Sovelto

Parametriharjoituksia

Parametrien välittäminen
vaihtelevan mittainen parametrilista
(in), ref ja out-parametrit

ACADEMY
A PART OF ACADEMIC WORK

Harjoitus 1

- Aloita uusi projekti, nimi olkoot ParametriHarjoitukset
- Tee luokka Testipenkki
- Lisää tähän luokkaan ensimmäinen metodi: NimiKuntoon, parametrit *string etunimi* ja *string sukunimi*. Tämä metodi palauttaa (return) nimen muodossa **Sukunimi, Etunimi** ja lisäksi tarkistaa ja tarvittaessa korjaa niin että etu- ja sukunimi alkaa isolla kirjaimella ja on muuten pienillä kirjoitettu
- Tee vielä toinen metodi NimiKuntoon jolle menee parametrina string nimi. Pystyt päättelemään edellisestä miten metodin pitäisi toimia. Tässä siis harjoitellaan overloadingia eli kuormittamista
- Tee Program-luokkaa uusi metodi static void Testi1() {}, tässä metodissa luo uusi instassi Testipenkki-luokasta ja testaa että NimiKuntoon metodi toimii oikein
- Huom.: viikolla 2 opetellaan oikeasti testausta Unit Testing (yksikkötestaus) -menetelmillä
- Lisätehtävä: lisää kolmas parametri (char) erotin joka tulee etunimen ja sukunimen väliin
  ➔ joudut korjaamaan kutsut ellet sitten lisää erotin-parametrille oletusarvoa esim.: ','

Harjoitus 1 – ohjeita ja vinkkejä

1. Käynnistä Visual Studio sitten File | New project ➔ Console Application tai
   dotnet new console –n Parametriharjoitukset –o c:\work\viikko1\ti\kertaus
2. Lisää projektiin uusi luokka valikosta Project | Add Class ja nimi on Testipenkki
3. Tähän luokkaa tee metodi
   ```
   public string NimiKuntoon(string etunimi, string sukunimi) { return "toimii?";}
   ```
4. ```
   static void Testi1() {
        Testipenkki tp = new Testipenkki();
        string nimi = tp.NimiKuntoon("aku", "ankka");
        Console.WriteLine(nimi);
   }
   ```
5. Testaamisen helpottamiseksi voit myös kysyä nimen osat käyttäjältä
6. Korjaa NimiKuntoon-metodi kuntoon vastaamaan edellisen sivun mukaista määritelmää tarvinnet: Substring, ToUpper, ToLower -metodeja
   ```

Harjoitus 1 – lisää ohjeita ja vinkkejä

- merkkijonon ensimmäinen merkki isoksi ja muut pieneksi?

1. saat ensimmäisen merkin:
   - char ekamerkki = etunimi[0]; // tai
   - string ekamerkki = etunimi.Substring(0, 1); // mitä näillä eroa? Muutakin kuin paluutyyppi?
2. seuraavat merkit/merkkijonon loppuosan saat edelleen Substring-metodilla kun valitset sen parametrit oikein
3. muuta ensimmäinen merkki isoksi ToUpper() –metodilla ja muut varmuuden vuoksi pieniksi ToLower() –metodilla

- Jotta ei olisi liian helppoa niin muista testata niin että etunimi parametrissa on "jeremias" ja sukunimessä "yli-rotiala", toimiiko? Jos ei niin miten korjaisit tämän?

Harjoitus 2

- Lisää projektiin oheisen esimerkin mukainen Tuote-luokka
- Lisää Testipenkki-luokkaan metodi:
  `public bool TarkistaTuote(Tuote t, out string virheet, out string varoitukset)`
- Metodi pitää tarkistaa Tuote-olion eheys seuraavien sääntöjen mukaan:
  - Tuotenumero oltava välillä [1000 – 9999[, muutoin varoitus
  - Nimi, minimipituus 5 merkkiä ja maksimi 30, virhe jos pituus muuta
  - Kuvaus puuttuu, varoitus
  - Hinta oltava välillä [0.01 – 999.99], virheilmoitus
- Metodi palauttaa true-arvon jos kaikki ehdot on kunnossa.
- Lisää Program-luokkaan testit
- Paluuarvo on false, jos on jotain ilmoituksia palautettavana. Virheilmoitukset ja varoitukset palautetaan *virheet* ja *varoitukset* –parametreissa.

```
class Tuote
{
    public int Tuotenumero { get; set; }
    public string  Nimi { get; set; }
    public string  Kuvaus { get; set; }
    public decimal Hinta { get; set; }
    // voit tehdä sopivan konstruktorin
}
```

