Uitwerking Tweede deeltentamen Imperatief programmeren – versie 1
Vrijdag 20 oktober 2017, 11.00-13.00 uur

1. In een constructormethode
   (a) moet je een nieuw object van de klasse aanmaken
   (b) • kun je membervariabelen een beginwaarde geven
   (c) kun je membervariabelen declareren
   (d) mag je de waarde `this` niet gebruiken

2. In de header van een `for`-opdracht staan twee puntkomma's.
   Wat mag er *niet* tussen deze twee puntkomma's staan?
   (a) alleen lege ruimte
   (b) `false`
   (c) een `bool` variabele
   (d) • een `int` expressie

3. De klasse `object`
   (a) is de enige klasse zonder methodes
   (b) is de enige klasse zonder subklasses
   (c) • is automatisch een superklasse van alle andere klasses
   (d) erft de methodes van zijn superklasse

4. De situatie die ontstaat door `class A : B { C D; }` kan beschreven worden door
   (a) `B` heeft het type `A`
   (b) • `D` heeft het type `C`
   (c) `B` heeft een member `D`
   (d) `A` heeft een member `B`

5. De expressie `3 * '2'`
   (a) heeft waarde `'6'`
   (b) heeft waarde `"222"`
   (c) • heeft waarde `150`
   (d) is niet goed getypeerd

6. De `object`-parameter van een eventhandler is
   (a) • het object dat het event heeft veroorzaakt
   (b) het object dat de eventhandler heeft geregistreerd
   (c) het object dat het event afhandelt
   (d) een object dat relevante details over het event bevat

7. De expressie `"hallo"[2]`
   (a) is niet mogelijk, want een string is geen array
   (b) is niet mogelijk, want `"hallo"` is geen variabele
   (c) • is wel mogelijk, want `String` heeft een indexer-property

(d) is wel mogelijk, want `String` heeft een array-property

8. Als er is gedefinieerd:

```
class A  { int x; }
struct B { A y;   }
B z;
```

Wat is dan de uitkomst van de expressie `z.y` ?

(a) er treedt een 'null reference exception' op
(b) • de waarde `null`
(c) de waarde `0`
(d) een verwijzing naar een `A`-object waarin `x` de waarde `0` heeft

9. Iemand schrijft een methode om een getal `x` tot een niet-negatieve macht `e` te verheffen:

```
int Macht(int x, int e)
{
    int res = x;
    for (int t=1; t<e; t++)
        res *= x;
    return res;
}
```

Welk ongewenst effect heeft deze methode?

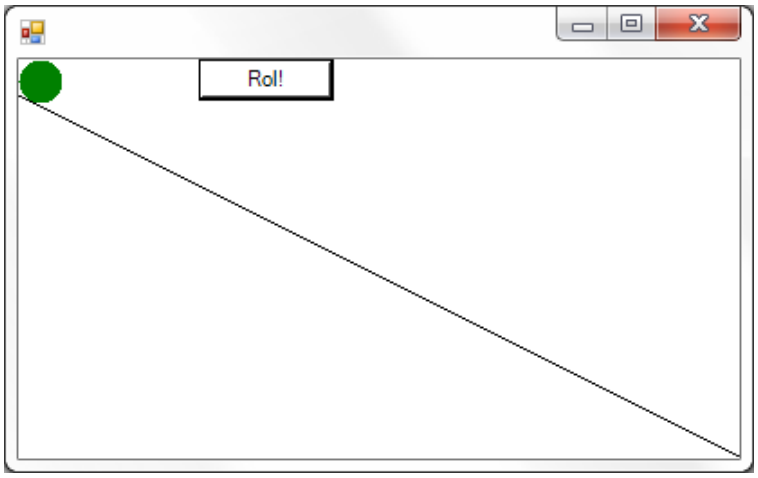
(a) de herhaling gaat één stap te lang door
(b) de uitkomst is altijd 1
(c) het werkt niet als `x` gelijk is aan 1
(d) • het werkt niet als `e` gelijk is aan 0

10. Om de frequentie-verdeling te bepalen van de waarden die voorkomen in de array `a`, schrijf je

(a) • `for (int t=0; t<a.Length; t++) b[a[t]]++;`
(b) `for (int t=0; t<b.Length; t++) a[b[t]++];`
(c) `for (int t=0; t<a.Length; t++) b[a[t]]+1;`
(d) `for (int t=0; t<b.Length; t++) a[b[t]]++;`

