

Caravan - integratie oefening

Voor deze opdracht krijgt u meerdere dagen de tijd. De oefening is opgedeeld in verschillende delen. Begin bij het eerste deel en werk de opgave lineair af.

Let zeker op:

- Onderhoudvriendelijkheid: hoe onderhoudbaar is mijn code?
- Leesbaarheid van de code: gebruik ik logische variabele namen? is de code goed begrijpbaar?
- Formateer je code correct.
- Gebruik geen onnodige witregels
- Herbruikbaarheid: hergebruik code waar mogelijk.
- Test je functionaliteit. Zo voorkom je fouten.
- Zorg zeker dat je code compileert en niet crasht bij uitvoering!

Deel 1

Interfaces

Maak 4 interfaces aan:



ITrekbaar

ITrekbaar stelt een object voor dat getrokken kan worden door een object dat de interface ITrekker implementeert.

ITrekker

ITrekker stelt een object voor dat een object dat ITrekbaar implementeert kan trekken. De property Aanhangwagen houdt het ITrekbaar object bij dat aangekoppeld is om te trekken. Met de methode KoppelAanhangwagen kan je een nieuw object aankoppelen om te trekken. De property MaximaalTrekGewicht geeft het maximale gewicht aan dat getrokken kan worden.

IVoertuig

Een IVoertuig heeft een MaximumSnelheid en een huidige snelheid(Snelheid). Er zijn ook 2 methoden voorzien om te versnellen en vertragen.

IBewoonbaar

IBewoonbaar is een eenheid waarin gewoond kan worden. Het heeft een bepaald aantal slaapkamers en een bewoonbaar oppervlakte.

Classes

Maak volgende 8 klassen aan. Elk van deze klassen moet 1 constructor hebben die het object volledig kan initialiseren. Overschrijf ook telkens de ToString methode. Denk aan encapsulatie! Voorzie geen publieke set properties als dat niet expliciet gevraagd is. Voorzie wel steeds get Properties voor alle fields.

Aanhangwagen

De klasse Aanhangwagen implementeert de interface ITrekbaar. Zorg ervoor dat het gewicht niet negatief kan zijn! Het aantal banden is minimaal 2.

Caravan

De klasse Caravan erft over van de klasse Aanhangwagen. Verder implementeert de klasse ook de interface IBewoonbaar. Zorg ervoor dat het bewoonbaar oppervlakte niet negatief kan zijn! Het aantal slaapkamers in een Caravan is altijd 1.

Auto

De Klasse Auto implementeert de interfaces IVoertuig en ITrekker. De auto kan versnellen en vertragen, maar zijn snelheid kan niet hoger zijn dan zijn MaximumSnelheid. De auto kan niet achteruit rijden. De snelheid kan dus niet negatief zijn.

Als de auto stilstaat is het mogelijk om een aanhangwagen(elk object dat ITrekbaar implementeert) aan te koppelen. Als de auto niet stilstaat en we proberen een aanhangwagen aan te koppelen, dan wordt er een exception opgegooid. Indien de wagen overladen is, dit betekent dat het gewicht van de aanhangwagen groter is dan de maximale trekkracht van de auto, dan kan hij niet vertrekken. Indien we dit toch proberen resulteert dit in het gooien van een exception.

Motorhome

De klasse Motorhome erft van de klasse Auto. Motorhome implementeert ook de interface IBewoonbaar. Het aantal slaapkamers in een motorhome is altijd 1. Het bewoonbare oppervlakte mag niet kleiner zijn dan 0 en kan enkel ingesteld worden bij het aanmaken van het object. Nadien kan het niet meer aangepast worden.

Boot

De klasse Boot implementeert de interface IVoertuig. Een boot kan niet achteruit varen dus de snelheid moet steeds groter zijn dan 0. Ook de maximum snelheid kan niet negatief zijn.

