

# Grafiske brugergrænseflader i F#

Programmering og problemløsning

Jon Sparring

# Input fra brugeren via Controls

## buttonControl.fsx

```
open System.Windows.Forms
open System.Drawing

let win = new Form () // make a window form
win.ClientSize <- Size (140, 120)

// Create a label
let label = new Label()
win.Controls.Add label
label.Location <- new Point (20, 20)
label.Width <- 120
let mutable clicked = 0
let setLabel clicked =
    label.Text <- sprintf "Clicked %d times" clicked
setLabel clicked

// Create a button
let button = new Button ()
win.Controls.Add button
button.Size <- new Size (100, 40)
button.Location <- new Point (20, 60)
button.Text <- "Click me"
button.Click.Add (fun e -> clicked <- clicked + 1; setLabel clicked)

Application.Run win // Start the event-loop.
```



# Input fra brugeren via Controls

## **buttonControlCompact.fsx**

```
open System.Windows.Forms
open System.Drawing
```

```
// Model: a state 'clicked' that counts how many times an event has occurred
let mutable clicked = 0
let message () = sprintf "Clicked %d times" clicked
let update () = clicked <- clicked + 1

// View: A window containing a label and a button
let win = new Form(ClientSize=Size(140, 120))
let label = new Label(Location=new Point(20, 20), Width=120)
let button = new Button(Size=new Size(100, 40), Location=new Point(20, 60),
Text="Click me")
win.Controls.Add label
win.Controls.Add button

// Connect model and view and start the event-loop
label.Text <- message ()
button.Click.Add (fun e -> update (); label.Text <- message ())
Application.Run win
```

# Organisering af Controls i grupper: Panels

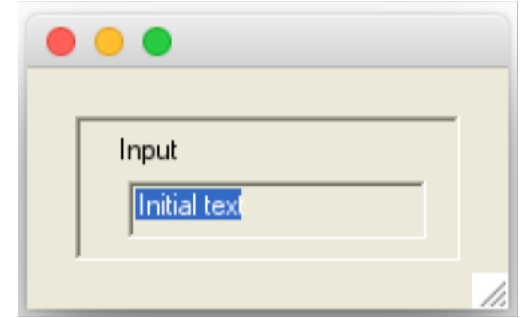
## panel.fsx

```
open System.Drawing
open System.Windows.Forms
```

```
// Create a window with a panel, label and a textbox
let win = new Form(ClientSize=new Size (200, 100))
let panel = new Panel(ClientSize=new Size(160, 60), Location=new Point(20,20), BorderStyle=BorderStyle.Fixed3D)
let label = new Label(ClientSize=new Size(120, 20), Location=new Point(15,5), Text="Input")
let textBox = new TextBox(ClientSize=new Size(120, 20), Location=new Point(20,25), Text="Initial text")

win.Controls.Add panel // Add panel to window
panel.Controls.Add label // add label to panel
panel.Controls.Add textBox // add textbox to panel

Application.Run win // Start the event-loop
```



# Automatisk tildeling af position i paneler

## flowLayoutPanel.fsx

```
open System.Windows.Forms
```

```
open System.Drawing
```

```
// Create a window, a FlowLayoutPanel, 4 buttons, a checkbox, a panel, and 4 radiobuttons
```

```
let win = new Form(ClientSize=new Size(302, 356), Text="A Flowlayout Example")
```

```
let flowLayoutPanel = new FlowLayoutPanel(Location=new Point(47, 55), BorderStyle=BorderStyle.Fixed3D, WrapContents=true)
```

```
let buttonLst =
```

```
    [new Button(Text="Button0");
```

```
     new Button(Text="Button1");
```

```
     new Button(Text="Button2");
```

```
     new Button(Text="Button3")]
```

```
let panel = new Panel(Location=new Point (47, 190),BorderStyle=BorderStyle.Fixed3D)
```

```
let wrapContentsCheckBox = new CheckBox(Location=new Point (3, 3), Text="Wrap Contents")
```

```
let radioButtonLst =
```

```
    [(new RadioButton(Location=new Point(3, 34), Text="TopDown"), FlowDirection.TopDown);
```

```
     (new RadioButton(Location=new Point(3, 62), Text="BottomUp"), FlowDirection.BottomUp);
```

```
     (new RadioButton(Location=new Point(111, 34), Text="LeftToRight"), FlowDirection.LeftToRight);
```

```
     (new RadioButton(Location=new Point(111, 62), Text="RightToLeft"), FlowDirection.RightToLeft)]
```

