

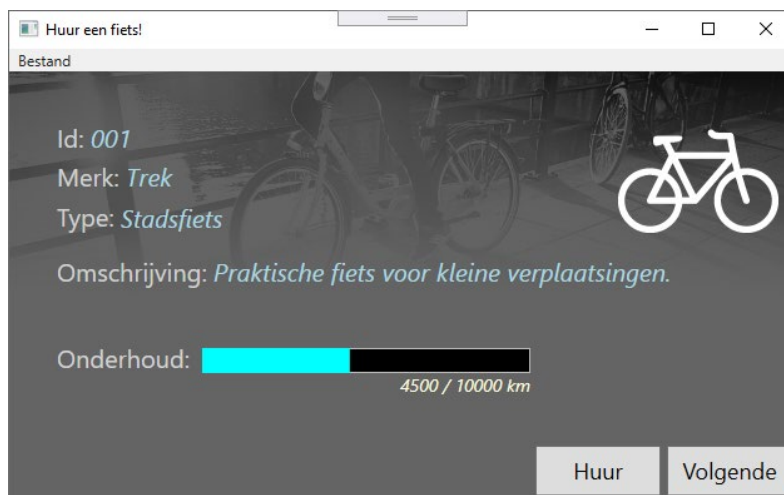
BikeRent

Fietsen is enorm populair geworden in coronatijden. Maar niet iedereen heeft een fiets. Daarom gaan we een applicatie bouwen om fietsen te verhuren! We beschrijven eerst de werking van de applicatie en daarna de implementatiedetails.

Werking

Opstarten

Bij het opstarten van de applicatie verschijnt de eerste fiets uit het assortiment.



Via de knop “Volgende” kan je naar de volgende fiets navigeren. Om de opgave beperkt te houden is er geen “Vorige”-knop voorzien.

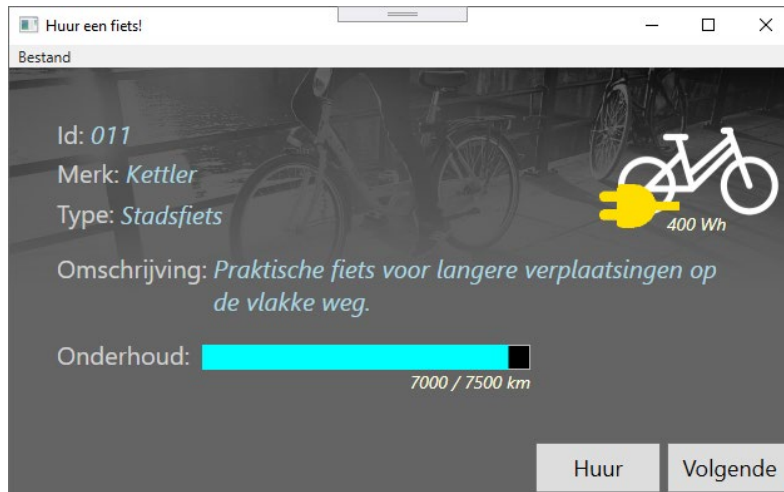


Volgende gegevens worden getoond:

- **Id** van de fiets : een getal dat steeds als 3 cijfers wordt getoond.
- **Merk, type en omschrijving**: dit zijn strings.
- Via een afbeelding zie je of het een mannen- of vrouwenfiets is.

