Windows Forms Kata

Dimostrativo

Un *windows forms kata* (*wf kata*) è un esercizio di codifica rivolto alla realizzazione di un'applicazione *windows forms* che implementi una o più funzioni collegate all'interfaccia utente (UI).

Dimostrazione

Di seguito viene mostrato un *wf kata*. L'esercizio viene proposto e risolto nel seguente modo:

- **Problema**: descrizione del problema da risolvere.
- ***Mockup* della UI**: schematizzazione della interfaccia utente.
- **Soluzione dimostrativa**:
    - ***Design time***: disegno dell'interfaccia mediante l'uso di *tool box*, *properties editor* e *form designer*.
    - **Run-time:** codifica.
    - **(Eventuale) refactoring**. (sia a *design-time* che a *run-time*)

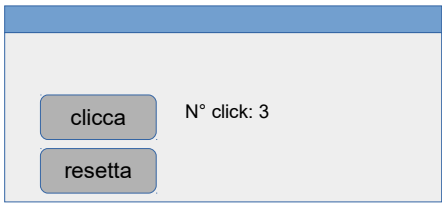
Nota bene: trattandosi di un'applicazione (e non di un metodo) non è previsto una fase di test. Quest'ultimo dovrà essere realizzato provando "manualmente" il corretto funzionamento del programma.

1 Contare i click su un bottone

Problema

Visualizzare il numero di click eseguiti su un *button*. Dare all'utente la possibilità di resettare il conteggio.

Muckup della UI



Soluzione dimostrativa

Design-time: conteggio

Aggiungo al form un *button*, `btnClicca`, e imposto la proprietà `Text` a `"click"`.

Aggiungo una *label*, `lblConta` e Imposto la proprietà `Text` a `"N° click:"`.

Creo un gestore dell'evento `Click` di `btnClicca`:

```
private void btnClicca_Click(object sender, EventArgs e)
{

}
```

Run-Time: conteggio

Dichiaro una variabile globale allo scopo di contare il numero di click eseguiti. La variabile sarà incrementata nel gestore di evento e visualizzata mediante la *label*.

```
int contaClick;
private void btnClicca_Click(object sender, EventArgs e)
{
    contaClick++;
    lblConta.Text = $"N° click:{contaClick}";
}
```

Design-time: reset del contatore

Aggiungo un secondo *button*, `btnResetta` e imposto la proprietà `Text` a `"resetta"`.

Creo un gestore dell'evento `Click` di `btnResetta`:

```
private void btnResetta_Click(object sender, EventArgs e)
{

}
```

Run-Time: reset del contatore

Nel metodo `btnResetta` azzero la variabile `contaClick` e aggiorno la *label*.

```
private void btnResetta_Click(object sender, EventArgs e)
{
    contaClick = 0;
    lblConta.Text = $"N° click:{contaClick}";
}
```

Refactoring

Poiché il codice di visualizzazione del conteggio è duplicato, lo incapsulo in un metodo e modifico i due gestori di evento.

```
void VisualizzaConteggio()
{
    lblConta.Text = $"N° click:{contaClick}";
}

private void btnClicca_Click(object sender, EventArgs e)
{
    contaClick++;
    VisualizzaConteggio();
}

private void btnResetta_Click(object sender, EventArgs e)
{
    contaClick = 0;
    VisualizzaConteggio();
}
```

Conclusioni

Pur trattandosi di un semplice esercizio, ho adottato un approccio incrementale, implementando una funzione per volta: prima il conteggio e soltanto dopo l'operazione di reset. Ho alternato le fasi di disegno e codifica, concentrandomi su una funzione per volta e provando il suo corretto funzionamento.

Alternativamente, si può pensare di disegnare completamente la UI prima di iniziare la codifica. È importante, comunque, eseguire la codifica di una funzione per volta.