CHAPTER 1

Vier op een Rij

FHICT

Leerdoelen	Class, method, constructor, object, private, enum.
Extra	Algoritme,
Vereiste voorkennis	object, class, method, if.
Challenge Type	Programming.

1.1 Vier op een rij

Maak het spel 4 op een rij waarbij je tegen de computer kunt spelen. Houd rekening met onderstaande eisen:

1. Programmeer class *Zet* met twee `integer` `properties` *Rij* en *Kolom* (de `property` *Rij* is `readonly`, deze wordt uitgerekend omdat de stenen naar beneden vallen). De constructor accepteert een *referentie* naar `class` Spel (zie hieronder) en een integer-kolom.

2. Een `enum` `Field` met mogelijke waarden *Rood* en *Geel*.

```
enum Veld {Rood,Geel}
```

3. `Class` *Spel* met intern een private array van 6 bij 7 (tweedimensionaal array)

```
Veld[,] bord = new Veld[6,7];
```

4. Het mag een console-applicatie zijn. Het speelbord hoeft niet te worden afgedrukt, mag wel, in een `Console` `app` kan dat met behulp van `method` `Console.WriteLines` (loop door het `array` en druk het stap voor stap af).

5. De `class` *Spel* heeft methoden als `BedenkEenZet()` om de beste zet voor de computer te bedenken en `AccepteerEenZet(Zet z)` (om een zet van de gebruiker te accepteren).

De methode `BedenkEenZet()` kan in eerste instantie op zoek gaan naar de eerste de beste vrije kolom. Indien er geen vrije kolom meer is dan kan de methode `GelijkSpel()` worden aangeroepen die het spel stopt. Maak private hulpmethoden zoals bool `HeeftSpelerGewonnen()` en `bool HeeftComputerGewonnen()` die je zelf aanroept.

```
private Random Randje;

public Zet BedenkEenZet()
{
   return new Zet(Randje.Next(7));
}
```