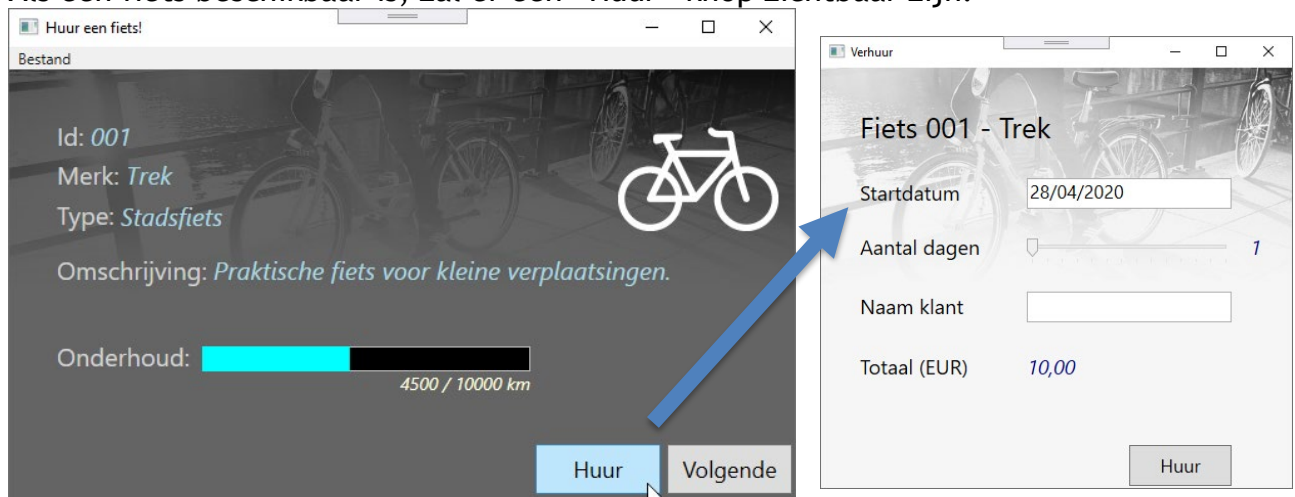
- Een voortgangsbalk geeft aan hoeveel kilometer er al is gereden met de fiets. Na een bepaald aantal kilometer kan de fiets niet meer verhuurd worden tot er een onderhoud is uitgevoerd. In bovenstaande afbeelding zijn er 1500 km met de fiets afgelegd en na 10000 km moet de fiets in onderhoud.
- Als een fiets verhuurd is, zie je dit in een tekstveld onderaan. Je kan die fiets dan niet meer huren. Als de klant de fiets terugbrengt moet er op de “Retour”-knop geklikt worden.

Er zijn ook elektrische fietsen in het assortiment. Dit zie je aan het gele logo en het batterijvermogen (400 Wh in het voorbeeld hieronder):



Een fiets huren

Als een fiets beschikbaar is, zal er een “Huur”-knop zichtbaar zijn.



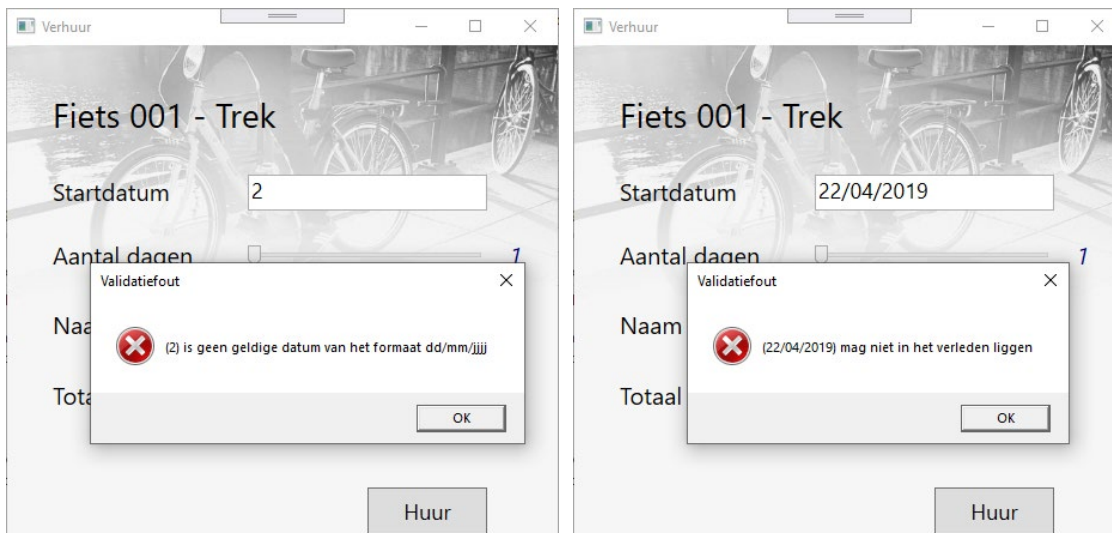
Door op de “Huur”-knop te klikken verschijnt een tweede venster (Verhuur). Het eerste venster is niet meer benaderbaar tot het Verhuurvenster weer is gesloten.

