

Opgave - Implementeer de volgende business logica.

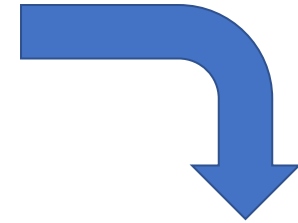
We begeven ons in de wereld van de scheepvaart. Het beheer van **schepen** gebeurt door **rederijen**, waarbij elke rederij over verschillende **vloten** beschikt en elke vloot bestaat uit een aantal schepen. Er zijn verschillende types van schepen, we onderscheiden :

- **Containerschip** : lengte, breedte, tonnage, naam, aantalcontainers, cargowaarde
- **RoRoschip** : lengte, breedte, tonnage, naam, aantal autos, aantal trucks, cargowaarde
- **Cruiseschip** : lengte, breedte, tonnage, naam, aantal passagiers, traject
- **Veerboot** : lengte, breedte, tonnage, naam, aantal passagiers, traject
- **Olietanker** : lengte, breedte, tonnage, naam, cargowaarde, volume (liters), lading (olie, benzeen, diesel of nafta)
- **Gastanker** : lengte, breedte, tonnage, naam, cargowaarde, volume (liters), lading (LPG, LNG of amoniak)
- **Sleepboot** : lengte, breedte, tonnage, naam

Cargowaarde : is de waarde van de vracht uitgedrukt in euro.

Tonnage : is het volume van het schip

Traject : bij een veerboot is dit een vast traject tussen 2 havens, bij een cruiseschip is dit een variabele lijst van **havens** die worden aangedaan. Bij de route van een cruiseschip is de volgorde van de havens van belang. Havens kunnen eenvoudig worden voorgesteld met hun naam.



Klassen :

- Schip
- Rederij
- Vloot
- Containerschip
- RoRoschip
- Cruiseschip
- Veerboot
- Olietanker
- Gastanker
- Sleepboot
- Traject
- Haven

We begeven ons in de wereld van de scheepvaart. Het beheer van **schepen** gebeurt door **rederijen**, waarbij elke rederij over verschillende **vloten** beschikt en elke vloot bestaat uit een aantal schepen. Er zijn verschillende types van schepen, we onderscheiden :

- **Containerschip** : lengte, breedte, tonnage, naam, aantalcontainers, cargowaarde
- **RoRoschip** : lengte, breedte, tonnage, naam, aantal autos, aantal trucks, cargowaarde
- **Cruiseschip** : lengte, breedte, tonnage, naam, aantal passagiers, traject
- **Veerboot** : lengte, breedte, tonnage, naam, aantal passagiers, traject
- **Olietanker** : lengte, breedte, tonnage, naam, cargowaarde, volume (liters), lading (olie, benzeen, diesel of nafta)
- **Gastanker** : lengte, breedte, tonnage, naam, cargowaarde, volume (liters), lading (LPG, LNG of amoniak)
- **Sleepboot** : lengte, breedte, tonnage, naam

Cargowaarde : is de waarde van de vracht uitgedrukt in euro.

Tonnage : is het volume van het schip

Traject : bij een veerboot is dit een vast traject tussen 2 havens, bij een cruiseschip is dit een variabele lijst van **havens** die worden aangedaan. Bij de route van een cruiseschip is de volgorde van de havens van belang. Havens kunnen eenvoudig worden voorgesteld met hun naam.

Een vloot heeft een naam (vb Noordzeevloot) en beschikt over een aantal schepen. Het moet

attributen

Schip (naam, lengte, breedte, tonnage)

Vrachtschip (cargowaarde)

Tanker (volume)