11. Dit programma tekent een balletje bovenaan een helling. Als de gebruiker op de knop drukt, moet het balletje in een 'tekenfilmpje' van circa 10 seconden van de berg af (en daarna uit beeld) rollen. Schrijf de daarvoor benodigde methode `rol` en eventuele extra hulp-methoden.

```
public class HellendVlak : Form
{
    static void Main()
    {   Application.Run(new HellendVlak());
    }
    public HellendVlak()
    {   Button b = new Button(); this.Controls.Add(b);
        b.Text = "Rol!";         b.Location = new Point(100, 0);
        b.Click += this.rol;     this.Paint += this.teken;
    }
    int x = 0, y = 0;
    public void teken(object obj, PaintEventArgs pea)
    {   pea.Graphics.DrawLine (Pens.Black, 0, 20, 400, 220);   // helling
        pea.Graphics.FillEllipse(Brushes.Green, x, y, 24, 24); // balletje
    }
}
```



Antwoord:

```
public void rol(object obj, EventArgs ea)
{   Thread t = new Thread(run);
    t.Start();
}
public void run()
{   while (true)
    {   x+=2; y+=1;
        this.Invalidate();
        Thread.Sleep(50);
    }
}
```

12. Gegeven zijn de volgende twee declaraties van variabelen:

```
Label lab;
double waarde;
```

Iemand schrijft de volgende opdracht om de wortel van de waarde te berekenen en aan de gebruiker te tonen:

```
lab.Text = "De wortel is " + Math.Sqrt(waarde);
```

Maar als `waarde` negatief is wordt het programma afgebroken met een foutmelding.

In plaats daarvan willen we liever dat de tekst 'onmogelijk' op de label verschijnt. Je kunt dit op twee manieren voor elkaar krijgen:

- Vooraf controleren of de foutsituatie zich gaat voordoen
- De wortel gewoon maar uitrekenen en de foutsituatie opvangen

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

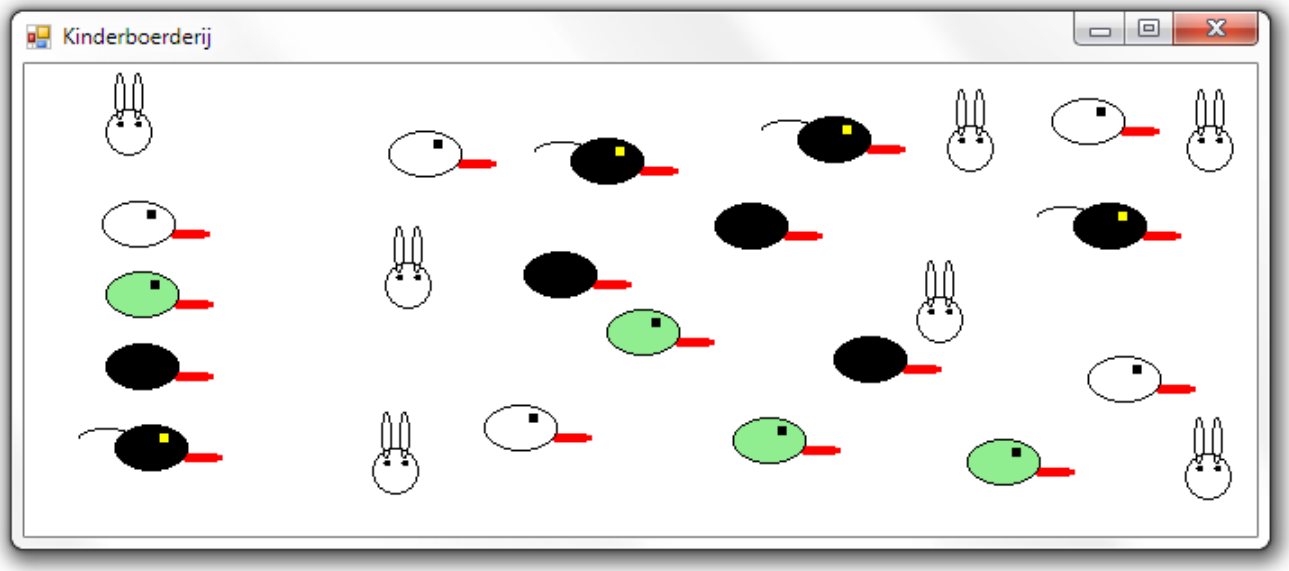
Antwoord:

```
// aanpak 1: vooraf controleren
if (waarde<0)
     lab.Text = "onmogelijk";
else lab.Text = "De wortel is " + Math.Sqrt(waarde);

// aanpak 2: foutsituatie afvangen
try
{   lab.Text = "De wortel is " + Math.Sqrt(waarde);
```

```
}
catch (Exception e)
{   lab.Text = "onmogelijk";
}
```