In dit verhuurvenster zie je volgende elementen:

- Het Id van de fiets en het merk
- De startdatum van een verhuuring

- Standaard staat dit op de “huidige dag” ingesteld. Deze opgave is opgesteld op 28/04, dus vandaar deze datum in de afbeelding hierboven
- Je kan de datum wijzigen, maar die moet steeds in de toekomst liggen
- Een slider om het aantal te huren dagen in te geven (varieert tussen 1 en 14)
- De naam van de huurder: voor de eenvoud hoeft je hiervoor geen validatie te voorzien, maar we gaan ervan uit dat het een verplicht veld is.

Als je fouten maakt bij het invullen, krijg je een berichtvenster met een gepaste melding. Het Verhuurvenster blijft geopend tot alle velden correct zijn ingevuld. Hieronder de twee mogelijke meldingen (andere moet je niet voorzien).

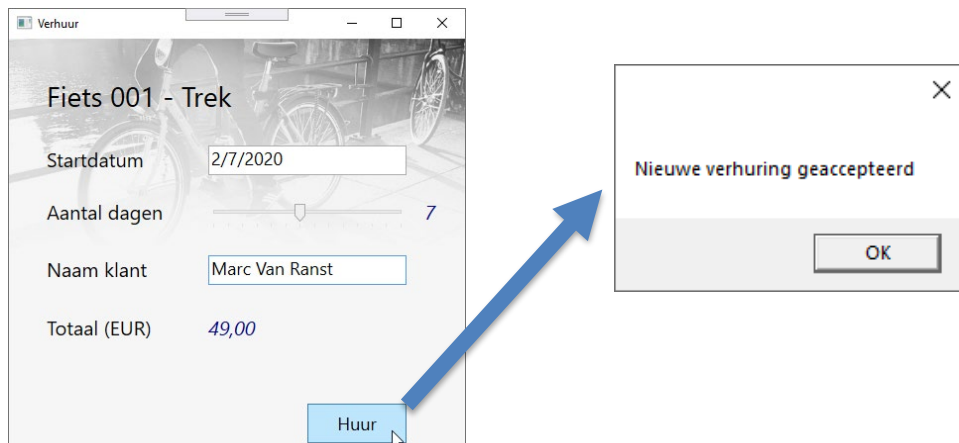


Onderstaand scherm is volledig correct ingevuld. Bemerkt dat het totaalbedrag berekend wordt en bijgewerkt zodra de gebruiker de slider aanpast.

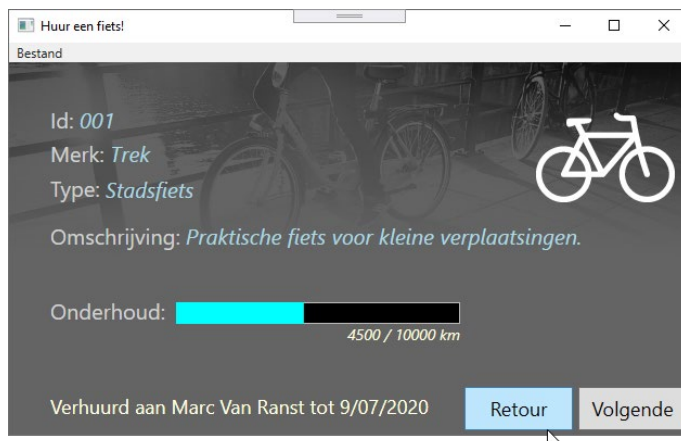
Voor het totaalbedrag krijgt de klant korting naarmate hij of zij langer huurt:

- Voor 1 dag betaal je de dagprijs (bijvoorbeeld: 10 EUR)
- Voor 2 dagen krijg je 20% korting op het totaal (bijvoorbeeld: 80% van 20 EUR = 16 EUR)
- Voor meer dan 2 dagen krijg je 30% korting.

Het totaalbedrag is een cijfer dat altijd tot 2 cijfers na de komma wordt getoond.

