# Inleiding

In dit document vind je één of meerdere challenges die passen bij het onderwerp **Abstracte klassen en interfaces**. Je kunt de challenges gebruiken zoals gegeven, of gebruiken als voorbeeld en inspiratie voor challenges die je zelf opstelt.

## Challenge 1

Zie de challenges van module Inheritance. Je hebt daarin een dierenasiel opdracht gemaakt met een administratie van dieren. Deze challenge is een uitbreiding daarop (één programma/ontwerp van maken).

**Uitbreiding 1**

Neem het klassediagram van de dierenasiel opgave (die met de ‘Administration’ er in, dus uitkomst van deel 2 van de Animal Shelter challenge).

Geef in het klassediagram aan welke klassen, methoden en properties abstract kunnen zijn.

Teken een nieuw klassediagram waarin de abstracte klassen, methoden en properties verwerkt zijn en pas de code aan zodat deze klopt met het gemaakt klassediagram.

**Uitbreiding 2**

Het asiel heeft geld nodig om alle dieren te voeren en gaat daarom dieren in het asiel verkopen ipv weggeven.

Maak een interface ‘ISellable’ aan met een ‘Price’ property (decimal), en gebruik deze interface om ervoor te zorgen dat een dier een ‘Price’ property moet hebben. Pas ook het klassediagram aan.

Voor de code: de prijs van een hond moet anders berekend worden dan de prijs van een kat:

* Voor honden is de prijs afhankelijk van hun chipnummer. Is het chipnummer kleiner dan 50.000, dan is de prijs 200,00 euro en anders is de prijs 350,00 euro.
* Voor katten is de prijs afhankelijk van de extra informatie (bad habits eigenschap), die bijgehouden wordt. In het algemeen geldt: hoe meer extra informatie, hoe meer negatieve opmerkingen over die kat en dus hoe lager de prijs. De prijs van een kat is gelijk aan "60,00 euro min de lengte van de string met de extra informatie over die kat". Mocht dit minder dan 20,00 euro zijn, dan is de prijs 20,00 euro.

**Uitbreiding 3**

Gebruik de IComparable interface om dieren op chipnummer te kunnen sorteren.

In de user interface moet deze sortering terug te zien zijn in de lijsten met gereserveerde en niet gereserveerde dieren.

**Uitbreiding 4**

Maak Unit Tests voor de klassen Administration, Cat en Dog.

De criteria voor de beoordeling zijn:

* Je heb de feedback op je vorige uitwerking verwerkt.
* Je hebt het klassediagram correct uitgebreid met de gevraagde interface(s) en abstracte klassen, methoden, properties.
* De klassen, attributen, operaties en associaties zijn op correcte wijze vertaald naar C# code.
* De gevraagde functionele eisen en testen zijn geïmplementeerd.
* Je kan het ‘hoe en waarom’ van de gemaakte code verdedigen.
* Het programma vertoont geen bugs (blijft hangen, crashed, …)
* De gemaakte code is netjes: goede encapsulatie (private/public), duidelijke variabele namen, netjes inspringen, geen dubbele code, zelf toegevoegde methoden hebben een duidelijk doel en een duidelijke naam.
* Juist gebruik van loops (for/while/foreach)
* Juist gebruik van keuze structuren (if/else/else if/switch).