# Automatisk tildeling af position i paneler

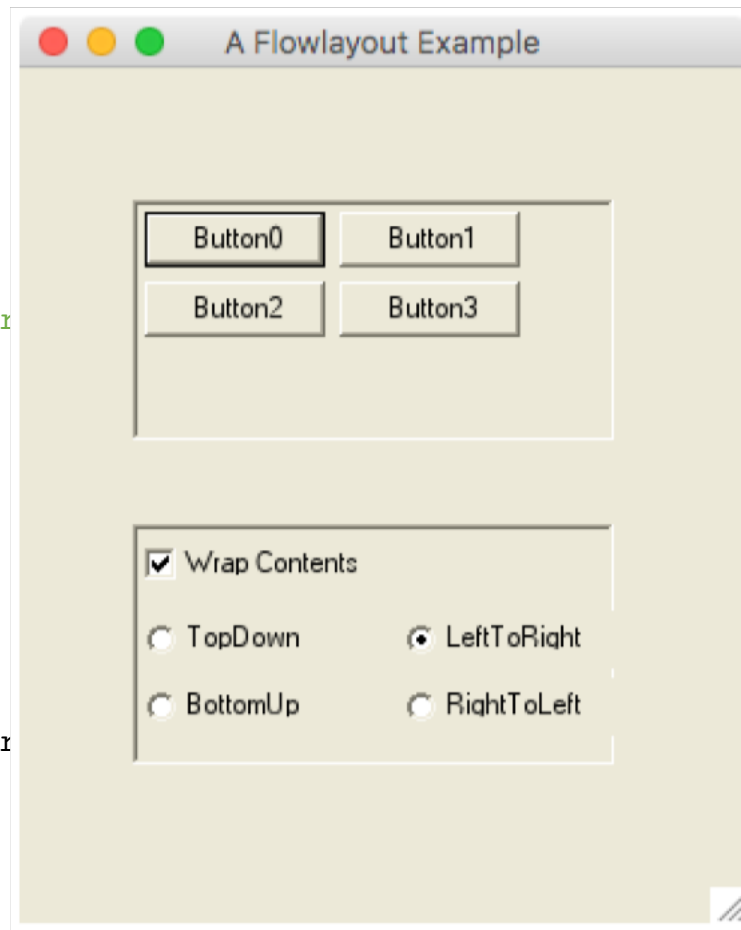
## flowLayoutPanel.fsx

```
// The window contains the panels which in turn contains the buttons, checkbox and radio buttons
win.Controls.Add flowLayoutPanel
for btn in buttonLst do
    flowLayoutPanel.Controls.Add btn
win.Controls.Add panel
panel.Controls.Add (wrapContentsCheckBox)
for btn, dir in radioButtonLst do
    panel.Controls.Add (btn)

// Link wrapContentsCheckBox and flowLayoutPanel.WrapContents
wrapContentsCheckBox.Checked <- flowLayoutPanel.WrapContents
wrapContentsCheckBox.CheckedChanged.Add (fun _ -> flowLayoutPanel.WrapContents <- wrapContentsCheckBox.Checked)

// Link radio buttons and flowLayoutPanel.FlowDirection
for (btn, dir) in radioButtonLst do
    btn.Checked <- flowLayoutPanel.FlowDirection = dir
    btn.CheckedChanged.Add (fun _ -> flowLayoutPanel.FlowDirection <- dir)

// Create a window, add controls, and start event-loop
Application.Run win
```



# Evaluering

## Viden

- Grundlæggende begreber indenfor imperativ, objektorienteret og funktionsprogrammeringsparadigmerne: Funktioner og metoder, variabler, udtryk, typer, kontrolstrukturer, løkker, blokstruktur, klasser og objekter, objektinteraktion, nedarvning, rekursion, polymorfi, abstraktion, undtagelser, pattern matching over rekursive datatyper, m.m.
- God programmeringsskik: Dokumentation i koden, design patterns, afprøvning inkl. unit testing, håndtering af køretidsfejl, m.m.
- Teknikker til problemløsning: Teknisk analyse af naturligsprogsproblemer, objektorienteret design, modelleringssprog, håndkøring, m.m.
- God rapportskrivningsteknik.