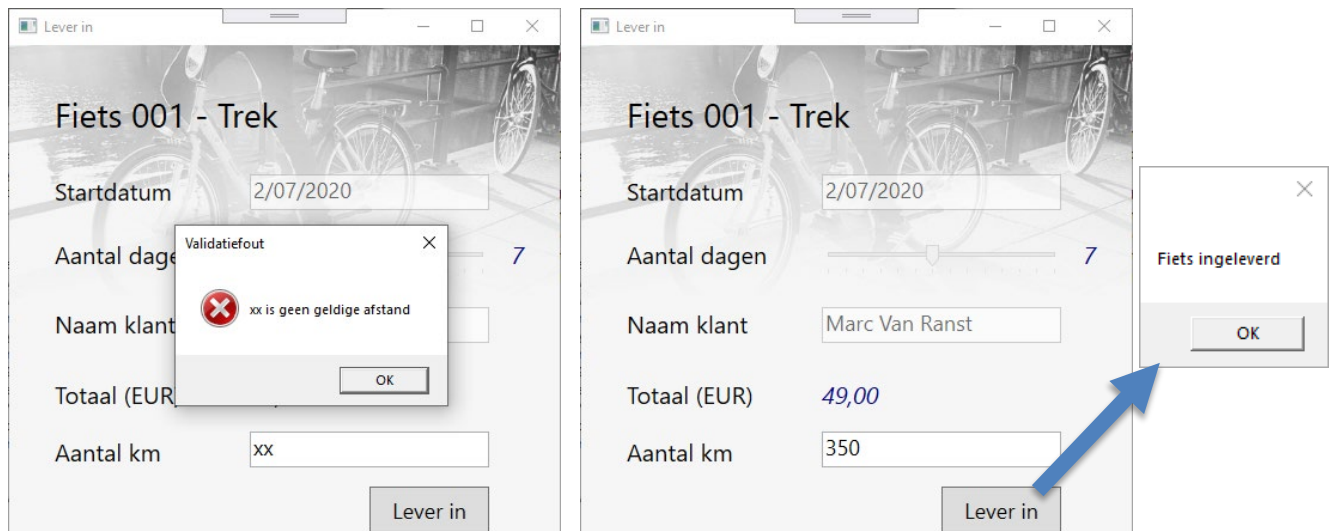


Na een druk op de “Huur”-knop verschijnt een berichtvenster dat de verhuur is geaccepteerd. Je keert dan terug naar het eerste scherm waarbij de informatie automatisch is bijgewerkt.

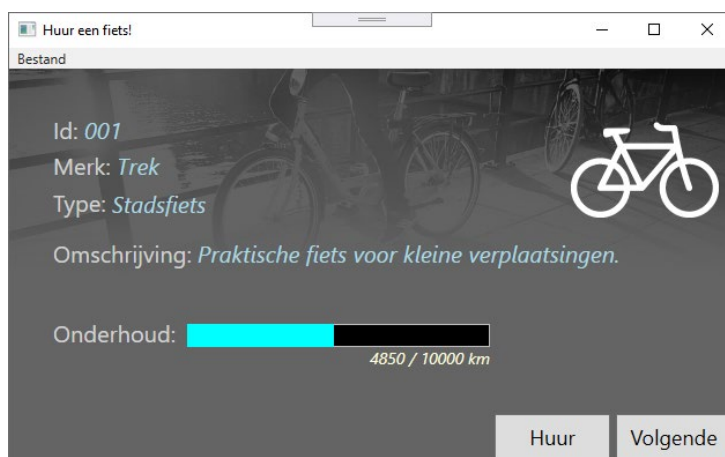


Je merkt nu dat onderaan het scherm de verhuurtermijn is aangegeven en dat er een “Retour”-knop is verschenen.

Zodra de huurder de fiets inlevert, zal er op de “Retour”-knop geklikt worden. Nu zal het “Lever in”-scherm verschijnen. Dit is hetzelfde scherm als bij het verhuren, maar nu heeft de gebruiker de mogelijkheid om het aantal afgelegde kilometers van de voorbije verhuurperiode in te geven. Dit veld moet een geldig getal bevatten. Alle andere velden bevatten de gegevens van de verhuur en kan je niet meer aanpassen.

