Eerste deeltentamen Imperatief programmeren
Vrijdag 30 september 2016, 16.00–18.00 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 en de bijbehorende antwoorden komen vanavond op de website.
- Het tentamen bestaat uit 4 opgaven. Elke opgave telt even zwaar mee. Als je een deel van een opgave niet weet, probeer dan toch zo veel mogelijk op te schrijven!
- Opgave 3 en 4 vragen een stukje programma. 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`-directieven nodig zijn om de klassen te kunnen gebruiken.

Veel succes!

---

1. Deze opgave bestaat uit een aantal tekstvragen.
   Houd het antwoord kort: een of twee zinnen per onderdeel kan al genoeg zijn.

   (a) Wat is er in een `static` methode anders dan in methoden die dat niet zijn?
   Wat is er in een `void` methode anders dan in methoden die dat niet zijn?

   (b) Beschrijf de syntax van de aanroep van een static void-methode met één parameter.
   Beschrijf daarna ook de semantiek daarvan.

   (c) Geef één opdracht om in een console-application de twee-regelige tekst

   ```
   Cesar zei:
   "veni vidi vici!"
   ```

   te laten zien (inclusief de leestekens).

   (d) Waaraan herken je de aanroep van een *constructormethode* ?
   Wat gebeurt er bij de aanroep van een constructormethode meer dan bij de aanroep van een gewone methode?

   (e) Welke twee rollen heeft het begrip *klasse* in C# ?
   Hoe hangen deze twee rollen samen?

2. Hieronder staan 16 fragmenten uit een programma. Maak 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

`true`	`while(false);`	`Graphics`	`string s = "//";`
`t+1=t;`	`t!=t`	`t*=1;`	`static s(int i)`
`float`	`return int;`	`(int)1.2`	`float f()`
`Pen p;`	`Pens.Red`	`{ }`	`g.DrawLine(p,0,0,0,0);`

3. In de wiskunde wordt vaak de *exponentiële functie*, of kortweg `exp` gebruikt. Deze kan worden berekend door:

$$exp(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \cdots$$

Hierin betekent 4! de *faculteit* van 4, dat is alle gehele getallen van 1 t/m 4 met elkaar vermenigvuldigd.

Schrijf een methode `exp` die een benadering van deze functie geeft door 20 termen van deze reeks op te tellen. Het is hierbij niet toegestaan om de methoden `Math.Exp` of `Math.Pow` of de operator `^` te gebruiken. Het is ook niet toegestaan om de 20 termen helemaal uit te schrijven. Wel mag je zelf extra hulp-methoden schrijven en aanroepen.

4. Gegeven is de volgende klasse:

```
class Program
{   public static void Main()
    {   Kralen t = new Kralen();
        t.Text = "Kralen";
        t.ClientSize = new Size(500, 500);
        Application.Run(t);
    }
}
```

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

Er zijn twee kralen (gekleurde cirkels met een doorsnede van 30 beeldpunten) te zien:

- een rode, die gecentreerd is op de linker bovenhoek van het window (en die je dus maar voor een kwart ziet)
- een blauwe, die gecentreerd is op de positie van de muis

De twee kralen zijn verbonden door een lijn.

Als de muis beweegt, beweegt de blauwe kraal mee. De kralen blijven verbonden door de lijn.

Elke keer als de gebruiker met de muis klikt, onstaat er een nieuwe kraal op de lijn. De kralen verdelen de ruimte op de lijn gelijkmatig.

De kleur van elke kraal op de lijn is verschillend: ze verkleuren langzaam van rood (in de bovenhoek) naar blauw (bij de muis). Een kraal in het midden bijvoorbeeld is dus paars.

Als de gebruiker heel vaak klikt, gaan de kralen overlappen. Dat is niet erg.

De illustraties hieronder tonen achtereenvolgens: de startsituatie, na 1 keer klikken, na 5 keer klikken, en na 14 keer klikken, waarbij ook de muis steeds ergens anders staat.