13. Bekijk het gegeven programma op de pagina hiernaast. De gebruiker kan er dierenkoppen mee tekenen: overal waar de gebruiker klikt met de linker muisknop onstaat de kop van een eend, en met de rechter muisknop de kop van een konijntje. Na een aantal kliks zou het scherm er zo uit kunnen zien:



Er kunnen maximaal 100 dieren getekend worden. Als de gebruiker daarna nog vaker klikt gebeurt er niets (ook geen foutmelding).

Er zijn vier soorten eenden:

- de gewone eend heeft een witte kop en een zwart oog
- de wilde eend heeft een groene kop en een zwart oog
- de bergeend heeft een zwarte kop en een zwart oog
  (waardoor het oog dus eigenlijk wegvalt tegen de achtergrond van de kop)
- de kuifeend heeft een zwarte kop met een geel oog en een sierlijke kuif achter op z'n kop.

(a) Er ontbreken nog declaraties in de klasse `Kinderboerderij`.
Schrijf deze declaraties en hun initialisatie.

(b) Schrijf de ontbrekende methode `klik`.
Als de gebruiker met de rechter muisknop klikt, moet er een konijntje verschijnen. Met de linker muisknop verschijnen er afwisselend de vier soorten eenden. Je kunt daarbij gebruik maken van de bestaande methode `eendKeuze`. Hint: gebruik de bijlage om te zien hoe je aan de parameters van `klik` kunt zien welke muisknop is gebruikt.

(c) Schrijf de ontbrekende klasse `Dier`.
Welke members en/of methoden daarin nodig zijn blijkt uit de rest van het programma.

(d) In de klasse `Eend` worden twee methoden gebruikt die nog niet zijn gedefinieerd.
Definieer deze methoden, zo dat de eend correct wordt getekend.
Houd daarbij wel alvast rekening met de uitbreidingen in de opgave e en f.

(e) Schrijf de ontbrekende klassen `WildeEend` en `BergEend`.
Vermijd daarbij zo veel mogelijk het dupliceren van code.

(f) Schrijf de ontbrekende klasse `KuifEend`.
Vermijd ook hier weer zo veel mogelijk het dupliceren van code.