Woonboot

De klasse woonboot erft van boot en imlementeert de interface IBewoonbaar. Aantal slaapkamers van een woonboot kan maximaal 2 zijn. Het bewoonbaar oppervlakte moet groter zijn dan 0

Huis

De klasse Huis implementeert de interface IBewoonbaar. Het aantal slaapkamers kan elke positieve waarde aannemen. Het bewoonbare oppervlakte moet steeds positief zijn.

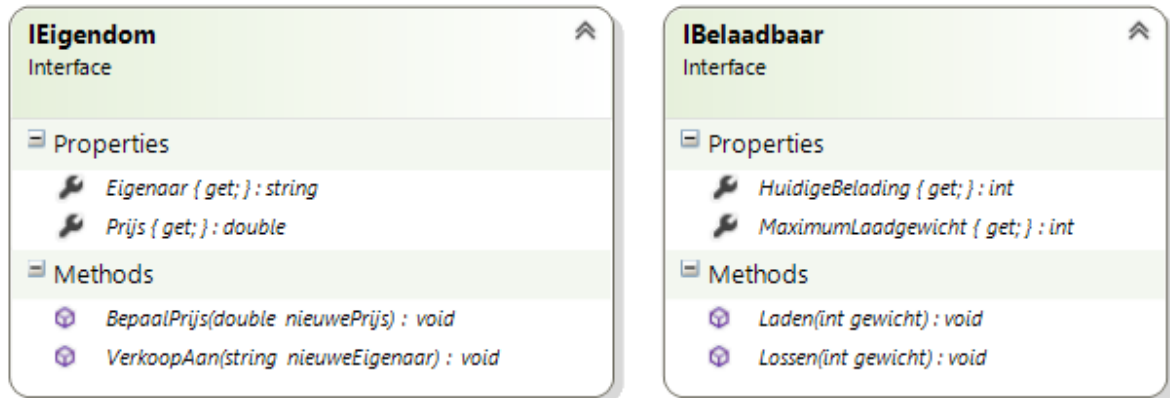
Kasteel

De klasse kasteel erft van Huis. Een kasteel heeft een minimum van 3 slaapkamers. Tip: gebruik polymorfisme om dit op te lossen.

Deel 2

Interfaces

Maak 2 nieuwe interfaces aan.



IEigendom

Een eigendom heeft een Eigenaar en een Prijs. Beide properties hebben enkel een public get. Een Eigendom heeft ook 2 methodes BepaalPrijs en VerkoopAan, respectievelijk om de prijs en de eigenaar te veranderen.

IBelaadbaar

De interface IBelaadbaar stelt een voorwerp voor dat we kunnen beladen. Het heeft een huidige belading en een maximaal laadgewicht. Beide properties zijn readonly. Er zijn ook 2 methodes voorzien om de lading te laden en te lossen.

Classes

Eigendom

Maak een abstracte klasse Eigendom die IEigendom implementeert. Zorg ervoor dat de prijs niet negatief kan zijn. Voorzie ook een constructor om Prijs en Eigenaar een initiele waarde te geven.

Laat de klassen Aanhangwagen, Huis, Auto en Boot overerven van Eigendom. **Merk Op:** Je moet voor deze 4 klassen en al hun child klassen ook constructors voorzien die Eigenaar en Prijs een initiele waarde kunnen geven! Pas ook de ToString methodes aan zodat eigenaar en prijs getoond worden.

