Informatica: C# WPO 2

1. Inhoud

If, globale variabelen, debuggen, randomgetallen, strings vergelijken

2. Oefeningen

- Demo 1: Deelbaar door 0
- Demo 2: Kassa
- Demo 3: Debug oplossingen demo's 1 en 2
- A: Verschillend van?
- A: I'm checked
- A: Binnen bereik!
- A: Nulpunten 2^{de}-graadsvergelijking
- E: SringCompare
- E: Min-Max
- X: Gokmachine

2.1 Demo 1: Deelbaar door 0

Een gebruiker wil snel 2 getallen door elkaar delen. Schrijf een programma dat 2 getallen inleest. Het 2^{de} getal (de deler) wordt gebruikt om het 1^{ste} getal mee te delen (deeltal). Zorg ervoor dat een deling door 0 niet mogelijk is. Geef het resultaat weer in een label.

Codefragment 1: Resultaat weergeven in een label

```
1 int valx = 5;
2 int valy = 10;
3 int result = valy/valx;
4 lblResult.Content = result.ToString();
```

2.2 Demo 2: Kassa

Een kassierster wenst het totaalbedrag van de verkochte producten te kennen. Hierbij voert ze de prijzen van de verschillende producten in. Het invoeren gebeurt d.m.v. een textbox en een knop. Het totaalbedrag wordt in een globale variabele bijgehouden. Vanaf een grensbedrag is er ook een factuur nodig. Wanneer de kassierster op de afrekenknop klikt, verschijnt het bedrag:

- "Bedrag =" gevolgd door het bedrag (totaalbedrag<grensbedrag)
- "Factuurbedrag = " gevolgd door het bedrag (totaalbedrag>=grensbedrag)

Codefragment 2: Gebruik van globale variabelen.

```
// these variables come at the top of the file, just below the class definition
    of our file
public partial class MainWindow : Window
{
   // declare global variable here
   private int value = 0;
   // click event of button Calc
   private void btnCalc_Click(object sender, RoutedEventArgs e)
   {
      // value does not need to be declared since it is globally declared
      // redeclaring here will overwrite the variable only locally
      MessageBox.Show("Globale variabele value = " + value.ToString());
   }
   // click event of button SetValue
   private void btnSetValue_Click(object sender, RoutedEventArgs e)
   {
      // value does not need to be declared since it is globally declared
      // redeclaring here will overwrite the variable only locally
      value = value + 1;
   }
   // click event of a button which declares the variable locally
   private void btnLocalVariable_Click(object sender, RoutedEventArgs e)
   {
      // declare value as a local variable, overrule the global declared
         variable only in this fucntion
      int value = 5;
      MessageBox.Show("Lokale variabele value is = " + value.ToString());
   }
}
```

2.3 Demo 3: Debug oplossingen demo's 1 en 2

Om de correctheid van de code na te gaan, worden voorgaande oplossingen gedebugged. Debuggen in C# kan men via onderstaande werkwijzen:

- Breakpoints: zet een rood bolletje in de kantlijn van de code. Het programma zal hier stoppen tijdens de uitvoering.
- Eens de breakpoint bereikt is, kan men de code stap voor stap uitvoeren.
- Tijdens het debuggen (breakpoint, stap voor stap uitvoeren), kan men de waarde van de variabelen opvragen in de *watch list*.

2.4 A: Verschillend van?

Laat de gebruiker toe om 2 gehele getallen via textboxen in te geven. Wanneer de gebruiker op de knop "Vergelijken" klikt, wordt nagegaan of de 2 getallen verschillend zijn van elkaar. Indien beide getallen verschillend zijn, wordt "Succes" weergegeven in een label. In het andere geval verschijnt er "Oeps".

2.5 A: Binnen bereik

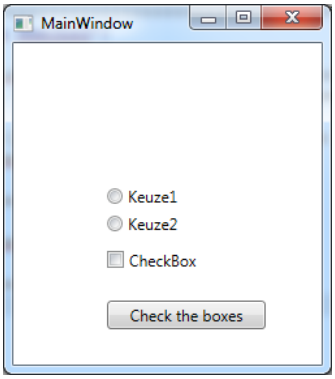
Schrijf een programma dat een getal van de gebruiker opvraagt. Via de knop "Vergelijk" wordt nagegaan of deze waarde zich tussen -100 en +100 bevindt. Indien dit zo is, wordt "Binnen bereik" in een messagebox getoond. In het andere geval geeft de messagebox "Too bad, retry!!!" weer.