## Færdigheder

- At kunne lave mindre programmer (op til ca. 1000 linjer) i de programmeringsparadigmer, der undervises i på kurset, med overholdelse af god programmeringsskik og -stil.
- At kunne evaluere fordele og ulemper ved at opskrive løsningen i de underviste programmeringsparadigmer, og at kunne implementere, afprøve, dokumentere, og evaluere løsningens kvalitet.
- Et sideordnet mål er, at den studerende opnår passende studieteknik således, at dette og parallelkurser består svarende til et fuldtidsstudium.

## Kompetencer

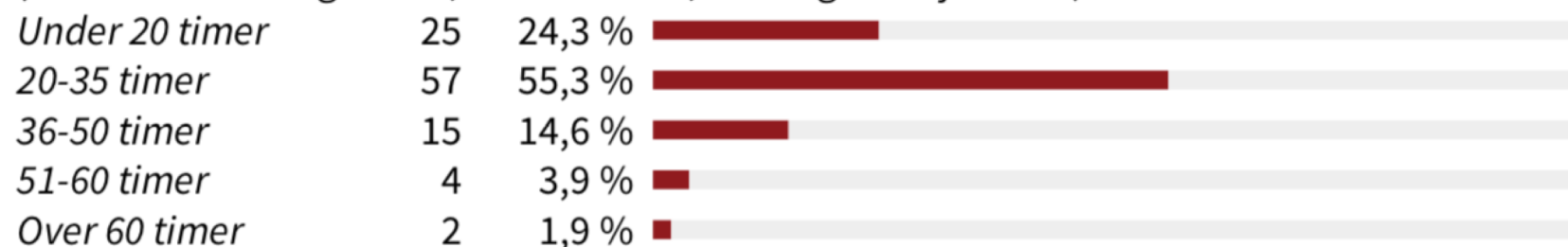
- Ud fra en præcist defineret problemformulering at kunne analysere problemet, udforme et program til løsning af dette, samt at verificere, afprøve, og dokumentere løsningen.

# Evaluating af Learning to program with F#

**<https://tinyurl.com/y7esr4b6>**

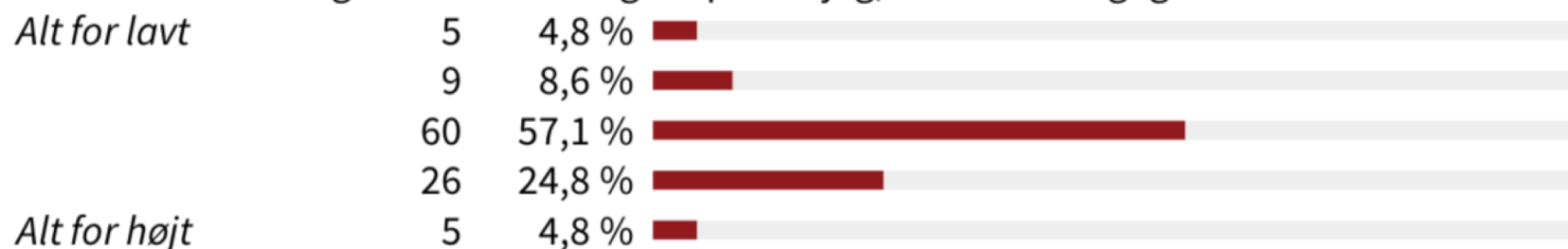


1.1 Min gennemsnitlige ugentlige arbejdsindsats på dette kursus har været på (inkl. undervisningstimer, forberedelse, skriftligt arbejde osv.):

