Aanvullende tentamen Imperatief programmeren
Woensdag 5 januari 2011, 8.30–10.30 uur

- Schrijf op elk ingeleverd blad je naam.
  Schrijf op het eerste blad ook je studentnummer en het aantal ingeleverde bladen.
- De lijst met standaardfuncties na afloop graag weer inleveren.
  De opgaven mag je houden (behalve als je heel vroeg vertrekt).
- Het tentamen bestaat uit 3 opgaven. Opgave 1 en 2 tellen voor 30% mee, opgave 3 telt voor 40%.
  Als je een deel van een opgave niet weet, probeer dan toch zo veel mogelijk op te schrijven!

Veel succes!

---

1. *Van onderstaande zes onderdelen mag je er één overslaan. De andere vijf tellen elk voor 2 punten.*
   Geef duidelijk aan welke je overslaat.
   Maak je ze toch alle zes, dan tellen ze elk voor 1.67 punten
   (als je er eentje fout beantwoordt, had je hem dus beter kunnen overslaan!)

   (a) Met een `Thread` object kun je een tekenfilmpje maken. Hoe doe je dat?

   (b) Iemand schrijft de volgende opdracht om het gemiddelde van eerder uit een file ingelezen getallen op het label `lab` op het scherm te tonen:

   ```
   lab.Text = "Gemiddelde: " + totaal / aantal;
   ```

   In sommige situaties wordt het programma echter afgebroken met een foutmelding.

   In plaats daarvan willen we liever dat de foutmelding op de label verschijnt.
   Je kunt dit op twee manieren voor elkaar krijgen:
   - Vooraf controleren of de foutsituatie zich zou gaan voordoen
   - Als de foutsituatie optreedt deze afhandelen

   Geef voor beide aanpakken aan hoe de opdracht er dan uit komt te zien.

   (c) In C# kan een methode worden aangeroepen met het speciale object `base` voor de punt.
   (In Java heet dat speciale object `super`).
   - In welke situatie is dat zinvol? (geef een abstracte beschrijving van de situatie)
   - Geef een voorbeeld van een praktijkgeval waar deze situatie zich voordoet.

   (d) Bij het schrijven van een tekstfile kun je kiezen uit een aantal verschillende *encodings*.
   Mogelijk zijn onder andere Ascii, Latin1 (ook bekend als iso-8859-1), Unicode, en UTF8.
   - Noem een voordeel en een nadeel van Latin1 in vergelijking met Unicode.
   - Noem een voordeel en een nadeel van UTF8 in vergelijking met Unicode.

   (e) Je kunt een object van het type `A` declareren en een waarde geven met `A a = new A()`.
   Het maakt daarbij uit of `A` is gedefinieerd als `class A` of als `struct A`.

   Wat is het verschil?

   (f) Sommige arrays zijn *drie-dimensionaal*.
   - Hoe ziet de declaratie van een drie-dimensionale array van strings er uit, en hoe kan daarbij het aantal elementen worden vastgelegd?
   - Geef een voorbeeld van een praktijksituatie waarin deze array zinvol gebruikt kan worden.

zie achterkant

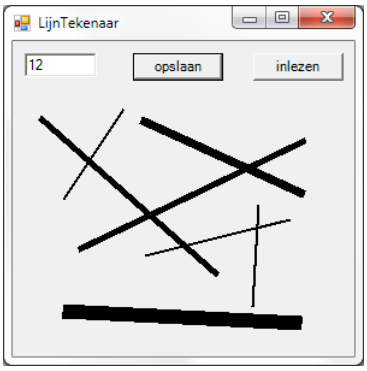
2. In onderstaand programma wordt een array gedeclareerd met daarin de uitslagen van een tentamen op een schaal van 0 tot en met 10:

```
using System.Windows.Forms;
using System.Drawing;

public class StaafDiagram : Form
{
    double [] cijfers
    = { 10, 8, 9, 7, 10, 6.5, 8, 7.5, 9, 7.5, 8, 8.5, 4, 6.5, 8.5, 8, 8, 8, 7, 8.5, 9, 8.5, 4, 5.5, 9, 9
      , 8.5, 9.5, 4, 9, 9, 8, 7.5, 6.5, 8, 7, 8, 9.5, 8, 8, 8, 6, 7.5, 9, 8, 9, 6, 8.5, 7, 9, 6, 8.5, 8.5
      , 9, 8, 6, 8, 7, 8.5, 7.5, 8, 7.5, 8, 8, 9, 9, 8.5, 8.5, 9.5, 9, 8.5, 0, 5.5, 8.5, 8, 7
      };

    public StaafDiagram()
    {   this.Text   = "Staafdiagram";  this.Size = new Size(500, 280);
        this.Paint += this.teken;
    }
    static void Main()
    {   Application.Run(new StaafDiagram());
    }
    // TODO: methode teken
}
```