Gastanker

Olietanker

Containerschip

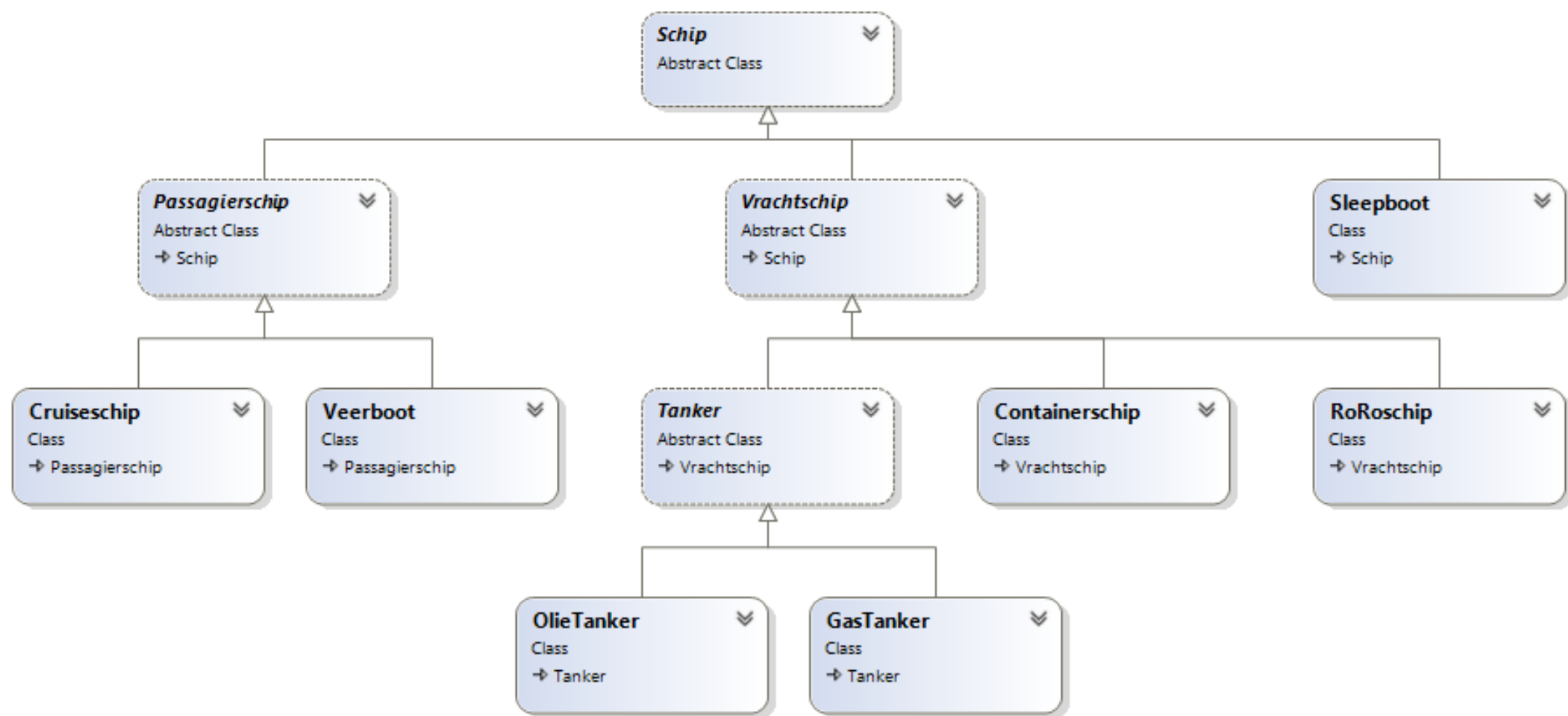
RoRoschip

Sleepboot

Passagierschip
(passagiers, traject)

Cruiseschip

Veerboot



We begeven ons in de wereld van de scheepvaart. Het beheer van **schepen** gebeurt door **rederijen**, waarbij elke rederij over verschillende **vloten** beschikt en elke vloot bestaat uit een aantal schepen. Er zijn verschillende types van schepen, we onderscheiden :

Traject : bij een veerboot is dit een vast traject tussen 2 havens, bij een cruiseschip is dit een variabele lijst van **havens** die worden aangedaan. Bij de route van een cruiseschip is de volgorde van de havens van belang. **Havens kunnen eenvoudig worden voorgesteld met hun naam**.

Een vloot heeft een naam (vb Noordzeevloot) en beschikt over een aantal schepen. Het moet mogelijk zijn om zowel schepen toe te voegen aan een vloot als om schepen te verwijderen. Een schip mag natuurlijk maar éénmaal in een vloot voorkomen. Het moet ook mogelijk zijn om zeer snel een schip op te zoeken op basis van de naam van het schip. En het moet ook mogelijk zijn om een overzicht te geven van de schepen die in een vloot zijn opgenomen.