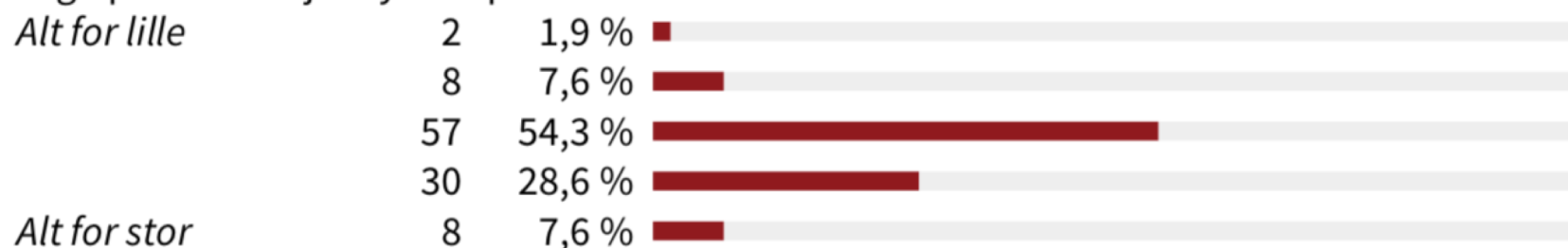


2

2.1 I forhold til mine egne forudsætninger oplever jeg, at kursets faglige niveau er:

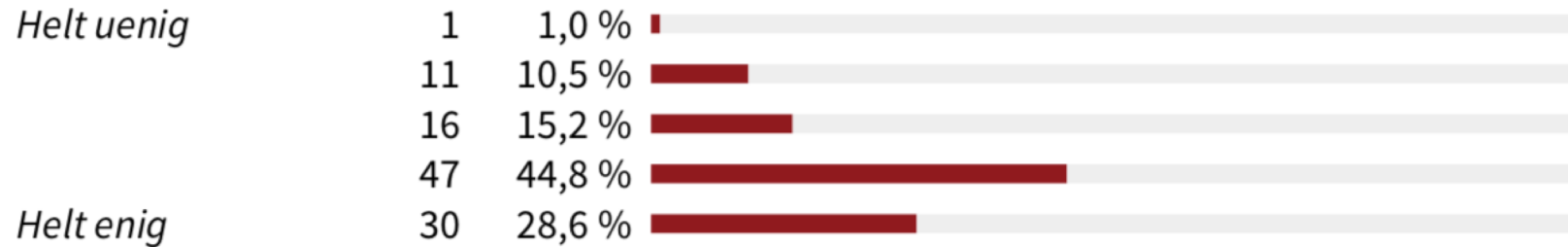


2.2 Jeg oplever arbejdsbyrden på kurset som:

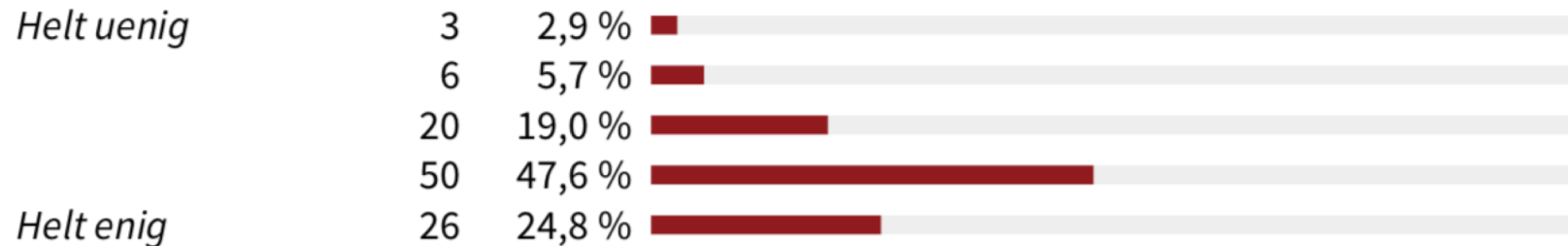


### 2.3 Jeg mener at have opnået kompetencerne beskrevet i kursusbemærkningerne [Se kursets

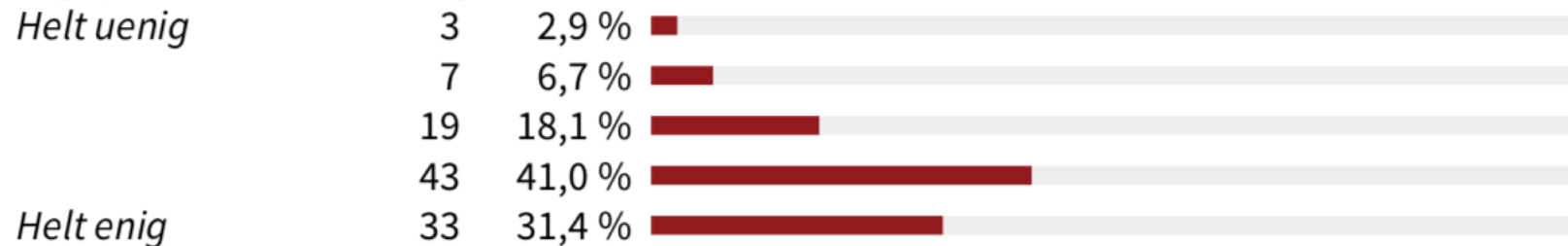
læringsmål fra kursusbeskrivelsen]



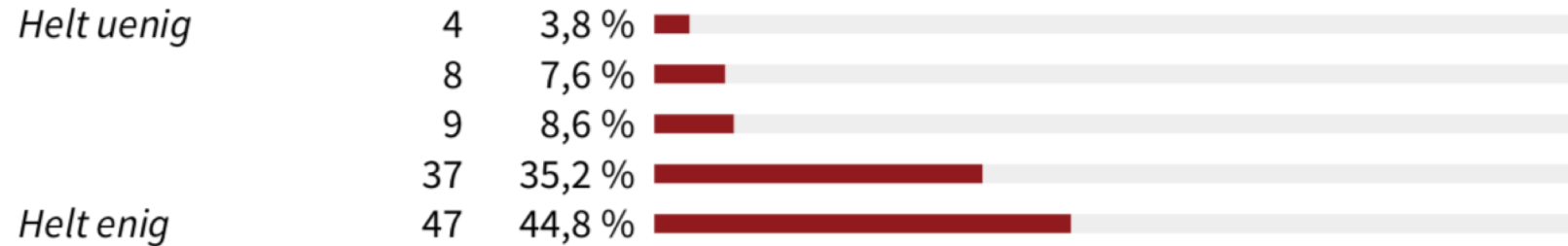
