

## 07 Array

Array (Taulukko)

1. Kirjoita Windows Forms -ohjelma, jonka formissa on toiminnot taulukon luontia, talletusta ja hakua varten (Kuva 1-1).

The screenshot shows a Windows Forms application window titled "Pääformi". It has a standard Windows interface with a title bar containing a minimize, maximize, and close button. Below the title bar is a menu bar with a single item "File". The main content area is divided into three sections by section headers: "TAULUKON LUONTI", "TALLENNUS", and "HAKU". In the "TAULUKON LUONTI" section, there is a label "Taulukon koko:" followed by a text input field containing the number "10" and a button labeled "Luo taulukko". In the "TALLENNUS" section, there is a label "Talletettava numero:" followed by a text input field containing "313", and below it, a label "Talletuspaikan indeksi:" followed by a text input field containing "4" and a button labeled "Talleta". In the "HAKU" section, there is a label "Talletuspaikan indeksi:" followed by a text input field containing "4" and a button labeled "Hae". Below the "Hae" button, there is a text box displaying the number "313".

Kuva 1-1. Pääformi

### Vaihe 1

Kirjoita ensin taulukon luontitoiminto (Kuvan 1-1 TAULUKON LUONTI). Taulukon koko kenttään syötetään taulukon muistipaikkojen lukumäärä. Kun 'Luo taulukko'-nappia klikataan, niin napin tapahtumankäsittelijässä luodaan taulukko.

Testaa ohjelman toimivuus tässä vaiheessa debuggerin avulla ennen kuin jatkat eteenpäin.

- Laita ohjelman ensimmäisen komennon kohdalle *breakpoint* klikkaamalla vasemmalla hiirennapilla vasemmasta palkista rivinumeron vasemmalta puolelta **tai** klikkaamalla riviä oikealla hiirennapilla ja valitsemalla *breakpoint -> insert breakpoint*. Breakpoint on piste johon debuggeri pysäyttää ohjelman ajon tarkastelua varten.
- Kun debugger on aktiivinen lisää halutut muuttujat *watch*-listaan joka löytyy oletusarvoisesti vasemmalta alhaalta. Tabin nimi on yleensä *Watch 1*.
- Siirry ohjelmassasi askel kerrallaan eteenpäin F11 -nappia painamalla. F10 -napilla pääset myös etenemään askel kerrallaan mutta siten, että debuggeri ei mene sisään aliohjelmakutsuihin. Tämä on hyödyllistä esimerkiksi käytettäessä systeemifunktioita, joiden toteutusta harvemmin on tarkoitus debugata.

## Vaihe 2

Lisää formiin tallennustoiminto. Talletettava numero syötetään 'Talletettava numero'-kenttään ja talletuspaikan indeksi kirjoitetaan 'Talletuspaikan indeksi'-kenttään. Huomaa, että taulukon ensimmäisen paikan indeksi on 0 toisen 1 ja kolmannen 2 jne. Kun 'Talleta'-nappia klikataan, niin talletettava numero talletetaan edellä luotuun taulukkoon talletuspaikan indeksin osoittamaan paikkaan.

Testaa ohjelman toimivuus tässä vaiheessa debuggerilla saamaan tapaan, kuin vaiheessa 1.

Huom!

Muuttujat ovat olemassa vain siinä lohkossa missä ne on määritelty. Lohko on aaltosulkujen {} rajaama alue. Jos esittelemäsi taulukko ei ole vielä luotu kun yrität sijoittaa siihen numeroita, aiheutat *Null Reference* virheen. Taulukko, kuten kaikki oliot, voidaan esitellä erikseen ja varsinainen taulukko-olio ja sen koko myöhemmin. Vasta tässä vaiheessa varataan tilaa muistista.

```
Int[] taulukko;          // esittelee kääntäjälle taulukon

taulukko = new int[5]; // varaa muistista tilaa viiden kokonaisluvun

                        // kokoiselle taulukolle
```

Esittele taulukkosii siis ensin ja koko lomakkeen kattavaksi ja määritä koko vasta kun käyttäjä on antanut tarvittan tiedon.

## Vaihe 3

Lisää formiin hakutoiminto. Kun haettavan numeron talletuspaikan indeksi syötetään 'Talletuspaikan indeksi'-kenttään ja klikataan 'Hae'-nappia, niin taulukosta haetaan indeksin mukaisesta talletuspaikasta numero ja tulostetaan se formissa olevaan Label-komponenttiin.

## Vaihe 4

Entä jos käyttäjä syöttää liian suuren indeksi numeron tai kirjaimen? Lisää virheen tarkistus hyödyntämällä [try-catch rakennetta](#).