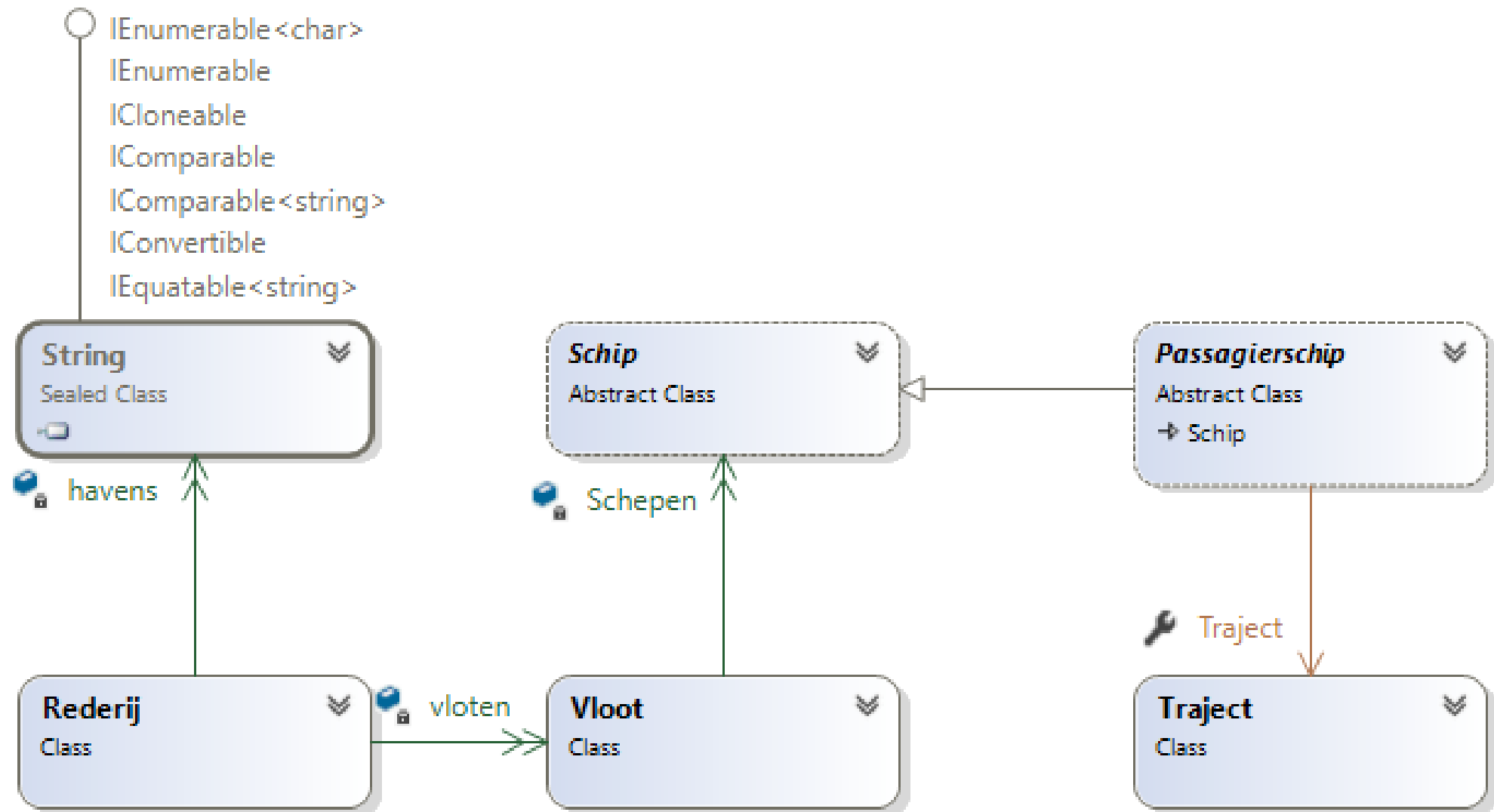
De rederij beschikt over verschillende vloten (vb Noord Atlantische vloot, Noordzeevloot, Caraïbische vloot ...). Het moet mogelijk zijn om een vloot toe te voegen of te verwijderen. Eénzelfde vloot komt natuurlijk maar éénmaal voor. Een vloot moet ook snel opzoekbaar zijn op basis van de naam. Een rederij is actief in verschillende havens, het moet dus mogelijk zijn om havens toe te voegen aan een rederij (en ook om havens te verwijderen). Het moet ook mogelijk zijn om een overzicht te geven van de havens die bij een rederij behoren in alfabetische volgorde. Havens worden eenvoudig voorgesteld door enkel een naam (vb Antwerpen, Rotterdam, ...).

relaties

List

Dictionary

SortedSet



Een vloot heeft een naam (vb Noordzeevloot) en beschikt over een aantal schepen. Het moet mogelijk zijn om zowel schepen toe te voegen aan een vloot als om schepen te verwijderen. Een schip mag natuurlijk maar éénmaal in een vloot voorkomen. Het moet ook mogelijk zijn om zeer snel een schip op te zoeken op basis van de naam van het schip. En het moet ook mogelijk zijn om een overzicht te geven van de schepen die in een vloot zijn opgenomen.

