonon seuraava luku. Tätä toistetaan. Esimerkiksi jos lähdetään liikkeelle luvusta 3, päädytään ketjuun $3 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1 \rightarrow 4 \rightarrow 2 \rightarrow 1 \dots$. Toistaiseksi ei ole löydetty ainuttakaan lukua jonka ketju ei lopulta päättyisi edellä olevaan lukujen 4, 2 ja 1 muodostamaan ikuiseen silmukkaan.

Toteuta funktio Collatz, joka ottaa parametrina yhden luvun. Tämä luku toimii lukujonon ensimmäisenä lukuna. Funktio palauttaa listan, joka sisältää tämän ensimmäisen luvun, ja kaikki lukujonon seuraavat luvut. Funktion suoritus lopetetaan, kun luku 1 tulee ensimmäisen kerran vastaan. Funktiota voi kutsua esimerkiksi seuraavasti:

```
List<int> luvut = Collatz(3);
Console.WriteLine(String.Join(" ", luvut));
```

Tästä saadaan tulostus:

```
3 10 5 16 8 4 2 1
```

Tehtävä 4, vaihtoehto b (6 p. + 2 p.)

Tee klassista Hirsipuu-peliä varten funktio, jonka avulla voidaan näyttää alkuperäisestä lauseesta arvatut merkit. Funktion nimi on `NaytaArvatut`, jolla on kaksi parametria:

- **alkuperäinen lause**, joka sisältää oikean vastauksen, sekä
- **arvatut kirjaimet**, joka on joukko käyttäjän jo arvaamia merkkejä.

Esimerkiksi jos funktiolle annetaan argumentteina

```
Alkuperäinen lause: "Tulevaisuus on nyt! Oletko valmis?"
```

```
Arvatut kirjaimet: a, l, o, j, t, u
```

niin funktio palauttaa merkkijonon `"Tul**a**uu* o* **t! Ol*t*o *al***?"`.

Päätä itse parametrien tyypit ja nimet. Luokkaa tai using-lauseita ei tarvitse kirjoittaa. Voit olettaa, että arvattujen kirjainten joukko sisältää pelkästään pienaakkosia. Saat **2 bonuspistettä**, mikäli erikoismerkit (kuten huutomerkki ja kysymysmerkit) sekä välilyönnit ovat aina mukana tulosjonossa yllä olevan esimerkin mukaisesti, vaikka ne eivät olisikaan arvattujen kirjainten joukossa. Kuitenkin, 6 pistettä on mahdollista saada vaikka tätä käsittelyä erikoismerkeille ei olisikaan toteutettu.

Vinkki 1: `String.IndexOf`. Vinkki 2: `Char.ToLower`.

Liite A

Luokka, using-lauseet ja dokumentaatiot on jätetty pois tilan säästämiseksi.

```
public static void Main()
{
    int summa = HassuSumma(6, 12, 2);
    Console.WriteLine(summa);
}

public static int HassuSumma(int alku, int loppu, int askel)
{
    int summa = 0;
    if (askel <= 0) return 0;
    for (int i = alku; i < loppu; i += askel)
    {
        summa += i;
    }
    return summa;
}
```

Liite B

Luokka, using-lauseet, pääohjelma ja dokumentaatiot on jätetty pois tilan säästämiseksi.

```
public static string UusiJono(string jono1, string jono2)
{
    int pituusMin = Math.Min(jono1.Length, jono2.Length);
    StringBuilder sb = new StringBuilder();
    for (int i = 0; i < pituusMin; i++)
    {
        sb.Append(jono1[i]);
        sb.Append(jono2[jono2.Length - 1 - i]);
    }
    return sb.ToString();
}
```