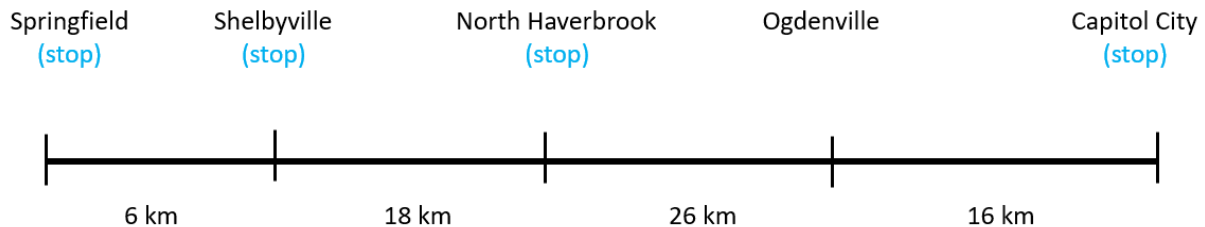


Route requirements

Een route bestaat uit een aantal haltes/locaties. Een locatie kan een stopplaats zijn of een doorrij halte. De afstand tussen twee locaties wordt meegegeven.



Implementeer de klasse XRoute met de volgende interface (IRoute).

```
public interface IRoute
{
    void AddLocation(string location, double distance, bool isStop);
    double GetDistance();
    double GetDistance(string startLocation, string endLocation);
    bool HasLocation(string location);
    bool HasStop(string location);
    void InsertLocation(string location, double distance, string fromLocation, bool isStop);
    void RemoveLocation(string location);
    void SetDistance(double distance, string location1, string location2);
    (string start, List<(double distance, string location)>) ShowFullRoute();
    (string start, List<(double distance, string location)>) ShowFullRoute(string startLocation, string endLocation);
    List<string> ShowLocations();
    (string start, List<(double distance, string location)>) ShowRoute();
    (string start, List<(double distance, string location)>) ShowRoute(string startLocation, string endLocation);
    List<string> ShowStops();
    void UpdateLocation(string location, string newName, bool isStop);
}
```

Info :

- Distance mag niet kleiner zijn dan een opgegeven waarde.
- Location mag niet leeg zijn en moet beginnen met een hoofdletter.
- Route instances mogen enkel worden aangemaakt via de klasse RouteFactory.
- RemoveLocation : vb RemoveLocation(B)
 - A – (5 km) – B – (10 km) - C
 - Nieuwe situatie : A – (15 km) – C
- InsertLocation : vb
 - A – (5 km) – B – (10 km) - C
 - InsertLocation(Q,7,B,true)
 - Resultaat : A – (5 km) – B – (7 km) - Q - (3 km) - C

De klasse RouteFactory implementeert de volgende methoden:

```
public class RouteFactory
{
    public XRoute BuildRouteFromFile(string fileName) { }
    public XRoute BuildRoute(List<string> locations, List<bool> stops, List<double> distances) { }
    public XRoute ReverseRoute(XRoute route) { }
}
```

Er zijn twee verschillende manieren waarop het tekstbestand kan zijn opgebouwd. De eerste ziet er als volgt uit :

```
The Shire,0,true
Bree,25,true
Rivendel,60,true
Edoras,33,true
Helm's Deep,5,false
Isengard,30,true
Minas Tirith,28,true
Minas Morgul,8,false
Mount Doom,38,true
```

Naam van de locatie, de afstand tot de locatie en 'true' als het een stop is, 'false' als het geen stopplaats is.

De tweede manier is de volgende :

```
The Shire(stop),Bree(stop),25
Bree(stop),Rivendel(stop),60
Rivendel(stop),Edoras(stop),33
Edoras(stop),Helm's Deep(transit),5
Helm's Deep(transit),Isengard(stop),30
Isengard(stop),Minas Tirith(stop),28
Minas Tirith(stop),Minas Morgul(transit),8
Minas Morgul(transit),Mount Doom(stop),38
```

De afstand tussen twee locaties wordt telkens meegegeven, voor elke locatie wordt ook aangegeven of het een stopplaats is of een transit locatie. De volgorde van de locaties is ook de volgorde van de route.

Opdracht :

- Implementeer de klassen XRoute en RouteFactory in een class library.
- Voor zie de klasse Route van de nodige Unit Tests (xUnit).
- Als er iets fout gaat, moet er een RouteException worden gegooit.