



Oefening 93: Figuren filteren

Neem het programma van **oefening 86 Figuren** van **hoofdstuk 15 Overerving en klasse-hiërarchie** er terug bij.

Vervang de methode `ToonFiguur()` met een override van de methode `ToString()`.

Filter: toon enkel rechthoekige figuren (dus geen driehoeken) door de objecten in een `Console.WriteLine()` te plaatsen in de methode `Main()`.



Tip:

Gebruik het sleutelwoord `is`.

Voorbeeld uitvoer:

```
Rechthoek: hoogte 5 cm - breedte 6 cm - oppervlakte 30 cm²
vierkant: Rechthoek: hoogte 4 cm - breedte 4 cm - oppervlakte 16 cm²
vierkant: Rechthoek: hoogte 2 cm - breedte 2 cm - oppervlakte 4 cm²
```

EXTRA: Breidt in de subklassen `Rechthoek` en `Vierkant` de methode `ToString()` uit met een tekening van de figuur (zie uitvoer).

Voorbeeld uitvoer:

```
Rechthoek: hoogte 5 cm - breedte 6 cm - oppervlakte 30 cm²
-----
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
-----
vierkant: Rechthoek: hoogte 4 cm - breedte 4 cm - oppervlakte 16 cm²
----
| |
| |
| |
| |
| |
----
vierkant: Rechthoek: hoogte 2 cm - breedte 2 cm - oppervlakte 4 cm²
--
--
```

Oefening 94: Boeken vergelijken

Neem het programma van **oefening 88 Boeken** van **hoofdstuk 15 Overerving en klasse-hiërarchie** er terug bij.

Implementeer de methoden `Equals()` en `GetHashCode()` in je klasse `Boek` via een override.

Twee boeken zijn dezelfde indien de volgende zaken gelijk zijn:

- ISBN,
- Titel,
- Auteur.

Vervang de methode `ToonInfo()` met een override van de methode `ToString()`.

Test deze methoden door enkele Boek-objecten in een `Console.WriteLine()` te plaatsen in de methode `Main()`.

Voorbeeld uitvoer (verder gaand op de vorige oefening):

```
The Shining - Stephen King (5848152) _ 50,00 EUR
Sleutelboek Computerhardware 3.0 - Marc Goris (9789403696737) _ 20,00 EUR _
3e graad Secundair onderwijs
The Black Swan - Nassim Nicholas Taleb (9780812973815) _ 40,99 EUR
Omnibus van Stephen King, Nassim Nicholas Taleb - (1924403303245) _ 45,50
EUR

True
False
```

Oefening 95: Restaurant

In deze oefening bootsen we de werking (bediening) van een restaurant na.

Maak een klasse `Restaurant` en maak zo goed mogelijk gebruik van **polymorfisme** om de code zo efficiënt op te stellen (kopieer en plak de onderstaande klassen in je programma).

```
public class Kok
{
    public void LeesOrder()
    {
        Console.WriteLine("De kok leest de order om te weten welke
            gerechten moeten worden klaargemaakt.");
    }
    public void NeemIngrediënten()
    {
        Console.WriteLine("De kok neemt de benodigde ingrediënten.");
    }
    public void KookGerechten()
    {
        Console.WriteLine("De kok kookt de gerechten.");
    }
    public void VerwittigKelner()
    {
        Console.WriteLine("De kok drukt op de bel om aan te kondigen dat
            de order van een tafel klaar is.");
    }
}

public class Barman
{
    public void LeesOrder()
    {
        Console.WriteLine("De barman leest de order om te weten welke
            dranken moeten worden gemaakt.");
    }
}
```

```

    public void MaakDrankjes()
    {
        Console.WriteLine("De barman maakt de drankjes klaar.");
    }
    public void GlazenAfwassen()
    {
        Console.WriteLine("De barman wast de vuile glazen af.");
    }
}

public class Kelner
{
    public void NeemOrderOp()
    {
        Console.WriteLine("De kelner neemt de order van een tafel op.");
    }
    public void ServeerDrinken()
    {
        Console.WriteLine("De kelner serveert het drinken van een
        tafel.");
    }
    public void ServeerEten()
    {
        Console.WriteLine("De kelner serveert het eten van een tafel.");
    }
    public void Afrekenen()
    {
        Console.WriteLine("De kelner geeft de rekening aan een tafel en
        ontvangt de betaling.");
    }
    public void MaakTafelsSchoon()
    {
        Console.WriteLine("De kelner maakt de tafels schoon.");
    }
}

```

Voorbeeld uitvoer:

```

De kok leest de order om te weten welke gerechten moeten worden
klaargemaakt.
De kok neemt de benodigde ingrediënten.
De kok kookt de gerechten.
De kok drukt op de bel om aan te kondigen dat de order van een tafel klaar
is.
De barman leest de order om te weten welke dranken moeten worden gemaakt.
De barman maakt de drankjes klaar.
De barman wast de vuile glazen af.
De kelner neemt de order van een tafel op.
De kelner serveert het drinken van een tafel.
De kelner serveert het eten van een tafel.
De kelner geeft de rekening aan een tafel en ontvangt de betaling.

```

De kelner maakt de tafels schoon.

Oefening 96: Een eigen huis

Gebruik compositie én overerving om een klasse `Huis` te voorzien van enkele kamers, waarbij iedere kamer een klasse op zich is (denk aan `BadKamer`, `Salon`, ...). Alle kamers erven over van de klasse `Kamer`.

Voorzie voor iedere kamer de volgende properties:

- Naam van het type `string`.
- Oppervlakte (in m^2) van het type `double`.
- Prijs van het type `double`.