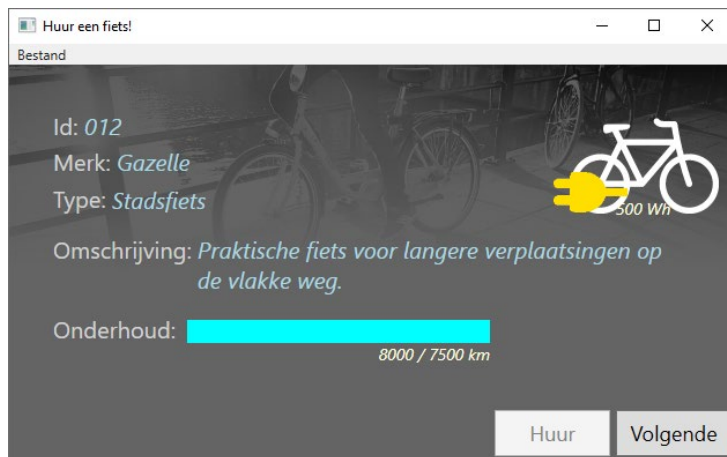


Na het klikken op de knop “Lever in”, ga je terug naar het vorige scherm (eerst verschijnt een berichtvenster). Je ziet nu dat het aantal afgelegde kilometers (en de voortgangsbalk) is aangepast. De fiets is nu ook opnieuw beschikbaar voor verhuur.



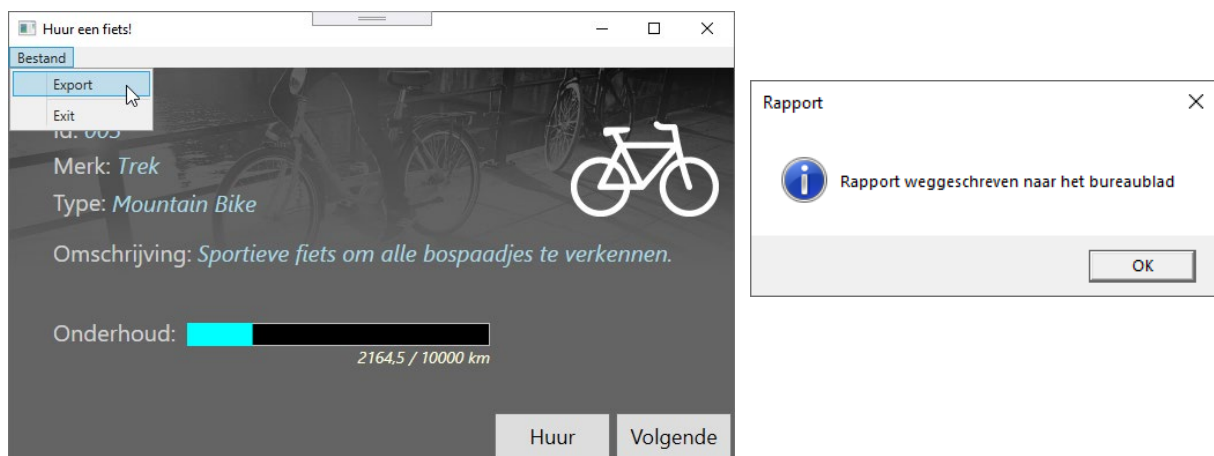
Fietsen moeten regelmatig onderhouden worden. Dit varieert afhankelijk van het soort fietsen. “Gewone” fietsen moeten om de 10000 kilometers onderhouden worden en elektrische fietsen om de 7500 kilometers.

Als een fiets boven het maximaal aantal kilometers is gekomen, moet de “Huur”-knop gedeactiveerd worden. Pas na onderhoud zal de fiets weer beschikbaar zijn. Het onderhoud valt buiten de scope van de opdracht.



Rapport

Via het menu Bestand > Export kan er van een fiets een rapport gegenereerd worden. Dit rapport zal in de vorm van een html-pagina op het bureaublad van de gebruiker terechtkomen. Onderstaand voorbeeld geeft het rapport van fiets 003.