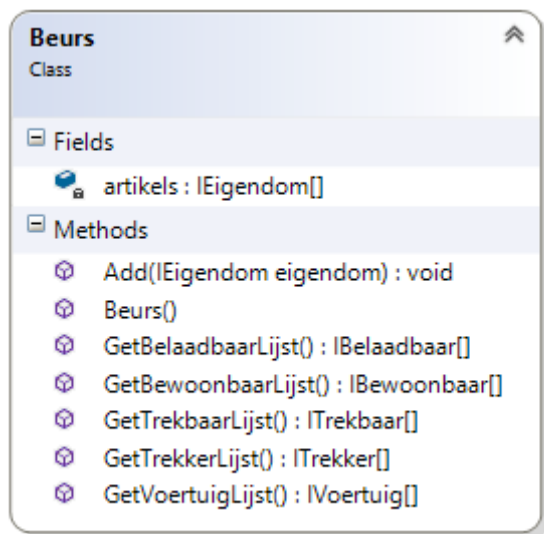
Transportboot

Eerst maken we de klasse Boot abstract. Daarna maken we een klasse Transportboot aan die overerft van de klasse Boot. Verder implementeert de klasse Transportboot ook de interface IBelaadbaar.

We kunnen de Transportboot laden en lossen. Maar als we de Transportboot overladen (*HuidigeBelading > MaximumLaadgewicht*) dan zinkt hij. Wanneer de Transportboot zinkt wordt zijn huidige snelheid op 0 gezet en wordt er een *InvalidOperationException* gegooid met als Message dat de boot gezonken is. Als we de Transportboot willen laden, lossen, versnellen of vertragen wanneer hij reeds gezonken is dan wordt er ook een *InvalidOperationException* gegooid met de boodschap dat de boot gezonken is. Voorzie ook een ToString methode voor de Transportboot.

Deel 3

Beurs



Maak een klasse Beurs aan. Een beurs heeft een lijst artikels van het type IEigendom. Het is mogelijk om verschillende eigendommen toe te voegen met de Add methode. Met de methodes Get...Lijst kan je alle artikels opvragen die aan een bepaalde voorwaarde voldoen. Bijvoorbeeld: De methode GetVoertuigLijst geeft een lijst terug van alle artikels die een voertuig zijn (IVoertuig implementeren). Zorg ervoor dat de lijsten die je terugkrijgt gesorteerd zijn op prijs. (Tip: IComparable)

Main programma

Maak in je hoofdprogramma een Beurs aan en voeg enkele artikels toe aan de artikellijst. Voeg minimaal 1 Aanhangwagen, 1 Caravan, 1 Huis, 1 Kasteel, 1 Auto, 1 Motorhome, 1 Transportboot en 1 Woonboot toe.

Geef de gebruiker de mogelijkheid om lijstjes op te vragen van de verschillende soorten artikels die op de beurs aanwezig zijn. Maak hiervoor gebruik van een menu waaruit de gebruiker kan kiezen.

Aanhangwagen implementeert IBelaadbaar

Laat de klasse Aanhangwagen de IBelaadbaar interface implementeren. We kunnen een aanhangwagen laden of lossen. Maar niet boven zijn maximale laadgewicht. Het maximale laadgewicht moet in de constructor kunnen worden meegegeven. Houd er rekening mee dat de lading mee het gewicht bepaald van de aanhangwagen en dat dit dus mee bepaald of een auto in staat is om de aanhangwagen te trekken of niet.

KoppelException

Maak een custom exception met de naam KoppelException. Geef deze exception een property Snelheid en een constructor met als parameters een message en een snelheid. De bedoeling is dat we deze KoppelException gaan opgooien wanneer we in de klasse Auto een ITrekbaar object (bvb een aanhangwagen) willen aankoppelen terwijl de auto nog rijdt. We geven dan een message mee om de gebruiker duidelijk te maken wat het probleem is en we geven in de property Snelheid ook de huidige snelheid van de wagen mee.

OverBeladenException

Maak een custom exception met de naam OverBeladenException. Geef deze exception 2 properties: MaximaleTrekkracht en TeTrekkenGewicht. Maak ook een constructor om de message en deze 2 nieuwe properties in te vullen. De bedoeling is dat we deze exception gaan gebruiken in onze klasse Auto wanneer we proberen te versnellen terwijl de auto overladen is. Geef de trekkracht van de Auto mee en het gewicht van de aanhangwagen.