Antwoord:

```
// opgave a
    Dier[] dieren = new Dier[100];
    int aantalDieren = 0;
    int aantalEenden = 0; // nodig in opgave c
// opgave b
    public void klik(object obj, MouseEventArgs mea)
    {   Dier dier;
        if (aantalDieren<100)
        {   if (mea.Button == MouseButtons.Right)
                dier = new Konijn();
            else
            {   dier = this.eendKeuze(aantalEenden % 4);
                aantalEenden++;
            }
            dier.Plek = mea.Location;
            dieren[aantalDieren] = dier;
            aantalDieren++;
            this.Invalidate();
        }
    }
// opgave c
class Dier
{   public Point Plek;
    public virtual void LaatZien(Graphics g) {}
}
// opgave d
   public virtual Brush KopKleur()
    {   return Brushes.White;
    }
    public virtual Brush OogKleur()
    {   return Brushes.Black;
    }
// opgave e
class WildeEend : Eend
{   public override Brush KopKleur()
    {   return Brushes.LightGreen;
    }
}
class BergEend : Eend
{   public override Brush KopKleur()
    {  return Brushes.Black;
    }
}
// opgave f
class KuifEend : BergEend
{   public override Brush OogKleur()
    {   return Brushes.Yellow;
    }
    public override void LaatZien(Graphics g)
    {   base.LaatZien(g);
        g.DrawArc(Pens.Black, Plek.X - 20, Plek.Y + 2, 30, 10, 180, 180); // kuif
    }
}

public class Kinderboerderij : Form
```

```
{
    // TODO opgave a: declaraties

    public Kinderboerderij()
    {   this.Text = "Kinderboerderij";
        this.Size = new Size(700, 300);
        this.BackColor   = Color.White;
        this.MouseClick += this.klik;
        this.Paint      += this.teken;
    }
    static void Main()
    {   Application.Run(new Kinderboerderij());
    }
    public void teken(object obj, PaintEventArgs pea)
    {   for (int t=0; t<aantalDieren; t++)
            dieren[t].LaatZien(pea.Graphics);
    }
    private Eend eendKeuze(int n)
    {
        if (n==1) return new WildeEend();
        if (n==2) return new BergEend();
        if (n==3) return new KuifEend();
        return new Eend();
    }
    // TODO opgave b: methode klik
}

// TODO opgave c: klasse Dier

class Konijn : Dier
{   public override void LaatZien(Graphics g)
    {   g.DrawEllipse(Pens.Black,    this.Plek.X+5,  this.Plek.Y-20,  5, 25); // linkeroor
        g.DrawEllipse(Pens.Black,    this.Plek.X+15, this.Plek.Y-20,  5, 25); // rechteroor
        g.DrawEllipse(Pens.Black,    this.Plek.X,    this.Plek.Y,    25, 25); // kop
        g.FillEllipse(Brushes.Black, this.Plek.X+6,  this.Plek.Y+6,   4,  4); // linkeroog
        g.FillEllipse(Brushes.Black, this.Plek.X+16, this.Plek.Y+6,   4,  4); // rechteroog
    }
}

class Eend : Dier
{   public override void LaatZien(Graphics g)
    {   g.FillEllipse  (Brushes.Red,     this.Plek.X+35, this.Plek.Y+15, 25,  6); // snavel
        g.FillEllipse  (this.KopKleur(), this.Plek.X,    this.Plek.Y,    40, 25); // kop
        g.DrawEllipse  (Pens.Black,      this.Plek.X,    this.Plek.Y,    40, 25); // rand vd kop
        g.FillRectangle(this.OogKleur(), this.Plek.X+25, this.Plek.Y+5,   5,  5); // oog
    }
    TODO opgave d: ontbrekende methoden
}

TODO opgave e: klasse WildeEend en BergEend
TODO opgave f: klasse KuifEend
```

14. Gegeven is het volgende console-programma:

```
class Program
{   static void Main()
    {   A a = new A();
        a.X = double.Parse( Console.ReadLine() );
        a.Y = double.Parse( Console.ReadLine() );
        Console.WriteLine("gemiddeld: " + a.Gem );
```

```
        //  Console.WriteLine("getal: " + a.X   );
        Console.ReadLine();
    }
}
```

Schrijf de ontbrekende klasse `A` zo, dat:

- Bij het runnen van het programma het gemiddelde van de twee ingelezen getallen wordt getoond
- Als de uitgecommentarieerde regel wordt teruggezet, de compiler een fout aangeeft: '... A.X cannot be used in this context ...'.

Antwoord:

```
class A
{   private double x, y;
    public double X   { set { x = value;      } }
    public double Y   { set { y = value;      } }
    public double Gem { get { return (x+y)/2; } }
}
```

15. In de klasse String zit onder andere de volgende methode:

```
string Insert(int n, string s)
```

Deze methode `Insert` geeft een nieuwe string waarbij op de positie aangeduid door de eerste parameter, de string in de tweede parameter wordt ingevoegd. Als de positie negatief is wordt de string vooraan ingevoegd, als de positie te groot is wordt de string achteraan toegevoegd. Voorbeelden:
`"hallo".Insert( 2, "XYZ")` geeft `"haXYZllo"`
`"hallo".Insert( 0, "XYZ")` geeft `"XYZhallo"`
`"hallo".Insert(-5, "XYZ")` geeft `"XYZhallo"`
`"hallo".Insert( 7, "XYZ")` geeft `"halloXYZ"`

Stel dat je de auteur van de klasse `String` bent. Enkele methoden en properties zijn al geschreven: de indexerings-property om een losse letter te pakken, de `Length`-property, de operator `+`, of als je wilt de methode `Concat` zijn al beschikbaar (die mag je dus gebruiken). Maar veel andere methoden (zoals `IndexOf`, `Substring`, `StartsWith` en `EndsWith`) ontbreken nog (die mag je dus niet aanroepen). Je mag wel zelf extra hulpmethodes schrijven en die aanroepen.

Schrijf de methode `Insert` in de klasse `String`.

Antwoord:

```
public string Insert(int n, string s)
{   int t;
    string res = "";
    for (t=0; t<n && t<this.Length; t++)
        res += this[t];
    res += s;
    for ( ; t<this.Length; t++)
        res += this[t];
    return res;
}
```