Harjoitus 2 – vinkkejä, luokka

1. Tee Tuote-luokka (Add Class-toiminto)
2. Toteuta Tuote-luokka edellisen sivun mallin mukaisesti
3. Palaa takaisin Testipenkki-luokkaan ja lisää TarkistaTuote-metodi, edellisellä sivulla malli
4. Lisää ensin ensimmäinen (vaikka tuotenumero) tarkistus

```
…TarkistaTuote( … ) {
bool ok = true;
virheet = ""; varoitukset = "";
if (t.Tuotenumero < 1000
       || t.Tuotenumero > 9999) {
       virheet = "Tuotenumero väärin";
       ok = false;
}
return ok;
}
```

```
static void TuoteTesti() {
    Testipenkki tp = new Testipenkki();
    Tuote t = new Tuote();
    t.Tuotenumero = 77;
    t.Nimi = "testituote";
    // esittele muuttujat virheille ja varoituksille tässä
    var tulos = tp.TarkistaTuote(t, out virhetekstit, out varoitustekstit);
    // tulosta paluuarvo ja out-parametrit konsolille
}
```

5. Lisää Program-luokkaan sopiva testi metodi, luo siinä Testipenkki-olio ja Tuote-olio. Aseta Tuotenumero arvoon 77 ja välitä se Testipenkin TarkistaTuote-metodille. Huomaa että tarvitset kaksi string-tyyppistä muuttujaa out-parametrejä varten.
6. Tulosta paluuarvo ja muut tiedot
7. Toteuta loput tarkistukset, kaikki virheilmoitukset palautetaan samassa merkkijonossa

Harjoitus 3

- Lisää Testipenkkiin uusi metodi LaskeYhteen:

```
public int LaskeYhteen(params int[] luvut)
```

- Olet siis tekemässä metodia jolle voi antaa parametrina n-kpl kokonaislukuja ja metodi laskee ne kaikki yhteen ja palauttaa saadun summan
- Lisää LaskeYhteen –metodiin paikallinen muuttuja esimerkiksi int summa = 0;
- käy koko parametritaulukko luvut läpi foreach-silmukalla ja silmukan sisällä summaa kaikki luvut yhteen
- palauta summa ja testaa toiminta:

```
Testipenkki tp = new Testipenkki();
int tulos = tp.LaskeYhteen(1, 33, 42, 57);
// tulosta konsolille tulos
```

Harjoitus 4

- Lisää Testipenkkiin uusi metodi TeeTaulukko:

```
public string[] TeeTaulukko(params string[] nimet)
```

- Metodissa luo taulukko jonka tyyppi on string[]
- Lisää luomaasi taulukkoon kaikki parametrina tulleet merkkijonot aakkosjärjestyksessä
- Palauta taulukko return-lauseella
- Testaa, esimerkki:

```
Testipenkki tp = new Testipenkki();
string[] st = tp.TeeTaulukko("Aku", "Pelle", "Roope", "Iines");
// tulosta konsolille st-taulukon alkiot pilkulla erotettuna
```

Harjoitus 4 - ohjeita

1. Lisää metodi edellisen sivun esimerkin mukaan
2. Jotta saat kääntymään niin ensin metodin koodi olkoot vain `return null;`
3. järjestä nimet-taulukko, Array.Sort() on yksi vaihtoehto
4. Luo uusi string-taulukko ja sen koko on sama kuin nimet-taulukon (nimet.Length)
5. kopioi merkkijonot nimet taulukosta juuri luomaasi taulukkoon
   - for tai foreach –silmukassa tai
   - Array.Copy() saattaa olla toimiva metodi
6. Testerikoodissa saat siis uuden taulukon jonka nimi on st jos teet esimerkin mukaan
7. tulosta taulukon sisältö konsolille niin että taulukon sisältö tulostuu yhtenä merkkijonona:
   - esittele muuttuja string s = ""; ja lisää siihen silmukassa st-taulukon sisältö, muista laittaa pilkut väliin ja tulosta se Console.WriteLine()-metodilla
   - tai kokeile onnistuuko String.Join() –metodilla helpommin