De rederij beschikt over verschillende vloten (vb Noord Atlantische vloot, Noordzeevloot, Caraïbische vloot ...). Het moet mogelijk zijn om een vloot toe te voegen of te verwijderen. Eénzelfde vloot komt natuurlijk maar éénmaal voor. Een vloot moet ook snel opzoekbaar zijn op basis van de naam. Een rederij is actief in verschillende havens, het moet dus mogelijk zijn om havens toe te voegen aan een rederij (en ook om havens te verwijderen). Het moet ook mogelijk zijn om een overzicht te geven van de havens die bij een rederij behoren in alfabetische volgorde. Havens worden eenvoudig voorgesteld door enkel een naam (vb Antwerpen, Rotterdam, ...).

Het moet ook mogelijk zijn om een schip in een andere vloot te plaatsen (vb plaats schip "De Plank" in vloot "Stille oceaan").

Verder moet het mogelijk zijn om een overzicht te geven van :

- De totale cargowaarde van de schepen die tot een rederij behoren.
- Het totaal aantal passagiers.
- De tonnage per vloot op te lijsten (van groot naar klein).
- Het totaal aantal volume die de tankers kunnen vervoeren.
- De beschikbare sleepboten.

Een rederij moet ook in staat zijn om info over een bepaald schip weer te geven (op basis van de naam van het schip).

Methodes

Voorbeeld 1

Voorbeeld 5

Voorbeeld 2

Voorbeeld 3

Voorbeeld 4

```
private Dictionary<string, Schip> Schepen = new Dictionary<string, Schip>();
```

2 references

```
public Vloot(string naam)...
```

10 references

```
public void voegSchipToe(Schip schip)
{
    if (!Schepen.ContainsKey(schip.Naam))
    {
        Schepen.Add(schip.Naam, schip);
        schip.Vloot = this;
    }
}
```

1 reference

```
public void verwijderSchip(Schip schip)
{
    if (Schepen.ContainsKey(schip.Naam))
    {
        Schepen.Remove(schip.Naam);
        schip.Vloot = null;
    }
}
```

2 references

Schepen toevoegen en verwijderen :

- ContainsKey om te vermijden dat we exceptions gaan krijgen
- Dictionary zorgt dat er geen dubbels zijn
- Private dictionary om te verplichten via methods te werken

Voorbeeld 1

Een vloot heeft een naam (vb Noordzeevloot) en beschikt over een aantal schepen. Het moet mogelijk zijn om zowel schepen toe te voegen aan een vloot als om schepen te verwijderen. Een schip mag natuurlijk maar éénmaal in een vloot voorkomen. Het moet ook mogelijk zijn om zeer snel een schip op te zoeken op basis van de naam van het schip. En het moet ook mogelijk zijn om een overzicht te geven van de schepen die in een vloot zijn opgenomen.

De rederij beschikt over verschillende vloten (vb Noord Atlantische vloot, Noordzeevloot, Caraïbische vloot ...). Het moet mogelijk zijn om een vloot toe te voegen of te verwijderen. Eénzelfde vloot komt natuurlijk maar éénmaal voor. Een vloot moet ook snel opzoekbaar zijn op basis van de naam. Een rederij is actief in verschillende havens, het moet dus mogelijk zijn om havens toe te voegen aan een rederij (en ook om havens te verwijderen). Het moet ook mogelijk zijn om een overzicht te geven van de havens die bij een rederij behoren in alfabetische volgorde. Havens worden eenvoudig voorgesteld door enkel een naam (vb Antwerpen, Rotterdam, ...).

Het moet ook mogelijk zijn om een schip in een andere vloot te plaatsen (vb plaats schip "De Plank" in vloot "Stille oceaan").

Verder moet het mogelijk zijn om een overzicht te geven van :

- De totale cargowaarde van de schepen die tot een rederij behoren.
- Het totaal aantal passagiers.
- De tonnage per vloot op te lijsten (van groot naar klein).
- Het totaal aantal volume die de tankers kunnen vervoeren.
- De beschikbare sleepboten.

