

UITWERKING EERSTE DEELTENTAMEN IMPERATIEF PROGRAMMEREN  
MAANDAG 6 DECEMBER 2010, 11.00-13.00 UUR

---

1. Deze opgave bestaat uit een aantal tekstvragen.

Houd het antwoord kort: een of twee zinnen per onderdeel kan al genoeg zijn.

(a) In welke gevallen staat er het woord `void` in de header van een methode?

Wat heeft dat voor gevolg voor de syntactische status van een aanroep van zo'n methode?

**Antwoord:** `void` staat in de header van een methode zonder resultaat. De aanroep van zo'n methode heeft geen waarde en is dus een opdracht in plaats van een expressie.

(b) In C# moet elke variabele die in het programma wordt gebruikt *gedecclareerd* worden. Noem twee voordelen van een taal waarin declaraties gebruikt worden, boven een taal waarin variabelen niet gedecclareerd hoeven te worden.

**Antwoord:** De compiler kan waarschuwen voor tikfouten bij het gebruik van een variabele. De compiler kan controleren of het type van methode-parameters correct is. De bedoeling van het programma wordt duidelijker voor de menselijke lezer.

(c) Wat is een object?

Een variabele met een object-type bevat in sommige gevallen het object zelf, maar soms bevat het een ander soort waarde. Wat voor soort waarde is dat, en wanneer is dat het geval?

**Antwoord:** Een object is een groepje variabelen dat bij elkaar hoort. Als het type van het object een *class* is (en niet een *struct*) bevat een object-variabele niet het object zelf, maar een *verwijzing* daarnaar.

(d) Wat is het effect van de opdracht `x+=y`; ?

Hoe hangt de waarde van `p&&q` af van de waarden van `p` en `q` ?

**Antwoord:** Het effect van `x+=y`; is dat de variabele `x` wordt vermeerderd met de waarde van `y`. De waarde van `p&&q` is `true` als `p` en `q` allebei de waarde `true` hebben, en `false` in alle andere gevallen.

(e) Waaraan herken je de definitie van een *constructormethode* ?

Op welke manier wordt een constructormethode aangeroepen?

**Antwoord:** De constructormethode heeft dezelfde naam als de klasse. Hij wordt aangeroepen met het woord `new`, gevolgd door de naam en parameters.

2. Hieronder staat 16 fragmenten uit een programma. Schrijf op je antwoordblad een blok van 4 bij 4 vakjes en zet in elk vakje een letter passend bij het overeenkomstige fragment:

- **T** als het programmafragment een **type** is
- **E** als het programmafragment een **expressie**
- **O** als het programmafragment een **opdracht** is
- **D** als het programmafragment een **declaratie** is
- **H** als het programmafragment een **methode-header** is
- **X** als het programmafragment geen van bovenstaande dingen is

```
class Hallo : Form      (int)x          0xff          return false;
Form f;                 int a(int x)    !x!=x         bool b()
Form                    x(int)           { }           decimal
new Form()              int x=1;        const         for(;;);
```

**Antwoord:**

```
X E E O
D H E H
T X O T
E D X O
```

Opgave 3 en 4 vragen een stukje programma. Kleine schrijffoutjes (hoofdletters, puntkomma's enz.) worden niet streng afgerekend, maar de elementen die de structuur van het programma bepalen (haakjes, accolades, aanhalingstekens enz.) zijn wel belangrijk. Schrijf die dus duidelijk en op de goede plaats op! Het is toegestaan (maar niet nodig) om C#-constructies die (nog) niet zijn behandeld toch te gebruiken. Je hoeft niet aan te geven welke using-opdrachten nodig zijn om de klassen te kunnen gebruiken.

3. Alle methodes in deze opgave maken deel uit van de klasse `Prog`.

- (a) Schrijf een methode `Kopieer` die een string als resultaat oplevert die uit een bepaald aantal kopieën van een bepaalde string bestaat. Zowel het aantal als de te kopiëren string zijn een parameter van deze methode.

