

EERSTE DEELTENTAMEN IMPERATIEF PROGRAMMEREN  
VRIJDAG 28 SEPTEMBER 2018, 11.00-13.00 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 en de opgaven na afloop graag weer inleveren.
  - 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!
- 

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 een object?

Hoe wordt in een programma vastgelegd uit welke onderdelen een object is opgebouwd?

(b) De aanroep van een methode kan er uitzien als `xxx.mmm(yyy)`, waarbij `mmm` de naam van de methode is, en `xxx` en `yyy` expressies zijn.

Waar wordt in het programma vastgelegd wat het type van `xxx` moet zijn?

Waar wordt in het programma vastgelegd wat het type van `yyy` moet zijn?

Waar wordt in het programma vastgelegd wat het type van `xxx.naam(yyy)` als geheel is?

(c) Beschrijf de syntax van de `while`-opdracht.

Beschrijf daarna ook de semantiek van de `while`-opdracht.

(d) Beschrijf de betekenis van de symbolencombinatie `*=`

Geef ook een voorbeeld van een opdracht waarin deze symbolencombinatie wordt gebruikt, en leg uit wat deze opdracht voor effect heeft.

(e) Wat voor soort waarden kunnen worden opgeslagen in een variabele van het type `int`, en waarom heet dit type zo?

Wat voor soort waarden kunnen worden opgeslagen in een variabele van het type `double`, en waarom heet dit type zo?

2. Hieronder staat 16 fragmenten uit een programma, gebruikmakend van de klassen uit de bekende standaard-libraries. 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

5%	<code>void p()</code>	<code>return bool;</code>	<code>PaintEventArgs</code>
<code>c++;</code>	<code>Color red;</code>	<code>Math.Sqrt(c)</code>	<code>while(c&gt;0)c-=1;</code>
<code>c==2</code>	<code>new Form()</code>	<code>double d=1,e;</code>	<code>double Sqrt(double 2.0)</code>
<code>byte</code>	<code>false&amp;&amp;true</code>	<code>bool b(bool b)</code>	<code>Console.WriteLine("\n");</code>

*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-directieven 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.

(b) Bekijk de volgende `Main` methode:

```
static void Main()
{   Console.WriteLine( Prog.Druiven(6) );
}
```

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

```
o-o-o-o-o-o
.o-o-o-o-o.
..o-o-o-o..
...o-o-o...
....o-o....
.....o.....
```

De parameter van de methode `Druiven` 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 `o`-tekens ontstaat, met streepjes ertussen, en stippen eromheen.

Als de parameter 1 is, verschijnt er alleen een enkel `o`-teken.

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

Opgave: schrijf deze methode `Druiven`.

4. Gegeven is de volgende klasse:

```
class Program
{
    public static void Main()
    {
        Application.Run(new Kaders());
    }
}
```

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

Er is een window zichtbaar, waarin een vierkant wordt getekend, dat bestaat uit vier kaders met verschillende kleuren. In het midden is het wit, daaromheen een lichtgrijs kader, daar weer omheen een donkergrijs kader, en aan de buitenkant een zwart kader. Zie de afbeelding links.

Het vierkant wordt getekend zo dat de muis steeds in het midden van het vierkant staat. Beweegt de muis, dan beweegt het vierkant mee.

Als de gebruiker op de muis klikt, ontstaan er nieuwe kaders. In de middelste afbeelding heeft de gebruiker vier keer geklikt, en zijn er dus acht kaders. De grijstinten van de kaders blijven het hele scala van wit naar zwart doorlopen, in dit voorbeeld dus met zes tinten grijs ertussen.

In de rechter afbeelding heeft de gebruiker nog eens twaalf keer geklikt, en zijn er dus twintig kaders. De gebruiker heeft de muis hier in het midden van het scherm gezet, zodat het nog net past.

De dikte van de kaders is 10 beeldpunten, en het witte blokje in het midden is dus 20 beeldpunten breed.

Hints: je kunt de kaders ook zien als een stapel vierkanten. Teken het grootste vierkant eerst. Vergeet niet de naam in de titelbalk van het window, en zorg ervoor dat het window zo groot is dat het na 16 keer klikken precies vol is.