Een rederij moet ook in staat zijn om info over een bepaald schip weer te geven (op basis van de naam van het schip).

Methodes

Voorbeeld 1

Voorbeeld 5

Voorbeeld 2

Voorbeeld 3

Voorbeeld 4

Rederij



Rederij

```
public int passagiers()
{
    int p = 0;
    foreach (Vloot v in vloten.Values)
    {
        p += v.passagiers();
    }
    return p;
}
```

```
public int passagiers()
{
    int p = 0;
    foreach (Vloot v in vloten.Values)
    {
        foreach (Schip s in v.geefSchepen())
        {
            if (s is Passagierschip) p += ((Passagierschip)s).AantalPassagiers;
        }
    }
    return p;
}
```

Vloot

```
public int passagiers()
{
    int p = 0;
    foreach (Schip s in Schepen.Values)
    {
        if (s is Passagierschip) p += ((Passagierschip)s).AantalPassagiers;
    }
    return p;
}
```

Rederij hoeft Passagierschip niet te kennen -> is een extra afhankelijkheid

Voorbeeld 2

Een vloot heeft een naam (vb Noordzeevloot) en beschikt over een aantal schepen. Het moet mogelijk zijn om zowel schepen toe te voegen aan een vloot als om schepen te verwijderen. Een schip mag natuurlijk maar éénmaal in een vloot voorkomen. Het moet ook mogelijk zijn om zeer snel een schip op te zoeken op basis van de naam van het schip. En het moet ook mogelijk zijn om een overzicht te geven van de schepen die in een vloot zijn opgenomen.

De rederij beschikt over verschillende vloten (vb Noord Atlantische vloot, Noordzeevloot, Caraïbische vloot ...). Het moet mogelijk zijn om een vloot toe te voegen of te verwijderen. Eénzelfde vloot komt natuurlijk maar éénmaal voor. Een vloot moet ook snel opzoekbaar zijn op basis van de naam. Een rederij is actief in verschillende havens, het moet dus mogelijk zijn om havens toe te voegen aan een rederij (en ook om havens te verwijderen). Het moet ook mogelijk zijn om een overzicht te geven van de havens die bij een rederij behoren in alfabetische volgorde. Havens worden eenvoudig voorgesteld door enkel een naam (vb Antwerpen, Rotterdam, ...).

Het moet ook mogelijk zijn om een schip in een andere vloot te plaatsen (vb plaats schip "De Plank" in vloot "Stille oceaan").

Verder moet het mogelijk zijn om een overzicht te geven van :

- De totale cargowaarde van de schepen die tot een rederij behoren.
- Het totaal aantal passagiers.
- De tonnage per vloot op te lijsten (van groot naar klein).
- Het totaal aantal volume die de tankers kunnen vervoeren.
- De beschikbare sleepboten.

Een rederij moet ook in staat zijn om info over een bepaald schip weer te geven (op basis van de naam van het schip).

Methodes

Voorbeeld 1

Voorbeeld 5

Voorbeeld 2

Voorbeeld 3

Voorbeeld 4

Voorbeeld 3

```
public SortedDictionary<int, List<Vloot>> tonnagePerVloot()
{
    //opm List<Vloot> want vloten kunnen zelfde tonnage hebben
    SortedDictionary<int, List<Vloot>> tpv = new SortedDictionary<int, List<Vloot>>();
    foreach(Vloot v in vloten.Values)
    {
        int t = v.tonnage();
        if (tpv.ContainsKey(t)) tpv[t].Add(v);
        else { tpv.Add(t, new List<Vloot>() { v }); }
    }
    return tpv;
}
```

Een vloot heeft een naam (vb Noordzeevloot) en beschikt over een aantal schepen. Het moet mogelijk zijn om zowel schepen toe te voegen aan een vloot als om schepen te verwijderen. Een schip mag natuurlijk maar éénmaal in een vloot voorkomen. Het moet ook mogelijk zijn om zeer snel een schip op te zoeken op basis van de naam van het schip. En het moet ook mogelijk zijn om een overzicht te geven van de schepen die in een vloot zijn opgenomen.

De rederij beschikt over verschillende vloten (vb Noord Atlantische vloot, Noordzeevloot, Caraïbische vloot ...). Het moet mogelijk zijn om een vloot toe te voegen of te verwijderen. Eénzelfde vloot komt natuurlijk maar éénmaal voor. Een vloot moet ook snel opzoekbaar zijn op basis van de naam. Een rederij is actief in verschillende havens, het moet dus mogelijk zijn om havens toe te voegen aan een rederij (en ook om havens te verwijderen). Het moet ook mogelijk zijn om een overzicht te geven van de havens die bij een rederij behoren in alfabetische volgorde. Havens worden eenvoudig voorgesteld door enkel een naam (vb Antwerpen, Rotterdam, ...).

