

# ITKP102 Ohjelmointi 1 (6 op)

Tentaattori: Antti-Jussi Lakanen

20. huhtikuuta 2018

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

Jos tämä tentti on sinulle syksyn 2017 kurssi-instanssin uusintatentti, niin laita *jokaisen* vastauspaperin yläreunaan erillinen huomautus: “Syksyn 2017 uusinta”.

## **In English**

If you want the questions in English, please ask the invigilator.



JYVÄSKYLÄN YLIOPISTO

## Tehtävä 1 (6 p.)

Järjestä seuraavat C#-ohjelman osat oikeaan järjestykseen niin, että tuloksesta muodostuu toimiva ohjelma. Käytä kutakin osaa täsmälleen kerran. Anna jokaisesta osasta todellinen koodiesimerkki. Kirjoita vastauspaperiin riveittäin luku (1–11) ja koodiesimerkki. Esimerkkien tulee muodostaa toimiva, kääntyvä kokonaisuus. Myös sisennykset, nimeäminen ja kirjoitusasu otetaan arvostelussa huomioon. Dokumentaatiokommentteja ei tarvitse kirjoittaa.

1. Main-pääohjelman esittelyrivi.
2. Järjestelmäkirjaston using-lause.
3. Muuttujan `x` määrittely. Muuttujaan tulee voida tallentaa kokonaisluokuarvo.
4. `x`-muuttujan tulostuslause.
5. Luokan esittelyrivi.
6. Satunnaisen, välillä 1–6 olevan luvun sijoittaminen muuttujaan `x`.
7. Random-olion määrittely ja alustus.
8. Luokan aloittava aaltosulku.
9. Luokan lopettava aaltosulku.
10. Pääohjelman aloittava aaltosulku.
11. Pääohjelman lopettava aaltosulku.

## Tehtävä 2 (6 p.)

(a) Alla on rekursiivinen aliohjelma `Neliot`, jota kutsutaan `Begin`-aliohjelmassa. Minkä oheisista kuvioista (a–d) ohjelma piirtää? Oletetaan, että käytössä on Jypeli-kirjasto, ja että luokka, `Main`-pääohjelma jne. on hyvin määritelty ja ohjelma kääntyy. (3 p.)

```

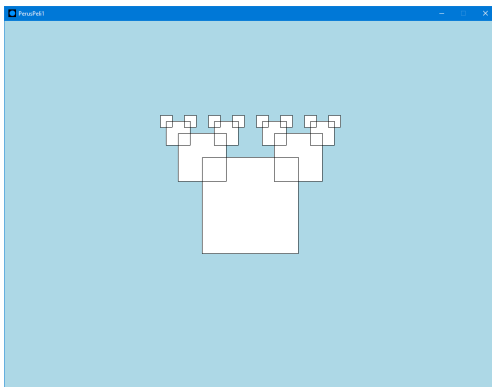
public override void Begin()
{
    Neliot(0, 0, 200);
}

public void Neliot(double x, double y, double w)
{
    if (w < 20) return;

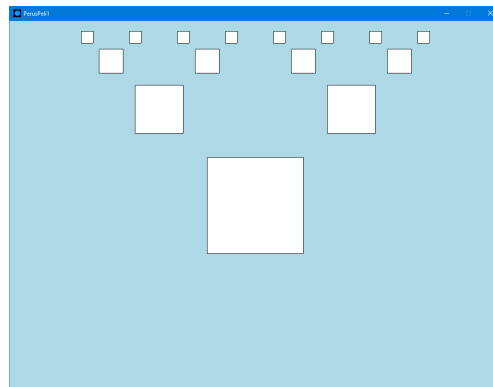
    GameObject nelio = new GameObject(w, w);
    nelio.Position = new Vector(x, y);
    Add(nelio);

    Neliot(x - w , y + w, w / 2);
    Neliot(x + w , y - w, w / 2);
}

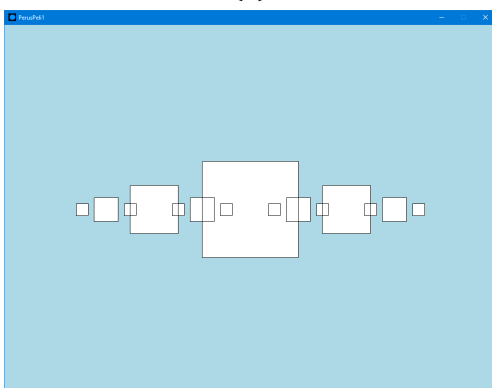
```



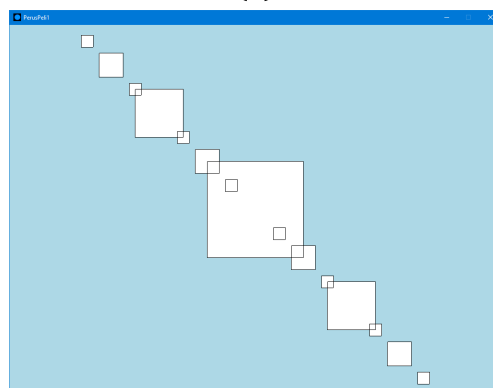
(a)



(b)



(c)



(d)

(b) Tee funktio `Kertoma`, joka laskee luvun  $n$  kertoman (rekursiivisesti tai iteratiivisesti, voit itse valita). Oletetaan, että  $n > 0$ . Kertoman määritelmä:

```
Kertoma(0) = 1
Kertoma(n) = n * Kertoma(n-1)
```

Dokumentaatiokommentit on kirjoitettava XML-tageilla kurssin käytännön mukaisesti. (3 p.)