Schrijf de ontbrekende methode `teken` van dit programma, waarmee een staafdiagram van deze uitslag op het scherm wordt getekend, zoals in de afbeelding. De uitslagen worden daarbij naar beneden afgerond, dus bijvoorbeeld een 7.5 wordt meegeteld in de staaf behorend bij 7.
Het programma moet ook gebruikt kunnen worden als, later, de array wordt aangepast (waarna het programma opnieuw wordt gecompileerd).
Hint: het ankerpunt bij het tekenen van een tekst ligt linksboven. De staven liggen 20 beeldpunten uit elkaar en gebruiken een breedte van 10 beeldpunten voor elke uitslag.

zie vervolgblad

3. Bekijk het programma `LijnTekenaar`, waarvan hiernaast een screenshots staat. In het window zijn een tekstveld en twee buttons zichtbaar. De gebruiker kan met de muis steeds twee punten aanklikken, die dan verbonden worden door een lijn. Het aantal lijnen dat de gebruiker mag tekenen is niet aan een maximum gebonden.
De dikte van de lijn wordt gespecificeerd door het getal dat in het tekstveld staat ingevuld op het moment van de tweede klik. Je mag zonder controle aannemen dat in het tekstveld alleen maar cijfertekens zijn ingevuld.



Hieronder is al een deel van het programma gegeven. Vul hierop het volgende aan:

(a) Schrijf een hulpklasse `Lijn`, zo dat in een object van die klasse alle gegevens die nodig zijn om een lijn te kunnen tekenen beschikbaar zijn. Maak een constructormethode die zo'n object van beginwaardes voorziet, een methode `ToString` die de waarden 'inpakt' in een string, en een tweede constructormethode die zo'n string weer 'uitpakt'.

(b) Schrijf de methoden `muisklik` en `teken` in de klasse `LijnTekenaar`, en geef de declaraties van de daarvoor benodigde member-variabelen.

(c) Als de gebruiker op de knop 'opslaan' drukt, wordt de toestand van de tekening opgeslagen in een file waarvan de naam in de constante `naam` staat. Als de gebruiker op de knop 'inlezen' drukt, wordt de huidige tekening vervangen door de opgeslagen tekening. Schrijf de methoden `opslaan` en `inlezen` die daarvoor nodig zijn.

```
using System;
using System.Windows.Forms;
using System.Drawing;
using System.Collections.Generic;
using System.IO;

namespace Opgave3
{   public class Program
    {   static void Main()
        {   Application.Run(new LijnTekenaar());
        }
    }
    public class Lijn
    {   // OPGAVE a
    }
    public class LijnTekenaar : Form
    {   const string naam = "tekening.txt";
        TextBox tb;

        public LijnTekenaar()
        {   this.Text = "LijnTekenaar";
            Button b1, b2;
            tb = new TextBox(); tb.Text="5";       tb.Location=new Point(10,10); tb.Size=new Size(60,30);
            b1 = new Button();  b1.Text="opslaan"; b1.Location=new Point(100,10);
            b2 = new Button();  b2.Text="inlezen"; b2.Location=new Point(200,10);
            this.Controls.Add(tb); this.Controls.Add(b1); this.Controls.Add(b2);
            this.MouseClick += this.muisklik;
            this.Paint      += this.teken;
            b1.Click        += this.opslaan;
            b2.Click        += this.inlezen;
        }
        // OPGAVE b en c
    }
}
```

EINDE TENTAMEN