Het moet ook mogelijk zijn om een schip in een andere vloot te plaatsen (vb plaats schip "De Plank" in vloot "Stille oceaan").

Verder moet het mogelijk zijn om een overzicht te geven van :

- De totale cargowaarde van de schepen die tot een rederij behoren.
- Het totaal aantal passagiers.
- De tonnage per vloot op te lijsten (van groot naar klein).
- Het totaal aantal volume die de tankers kunnen vervoeren.
- De beschikbare sleepboten.

Een rederij moet ook in staat zijn om info over een bepaald schip weer te geven (op basis van de naam van het schip).

Methodes

Voorbeeld 1

Voorbeeld 5

Voorbeeld 2

Voorbeeld 3

Voorbeeld 4

Voorbeeld 4

```
public abstract class Schip
{
    3 references
    public Schip(int lengte, int breedte, string naam, int tonnage) {...}

    2 references
    public int Lengte { get; set; }
    2 references
    public int Breedte { get; set; }
    6 references
    public string Naam { get; set; }
    3 references
    public int Tonnage { get; set; }
    5 references
    public Vloot Vloot { get; set; }
    15 references
    public override string ToString()
    {
        return $"(Schip {Naam},{Lengte},{Breedte},{Tonnage},{Vloot.Naam},{this.GetType()})";
    }
}
```

Vloot

```
public Schip zoekSchip(string naam)
{
    if (Schepen.ContainsKey(naam))
        return Schepen[naam];
    else
        return null;
}
```

Rederij

```
public Schip zoekSchip(string schipnaam)
{
    foreach (Vloot v in vloten.Values)
    {
        Schip s;
        if ((s = v.zoekSchip(schipnaam)) != null)
            return s;
    }
    return null;
}
```


Een vloot heeft een naam (vb Noordzeevloot) en beschikt over een aantal schepen. Het moet mogelijk zijn om zowel schepen toe te voegen aan een vloot als om schepen te verwijderen. Een schip mag natuurlijk maar éénmaal in een vloot voorkomen. Het moet ook mogelijk zijn om zeer snel een schip op te zoeken op basis van de naam van het schip. En het moet ook mogelijk zijn om een overzicht te geven van de schepen die in een vloot zijn opgenomen.

De rederij beschikt over verschillende vloten (vb Noord Atlantische vloot, Noordzeevloot, Caraïbische vloot ...). Het moet mogelijk zijn om een vloot toe te voegen of te verwijderen. Eénzelfde vloot komt natuurlijk maar éénmaal voor. Een vloot moet ook snel opzoekbaar zijn op basis van de naam. Een rederij is actief in verschillende havens, het moet dus mogelijk zijn om havens toe te voegen aan een rederij (en ook om havens te verwijderen). Het moet ook mogelijk zijn om een overzicht te geven van de havens die bij een rederij behoren in alfabetische volgorde. Havens worden eenvoudig voorgesteld door enkel een naam (vb Antwerpen, Rotterdam, ...).

Het moet ook mogelijk zijn om een schip in een andere vloot te plaatsen (vb plaats schip "De Plank" in vloot "Stille oceaan").

Verder moet het mogelijk zijn om een overzicht te geven van :

- De totale cargowaarde van de schepen die tot een rederij behoren.
- Het totaal aantal passagiers.
- De tonnage per vloot op te lijsten (van groot naar klein).
- Het totaal aantal volume die de tankers kunnen vervoeren.
- De beschikbare sleepboten.

Een rederij moet ook in staat zijn om info over een bepaald schip weer te geven (op basis van de naam van het schip).

Methodes

Voorbeeld 1

Voorbeeld 5

Voorbeeld 2

Voorbeeld 3

Voorbeeld 4

Voorbeeld 4

Rederij

```
public void plaatsSchipInAndereVloot(string schipnaam, string vlootnaam)
{
    Schip s = zoekSchip(schipnaam);
    if (s != null)
    {
        vloten[s.Vloot.Naam].verwijderSchip(s);
        vloten[vlootnaam].voegSchipToe(s);
        s.Vloot = vloten[vlootnaam];
    }
}
```