Rapport

File | C:/Users/Kris/Desktop/Bike_003.html

Rapport voor fiets 003

Startdatum	Einddatum	Klant	Prijs (EUR)	Afstand (km)
15/03/2020	25/03/2020	Koen Wauters	105,00	788,00
10/03/2020	13/03/2020	Steven Van Gucht	31,50	341,50
2/03/2020	9/03/2020	Marc Van Ranst	73,50	535,00

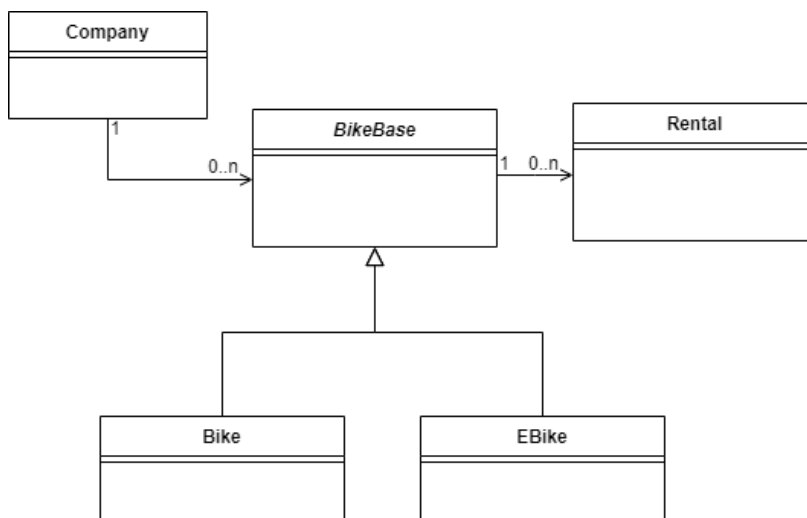
In dit rapport staan alle verhuringen van de fiets. Als een fiets niet is ingeleverd, wordt dat ook opgenomen als een lijn in het rapport (met Afstand 0 km).

Als er geen verhueringen zijn, ziet het rapport er als volgt uit:

Rapport				
C:/Users/Kris/Desktop/Bike_011.html				
Rapport voor fiets 011				
Startdatum	Einddatum	Klant	Prijs (EUR)	Afstand (km)
Geen verhueringen				

Implementatiedetails

Globale klassenstructuur



Centraal staat de abstracte klasse *BikeBase*. Deze klasse bevat de gezamenlijke kenmerken van alle fietsen. Er zijn twee soorten: een gewone fiets (*Bike*) en een elektrische fiets (*EBike*).

Het verhuurbedrijf is voorgesteld door de klasse *Company*. Via deze klasse kan je een lijst verkrijgen van alle fietsen die het bedrijf bezit.

Elke fiets zal verschillende malen verhuurd worden (met verschillende huurperiodes). Elke huurperiode wordt voorgesteld door een object van de klasse *Rental*. Zo zal dus na enige tijd elke fiets een lijstje van verhueringen bevatten.

Stappenplan

Bij deze opgave dien je de voorgestelde aanpak **strikt** op te volgen! De volgende zaken krijg je reeds cadeau in het project:

- De nodige XAML voor de user interface voor het hoofdvenster (MainWindow.xaml) en het tweede venster (RentalWindow.xaml).
- Een aantal klassen die je nog moet vervolledigen. Gelieve de namen van deze klassen niet te wijzigen.
- Code in commentaar. Door stap voor stap de instructies te volgen, zal je deze code uit commentaar kunnen zetten om de applicatie te vervolledigen.

We overlopen nu de code die je moet toevoegen en/of aanpassen. Best volg je de stappen in deze volgorde en hanteer je exact de voorgestelde naamgeving.

BikeBase

- Deze klasse is de basis voor alle soorten fietsen. Er zijn al een aantal properties en methodes gegeven. De meesten spreken voor zich, maar onderstaande lichten we nog even toe:
 - *Brand* : het merk van de fiets
 - *TotalDistance* : de totale afstand die met de fiets is afgelegd sinds het vorige onderhoud. Dit loopt over meerdere huurperiodes heen
 - *LastMaintenanceDate* : datum van het laatste onderhoud
 - *Gender* : waarde die aangeeft of het een dames- of herenfiets is
 - *PricePerDay* : huurprijs per dag (in EUR)
 - *KmPerMaintenanceCycle* : geeft aan hoeveel km er moet afgelegd worden vooraleer de fiets in onderhoud moet. Dit is verder gepreciseerd in de concrete klassen *Bike* en *EBike*.
- *NeedsMaintenance*
 - Implementeer deze methode
- Een fiets beheert een lijst van verhuringen: dit zijn *Rental*-objecten.
 - *IsOccupied*: geeft aan of een fiets bezet (verhuurd) is of niet. Bestudeer de broncode om de logica te snappen hoe je kan bepalen of een fiets al dan niet is verhuurd. Tip: het heeft te maken met een