**Antwoord:**

```
static string Kopieer(int n, string x)
{
    string res = "";
    int t;
    for (t=0; t<n; t++)
        res += x;
    return res;
}
```

- (b) Bekijk de volgende `Main` methode:

```
static void Main()
{
    Console.WriteLine( Programma.Sneeuw(6) );
}
```

Als dit programma wordt uitgevoerd, verschijnt de volgende tekst op de console:

```
.....*.....
...***....
..*****..
.*****.
.*****.
*****
```

De parameter van de methode `Sneeuw` bepaalt het aantal regels tekst. In dit voorbeeld is dat 6, maar als er een groter getal wordt gebruikt verschijnen er meer regels, die ook langer zijn: steeds zo, dat er een driehoek van sterretjes ontstaat, met stippen daar omheen.

Als de parameter 1 is, verschijnt er alleen een enkel sterretje.

Als de parameter 0 is of kleiner, dan verschijnt er helemaal niets.

Opgave: schrijf deze methode `Sneeuw`.

**Antwoord:**

```
static string Sneeuw(int n)
{
    string res="", punt, ster;
    int t;
    for (t=0; t<n; t++)
    {
        punt = Prog.Kopieer(n-1-t, ".");
        ster = Prog.Kopieer(2*t+1, "*");
        res += punt+ster+punt+"\n";
    }
    return res;
}
```

4. Gegeven is de volgende klasse:

```
class Program
{
    public static void Main()
    {
        Toverbal t = new Toverbal();
        t.Text = "Toverbal";
        Application.Run(t);
    }
}
```

Schrijf de klasse `Toverbal`, zo dat het programma zich als volgt gaat gedragen.

Er verschijnt een window met een lengte en breedte van 256 beeldpunten. De rand van het window is daarin niet meegerekend; het totale window zal dus iets groter worden, afhankelijk van de instellingen van het operating system.

In het window is een opgevulde cirkel zichtbaar. De cirkel beweegt mee met de muis: het midden van de cirkel staat op de plaats van de muispointer. (Als de muis vlakbij de rand van het window staat, valt een deel van de cirkel buiten de rand).

In het window is bovendien een tekst-invulveld zichtbaar. Daarin kan de gebruiker een getal intikken. Dat getal wordt gebruikt als straal (afstand van de rand tot het middelpunt) van de cirkel. Aan het begin staat er alvast 10 ingevuld, maar dat kan later door de gebruiker veranderd worden. Je mag zonder controle aannemen dat gebruiker alleen maar cijfers in het tekstveld invult.

De *kleur* van de cirkel varieert met de plek in het window:

- in de hoek *links boven* is de kleur *zwart*
- in de hoek *links onder* is de kleur *groen*
- in de hoek *rechts boven* is de kleur *rood*
- in de hoek *rechts onder* is de kleur *geel*

Bij het bewegen van de muis verandert de kleur vloeiend via tussenliggende kleuren. Bijvoorbeeld: middenboven is de kleur donkerrood, middenrechts is de kleur oranje, en precies in het midden is de kleur donkergeel.

**Antwoord:**

```
class Toverbal : Form
{   TextBox tb;
    int x, y;

    public Toverbal()
    {   this.ClientSize = new Size(256,256);
        tb = new TextBox();
        tb.Size = new Size(100,20);
        tb.Location = new Point(10,10);
        tb.Text = "10";
        this.Controls.Add(tb);
        this.Paint += teken;
        this.MouseMove += beweeg;
    }
    public void beweeg(object o, MouseEventArgs mea)
    {   x = mea.X;
        y = mea.Y;
        this.Invalidate();
    }
    public void teken(object o, PaintEventArgs pea)
    {   int r = int.Parse(tb.Text);
        Brush br = new SolidBrush( Color.FromArgb(x,y,0) );
        pea.Graphics.FillEllipse(br, x-r, y-r, 2*r, 2*r);
    }
}
```

<b>EINDE TENTAMEN</b>
-----------------------