

EERSTE DEELTENTAMEN GAMEPROGRAMMEREN  
WOENSDAG 29 SEPTEMBER 2010, 13.00-15.00 UUR

- **Let op: Lever het tentamen in op twee bladen! Opgaven 1 en 2 moeten uitgewerkt worden op het eerste blad, opgaven 3 en 4 op het tweede blad.** Schrijf op elk ingeleverd blad je naam en je studentnummer.
- De opgaven mag je houden (behalve als je heel vroeg vertrekt).
- Het tentamen bestaat uit 4 opgaven. Elke opgave levert 10 punten op. Je cijfer is het totaal aantal punten gedeeld door 4. Als je een deel van een opgave niet weet, probeer dan toch zo veel mogelijk op te schrijven!

*Veel succes!*

- 
1. (*2 punten per deelvraag*) Deze opgave bestaat uit een aantal tekstvragen. Houd het antwoord kort: een of twee zinnen per onderdeel kan al genoeg zijn.
    - (a) Wat wordt er in een C#-methode aangeduid met `this`? In welke situatie is het niet toegestaan om `this` te gebruiken?
    - (b) Wanneer is een programmeertaal een *imperatieve* programmeertaal?
    - (c) De vertaling van een C#-programma naar machinecode gebeurt in twee stappen. Eerst wordt het programma vertaald naar een *intermediate language*, en die wordt vervolgens weer vertaald naar machinecode. Noem twee voordelen van het vertalen in twee stappen.
    - (d) Wat doen we over het algemeen in de Update-actie van de game loop?
    - (e) Geef een voorbeeld van een type uit een standaardlibrary dat een `struct` is, en een voorbeeld van een type dat een `class` is. Wat is het verschil tussen een struct-type en een class-type?
  2. (*-1 punt per fout*) 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** (maar geen constante) is
    - **O** als het programmafragment een **opdracht** is
    - **D** als het programmafragment een **declaratie** is
    - **C** als het programmafragment een **constante** (en dus ook een expressie) is
    - **X** als het programmafragment geen van bovenstaande dingen is

```
bool b;          new Vector2(0.0, 3.0)    "1+1" = "2";    10<5
int.Parse;      string s = "hallo";    SpriteBatch     while (true) i++;
i+=10;         s.Length              Color.Blue      45
b==i<5         Console.WriteLine(s); s + "!"        /* Commentaar.
```

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. (a) (4 punten) Schrijf een methode `streepjes` met een getal als parameter. Je mag zonder controle aannemen dat de parameter 0 of groter is. De methode moet als resultaat een string opleveren met daarin zoveel verticale streepjes als de parameter aangeeft. Bijvoorbeeld: de aanroep `this.streepjes(5)` levert "||||" op.
- (b) (6 punten) Schrijf een methode `spijker` met een getal als parameter. Je mag zonder controle aannemen dat de parameter 1 of groter is.
- De methode moet als resultaat een string opleveren met daarin het getal in spijkerschrift-notatie. Elk cijfer wordt daarin weergegeven met verticale streepjes, en de cijfers worden gescheiden door een liggend streepje. Er staan ook liggende streepjes aan het begin en het eind. Hier zijn een paar voorbeelden:
- ```
this.spijker(25) geeft "-||-||||-"
this.spijker(12345) geeft "-|-||-|||-||||-||||-"
this.spijker(7) geeft "-||||||-"
this.spijker(203) geeft "-||--||-"
```
- Hint: verwerk eerst het laatste cijfer, en herhaal dan voor de rest van de cijfers.

4. Beschouw de volgende XNA applicatie:

```
class UfoTekenen : Game
{
    static void Main()
    {
        UfoTekenen spel = new UfoTekenen();
        spel.Run();
    }

    SpriteBatch spriteBatch;
    Texture2D ufoPlaatje;

    public UfoTekenen()
    {
        GraphicsDeviceManager graphics = new GraphicsDeviceManager(this);
        Content.RootDirectory = "Content";
    }
    protected override void LoadContent()
    {
        this.spriteBatch = new SpriteBatch(this.GraphicsDevice);
        this.ufoPlaatje = Content.Load<Texture2D>("ufo");
    }
}
```

- (a) (3 punten) Breid dit programma uit, zodat de ufo altijd op de positie van de muis wordt getekend (de muispointer staat in het midden van de ufo).
- (b) (4 punten) Voeg een aantal opdrachten toe zodat de ufo nooit buiten het scherm getekend wordt, en zet deze opdrachten in een aparte methode. Je mag hier geen gebruikmaken van methoden uit de `MathHelper` klasse.
- (c) (3 punten) Tenslotte willen we dat de ufo *niet* getekend wordt als de speler de A-toets ingedrukt houdt. Breid het programma uit zodat dit bewerkstelligd wordt.

EINDE TENTAMEN