Standaard is de prijs van een kamer 400 euro, maar mogelijke subklassen zullen deze property soms overriden. De Prijs is een read-only property (en heeft ook geen achterliggende instantievariabele).

Maak minstens volgende klassen:

- `Badkamer`: kost 500 euro.
- `Gang`: kost 10 euro per vierkante meter dat deze kamer groot is.
- `Salon`: kost 300 euro indien er geen schouw aanwezig is, anders kost deze kamer 500 euro. De aanwezigheid van een schouw hou je bij in een aparte property van het type `bool`.

De klasse `Huis` heeft een lijst van kamers. De klasse heeft ook een methode `BerekenPrijs()` die de totale prijs van het huis berekent gebaseerd op de prijzen van iedere kamer in de lijst. Deze methode mag niet worden aangeroepen buiten deze klasse.

De totale prijs van een huis wordt getoond via de methode `ToString()`.

Test je klasse in de methode `Main()` door huizen te maken en er enkele kamers in te plaatsen. Bepaal zelf hoe je de kamers aan het huis toevoegt: via een methode, constructor, ...).

Voorbeeld uitvoer (`badkamer`, `slaapkamer`, `salon met schouw` en een gang van $20 m^2$):

Totale prijs van het huis: 1 600,00 euro

EXTRA 1: StackOverflow bericht

Ontwerp een klasse genaamd `Bericht` om een StackOverflow-post na te maken. Een post bevat de volgende attributen:

- stemwaarde van het type `int`.
- titel van het type `string`.
- beschrijving van het type `string`.
- De aanmaakdatum van het type `DateTime` met de datum en tijd waarop het is aangemaakt. Dit is altijd het moment waarop het programma wordt uitgevoerd.

Voorzie via eigenschappen de mogelijkheid om buiten de klasse de titel en beschrijving in te vullen.

Voorzie methoden om een post te kunnen Upvoten() of Downvoten(). Wanneer je een instantie van de klasse `Bericht` wilt uitvoeren met een `Console.WriteLine()`, moet alle info worden afgedrukt op het scherm door de methode `ToString()` te overriden.



Hou rekening met inkapseling:

Je mag nooit de mogelijkheid bieden om de stemwaarde van buitenaf in te stellen. Anders kan je (per ongeluk) de stemmen van een klasse veranderen naar een 0 of een willekeurig getal, waardoor er bugs ontstaan in het programma in onze programma's. De klasse moet altijd zijn staat beschermen en de implementatiedetails verbergen.

In de methode `Main()` maak je een post, upvote en downvote je deze enkele keren en geef je vervolgens de volledige StackOverflow-post met de huidige stemwaarde (het aantal votes) weer.

Voorbeeld uitvoer:

(bron: <https://stackoverflow.com/questions/78822948/how-to-convert-decimal-point-to-comma-c-sharp>):

```
^
2      How to convert decimal point to comma C#
v

I need to write to a CSV file a decimal value with comma instead of decimal
point dot like 12,3 11,9 instead of 12.3 or 11.9. I am reading data from DB
as 12.3 but need to write to the CSV file as 12,3.

I've tried this so far:

string s = "1.14535765";
var d = Convert.ToDecimal("1.2345", new CultureInfo("es-ES"));

But in the console I can see 12345 without any dot or comma. I would
appreciate any suggestion here. Thanks in advance.

5/08/2024 14:51
```

EXTRA 2: Ontwerp een stack



Even herhalen: een Stack is een datastructuur voor het opslaan van een lijst met elementen op een LIFO (last in, first out) manier. Je hebt reeds leren werken met de klasse `Stack<>` van de namespace `System.Collections.Generic`.

In deze oefening ga je deze klasse namaken in jouw eigen namespace genaamd `MyStack`. Het doel van deze oefening is jou meer vertrouwd te maken met de klasse `System.Object`. Bovendien biedt dit zo meer inzicht in de achterliggende werking van een stack.

Ontwerp een klasse genaamd `Stack` met drie methoden:

- `void Push(object obj)`
- `object Pop()`
- `void Clear()`

De methode `Push()` slaat het gegeven object bovenop de stack op. We gebruiken hier het type `Object` zodat we elk object in de stack kunnen opslaan. Aangezien de klasse `Object` de basis is van alle klassen in het .NET Framework, kan elk type automatisch naar `Object` worden gepromoveerd. Zorg er ook voor dat je rekening houdt met het scenario dat `null` kan worden doorgegeven aan deze methode. Aangezien er geen `null`-referenties in de stack mogen worden opgeslagen, moet je een `InvalidOperationException` werpen als `null` wordt doorgegeven aan deze methode.

De methode `Pop()` verwijdert het object bovenop de stack en retourneert het. Zorg ervoor dat je rekening houdt met het scenario dat we deze methode aanroepen op een lege stack. In dit geval moet deze methode een `InvalidOperationException` werpen.



Zorg ervoor dat jouw klassen altijd in een geldige staat zijn en correct gebruikt worden. Wanneer ze verkeerd gebruikt worden, moeten ze uitzonderingen werpen. Door na te denken over al deze mogelijkheden zal je code robuuster zijn en minder bugs bevatten.

De methode `Clear()` verwijdert alle objecten uit de stack.

Test deze klasse in de methode `Main()` met de onderstaande code:

```
var stack = new Stack();
stack.Push(1);
stack.Push(2);
stack.Push(3);
Console.WriteLine(stack.Pop());
Console.WriteLine(stack.Pop());
Console.WriteLine(stack.Pop());
```

Voorbeeld uitvoer:

```
3
2
1
```