### 2.4 Jeg oplevede, at der var en god sammenhæng mellem de forskellige delelementer (forelæsninger, øvelser m.v.), der indgik i kurset (Uddyb gerne på næste side)



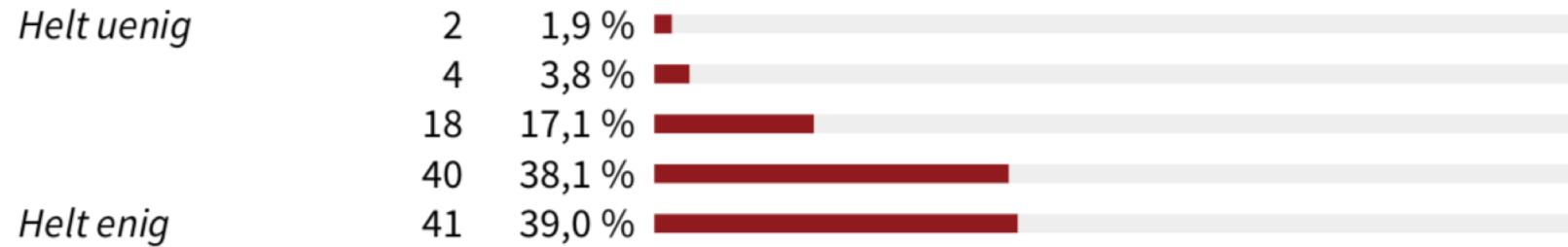
### 2.5 Jeg synes, at undervisningsmaterialet var relevant i forhold til kurset



2.6 Jeg synes, at jeg har fået relevant faglig respons på mit skriftlige og mundtlige arbejde på kurset



2.7 Jeg synes, at jeg har haft adgang til de nødvendige informationer omkring kurset



2.8 Jeg synes samlet set, at kurset har været udbytterigt