specifiek *Rental*-object uit de lijst. Je mag deze methode niet veranderen

- *Rent*: deze methode start een nieuwe verhuurperiode. Vul de code aan op basis van de *//ToDo* - commentaar
- *Return*: beëindig de verhuurperiode door de afgelegde afstand te registreren. Lees opnieuw de *//ToDo* -commentaar

Bike en EBike

- Deze klassen zijn gegeven en preciseren wanneer onderhoud moet gebeuren
- EBike heeft een property *BatteryCapacity* van het type *int*

ValidationException

- Maak deze klasse aan en zorg dat dit voldoet aan de eisen van een zelfgemaakte exceptionklasse.

Rental

- De properties uit deze klasse mag je niet veranderen, maar de initialisatie in de constructor moet je nog vervolledigen.
- Je zal hiervoor een methode *CalculatePrice* moeten schrijven in de statische klasse *BusinessRules*. Deze klasse moet statisch blijven!

Company

- Er is code om fietsen aan te maken en testgegevens voor de verhuur. Deze mag je niet wijzigen.
- *CurrentBike* is de huidige fiets die op het “MainWindow” wordt getoond (TIP: zie code in de constructor van *MainWindow*)
- De *Next*-methode selecteert de volgende fiets uit de lijst en deze wordt dan de huidige fiets (*CurrentBike*). Ben je op het einde van de lijst, dan ga je automatisch opnieuw naar de eerste fiets uit de lijst (circulair doorlopen).

MainWindow

- Implementeer *BindCurrentBike*. Deze methode zal de informatie uit het huidige fietsobject halen en tonen op het scherm.
 - Id moet steeds getoond worden met 3 cijfers
 - Voor een EBike verschijnt ook de batterijcapaciteit en een extra symbool
 - Bestudeer goed de screenshots om te bepalen welke knoppen actief zijn en wat hun inhoud is.

RentalWindow

- Zorg dat er een fiets object kan doorgegeven worden naar dit venster. Implementeer in MainWindow de juiste methode om naar dit venster te navigeren.
- Initialiseer het venster door correcte aanroepen van EnableControls en BindCurrentBike. Deze methoden moet je zelf niet meer schrijven.
- Zorg dat de methode UpdateDaysAndTotalPrice wordt aangeroepen elke keer als de gebruiker de slider beweegt. Pas UpdateDaysAndTotalPrice aan zodat het totaal correct wordt berekend.
- Implementeer het verhuren en retourneren:
 - *RentTheBike*
 - Haal de code uit commentaar en implementeer de validatieregel in methode CheckStartDate van BusinessRules. Deze methode zal een ValidationException kunnen gooien.
 - Vang hier geen exceptions op
 - *ReturnTheBike*
 - Haal de code uit commentaar
 - Vang hier geen exceptions op
 - submitButton_click()
 - Haal de code uit commentaar
 - Zorg dat alle mogelijke exceptions uit *ReturnTheBike* en *RentTheBike* worden opgevangen
 - Toon een gepaste foutmelding (let op de screenshots!). Bij een foute ingave mag het venster niet worden gesloten.

MainWindow

- Implementeer *exportItem_Click* om het rapport te genereren
 - Zorg dat het rapport terechtkomt op het bureaublad van de huidige gebruiker
 - Gebruik de methode *CreateReport* uit de klasse *ReportUtils*. Deze methode moet je nog vervolledigen. Het rapport is in essentie één lange string waarin je nog een zoek- en vervangalgoritme moet toepassen op de meest optimale manier. Lees hierover de //ToDo - commentaar.

- Er zijn in je solution ook twee voorbeeldrapporten beschikbaar:
bestudeer dit aandachtig.