2.6 A: I'm checked!

Checkboxen en radiobuttons komen relatief vaak voor wanneer men formulieren gebruikt. In deze opgave is het de bedoeling dat je zowel 2 radiobuttons als een checkbox op het formulier plaatst. Via een button ga je na of deze al dan niet aangevinkt zijn. Geef het resultaat weer in een aparte label. Gebruik onderstaande voorbeeldjes om tot het gewenste resultaat te komen. Merk op dat het vergelijken van de waarden van checkboxen en radiobuttons a.d.h.v. de "??" operator verloopt. Deze operator wordt gebruikt wanneer men type "bool?" tegenkomt.

Codefragment 3: Gebruik van radiobuttons en checkboxen in WPF.

```
if (chbox1.IsChecked ?? true)
{
   MessageBox.Show("Im checked");
}
if (radiobtn1.IsChecked ?? true)
{
   MessageBox.Show("Radiobutton 1 is checked");
}
```



Figuur 1: Voorbeeldprogramma.

2.7 A: Nulpunten 2^{de}-graads vergelijking

Bereken de nulpunten van een 2^{de}-graadsvergelijking. A, b en c zijn door de gebruiker in te voeren. Als resultaat worden er ofwel geen nulpunten, ofwel 1 nulpunt ofwel 2 nulpunten weergegeven. De formules voor het berekenen van de nulpunten worden hieronder opgelijst.

$$D = b^2 - 4ac \tag{1}$$

Als $D < 0$, dan zijn er geen nulpunten.
Als $D = 0$, dan is er 1 nulpunt, nl:

$$x = \frac{-b}{2a} \tag{2}$$

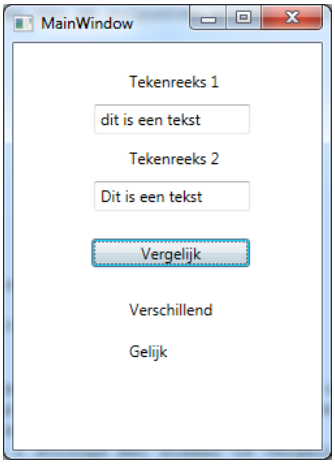
Als $D > 0$, dan zijn er 2 nulpunten, nl:

$$x_{1,2} = \frac{-b \pm \sqrt{D}}{2a} \tag{3}$$

Hint: In deze opgave kan je if-structuren nesten of een if/else if/else structuur gebruiken.

2.8 E: StringCompare

Schrijf een opgave dat 2 tekenreeksen (strings) inleest. Beide strings worden met elkaar vergeleken en het resultaat wordt in een label geplaatst. Nadien worden alle letters van beide strings omgezet in hoofdletters. Beide strings worden opnieuw tegen elkaar vergeleken, waarbij het resultaat toegevoegd wordt aan het voorgaande resultaat. Zorg ervoor dat beide resultaten onder elkaar in dezelfde label verschijnen.
Hint: Gebruik {naamstring}.ToTupper() om een een kopie van de string in hoofdletters te verkrijgen.
Hint: Gebruik "\n\r" om een nieuwe regel te starten in een label.



Figuur 2: Voorbeeldprogramma stringcompare.

2.9 E: Min-Max

Laat de gebruiker toe om 10 kommagetallen in te geven. Voorzie hiervoor 1 textbox en 1 knop om getal per getal in te voeren. Van alle ingevoerde getallen worden de kleinste en grootste waarden bijgehouden. Telkens een nieuw getal ingevoerd wordt, wordt dit getal bijgehouden indien deze kleiner of groter is dan de bestaande waarden. Na 10 ingevoerde getallen worden de grootste en kleinste waarden geprint.
Hint: Gebruik globale variabelen.

2.10 X: Gokmachine

Een speler probeert geld te winnen aan een gokmachine. Hierbij moet de speler binnen de 3 pogingen een randomgetal tussen 0 en 10 raden. Het randomgetal wordt via een randomgenerator bepaald (zie codefragment 4) en wordt als globale variabele bijgehouden. De gebruiker kan het getal raden door een getal in te geven en door op "Raden" te klikken. Hierbij wordt bijgehouden hoeveel pogingen gespeeld zijn. Indien het aantal progingen groter is dan 3, is het spel afgelopen. Op dat moment wordt "Mouhahaha" in een label geprint. Indien de speler het getal raadt, wordt in een label "Hoera" geprint.
Hint: Declareer de randomgenerator als globale variabele. Laat de gebruiker toe om via een aparte knop "Nieuw randomgetal" een nieuw randomgetal te genereren. Tijdens het aanmaken van dit nieuw randomgetal wordt ook het aantal pogingen op 0 gezet.

Codefragment 4: Randomgetallen genereren

```
1 // globaal declareren
2 Random rnd = new Random();
3 int randomwaarde;
4 // lokaal gebruiken
5 randomwaarde = rnd.Next(0,10);
```

Figuur 3: Voorbeeldprogramma randomspel.