### Tehtävä 3 (6 p.)

Valitse jokaisesta kohdasta (1–6) täsmälleen yksi vaihtoehtoista (a–d). Kukin kohta on yhden pisteen arvoinen. Perusteluita ei tarvitse eikä pidä kirjoittaa. Kirjoita vastauksesi vastauspaperille, ei tähän paperiin.

1. Olkoon meillä oheinen funktion esittelyrivi. Mikä seuraavista väitteistä pitää paikkansa esittelyrivin perusteella?

```
public static bool Anagrammi(string merkkijono)
```

- (a) Funktion kutsuminen tyhjällä merkkijonolla (string-olio, jossa ei ole yhtään merkkiä) aiheuttaa ajonaikaisen virheen.
- (b) Mikäli merkkijono-muuttujan sisältö on saippuakauppias, funktio palauttaa merkkijonon saippuakauppias.
- (c) Funktiota voi kutsua ilman, että paluuarvoa sijoitetaan bool-muuttujaan kutsuvassa aliohjelmassa.
- (d) Funktio voi palauttaa arvon 0 tai 1.

2. Mikä seuraavista väittämistä pitää paikkansa C#:ssa?

- (a) Operaattoria = käytetään `if`-lauseen ehto-osassa kahden totuusarvon vertailuun.
- (b) Operaattoreilla `<` ja `>` voidaan muuttaa muuttujan arvoa.
- (c) Muuttujan voi määritellä ennen arvon sijoittamista kyseiselle muuttujalle.
- (d) Sijoitusoperaattori vaatii aina aalto- tai kaarisulkeet ympärilleen.

3. Mikä seuraavista väittämistä pitää paikkansa C#:ssa?

- (a) Taulukon pituutta voi muuttaa taulukon luomisen jälkeen.

- (b) Taulukon elementit sijaitsevat indekseissä 0...Length.
- (c) Taulukko voi olla enintään 2-ulotteinen.
- (d) Taulukon a pituus (a.Length) voidaan päätellä seuraavan lauseen perusteella:

```
int[] a = new int[] { };
```

4. for-silmukan yleinen muoto on  

```
for (alustus ; toistoehto ; päivityslauseet) { runko }
```

Mikä seuraavista väittämistä pitää paikkansa C#:ssa?

- (a) Toistoehdoksi käy mikä tahansa kokonaislukuarvoinen lauseke.
- (b) for-silmukalla ei ole mahdollista tehdä ikuista silmukkaa.
- (c) Mikä tahansa for-silmukan osista (alustus, toistoehto, päivityslauseet, runko) voidaan jättää tyhjäksi.
- (d) Alustus-kohdassa on määriteltävä ainakin yksi int-tyyppinen muuttuja.

5. Olkoon meillä seuraava koodi:

```
string jono = "1";
int maara = MerkkienMaara(jono + 23);
```

Oletetaan lisäksi, että on olemassa MerkkienMaara-funktio ja sille esitelyrivi:

```
public static int MerkkienMaara(string s)
```

Mikä seuraavista väittämistä pitää paikkansa?

- (a) Ohjelma ei käänny.
- (b) MerkkienMaara-funktiolle annetaan kutsussa yksi parametri, "123".
- (c) MerkkienMaara-funktiolle annetaan kutsussa yksi parametri, "24".
- (d) MerkkienMaara-funktiolle annetaan kutsussa kaksi eri parametria, "1" ja "23".

6. Mikä seuraavista väittämistä pitää paikkansa C#:ssa?

- (a) Ei-void-funktiossa (ts. funktion paluuarvon tyyppi on jotain muuta kuin void) täytyy olla vähintään yksi return-lause.
- (b) Olkoon sijoituslause `int d = F(1.1);`. Funktion F paluuarvon tyyppi voi olla double.

- (c) `void`-tyyppisessä funktiossa `return`-lause aiheuttaa käänkösvirheen.
- (d) `return`-lauseen voi korvata tarvittaessa `break`-lauseella.

## Tehtävä 4 (6 p.)

Tee funktio, joka laskee pankissa olevan talletuksen arvon  $N$  vuoden jälkeen käyttäen kaavaa

$$P * (1 + R)^N$$

missä  $P$  on alkupääoma,  $R$  on vuosittainen korko ja  $N$  on talletuksen pituus vuosina. Oletetaan, että talletuksen kesto on kokonaisia vuosia. Potenssiin korotusta ei saa tehdä valmiilla funktiolla, kuten `Math.Pow`:lla.

Esimerkiksi alkupääomalla 100, korolla 0.03 (siis 3%) ja viiden vuoden talletusajalla talletuksen arvo olisi noin 115.93. Tällöin funktiota voisi kutsua esimerkiksi seuraavasti:

```
double arvo = Talletus(100.0, 0.03, 5);  
Console.WriteLine(arvo) // 115.9274...
```

Dokumentaatiokomentit on kirjoitettava XML-tageilla kurssin käytännön mukaisesti.