04 Switch

Kirjoita Windows Forms -ohjema, jonka pääformissa (Kuva 1-1) on 5 nappia. Huomaa, että kaikki napit käyttävät **samaa** tapahtumankäsittelijää (metodia). Älä siis luo erikseen onButtonClick tapahtumakäsittelijää jokaiselle eri napille vaan käytä yhtä ja samaa tapahtumankäsittelijää jonka voit lisätä ensimmäisen jälkeen loppuihin nappeihin valitsemalla käsittelijän halutun napin event-valikon onButtonClick eventin kohdalta.

Kun jotakin nappia klikataan hiirellä, niin nappia vastaava Messagebox esitetään näytöllä (Kuva 1-2). Käytä valintojen tekemiseen switch-lausetta.

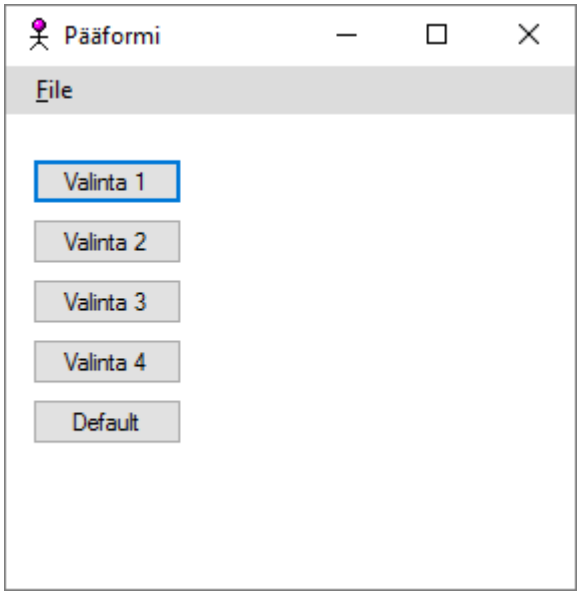
Button click event antaa atribuutteina objektin mikä kutsui sitä ja tapahtuman tiedon. Object on kaikkien lomakkeen controllien(nappi, textbox, label ym.) yläluokka. Ja se täytyy muuttaa haluttuun muotoon, jotta voimme hyödyntää sen ominaisuuksia.

```
private void ButtonClick(object sender, EventArgs e)

{

          Button painettuNappi = sender as Button;

          // tai
          // Button painettuNappi = (Button)sender;

}
```

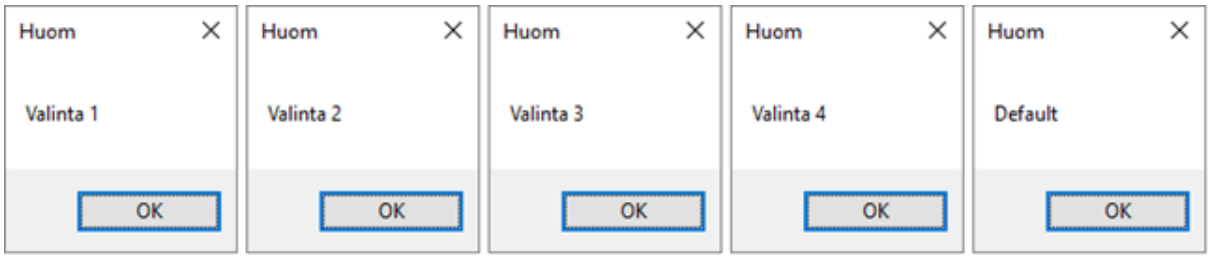
Castaus on eksplisiittinen tyyppimuunnos jonka merkkinä on avainsana *as* tai luokan nimi suluissa olion edessä. Castaus on vaarallinen operaatio ja johtaa helposti virheisiin ja tiedon menettämiseen, jos sitä ei käytä harkiten. Koska aliluokassa (*Button*) on käytössä kaikki yläluokan (*object*) kentät ja metodit **ja** koska tiedämme että lähettäjän luokka on itseasiassa *Button* voimme castata sender olion *Button* tyyppiseksi.
Jos teemme virheen ja castaamme olion luokkaan, jonka jälkeläinen se ei ole, ja jolle sille ei ole määritelty implisiittistä (automaattista) muunnosta, saamme aikaan ajonaikaisen virheen jolta kääntäjä tai IDE (Visual Studio) ei voi meitä suojella.

Muunnoksen jälkeen kääntäjä ja IDE pitävät oliota sen luokan edustajana, johon se on castattu ja voit käyttää kaikkia *Button* luokan ominaisuuksia:

```
painettuNappi.BackColor, painettuNappi.Text jne…
```



Kuva 1-1. Pääformi



Kuva 1-2. Messupoksit