**dictionary()**

1. Start.
2. Een dictionary maken van ‘KluisDict’ met twintig keys.
3. FietsenStalling.txt opslaan als readFile.
4. Er wordt een for-loop aangemaakt, elk element in readFile wordt lijn genoemd.
5. In de for-loop wordt per element de lijn gesplit, en dit wordt ‘splitLine’ genoemd.
6. ‘kluisNummer’ is het eerste gesplitste stukje.
7. ‘dateTime’ is het 2de en 3de gesplitste stuk.
8. OVNummer is het laatst gesplitste stuk.
9. Voor elk item (kluisNummer) in de dictionary kluisDict, wordt de tijd (dateTime) en het OV nummer (OVNummer) toegevoegd.
10. De dictionary ‘KluisDict’ wordt gereturned aan de functie.
11. Einde.

**kluisAanvragen()**

1. Start.
2. Je stopt de uitkomst van de functie dictionary() in kluisDict.
3. Je opent de file FietsStalling.txt en stopt dit in de variabele readFile.
4. In de for-loop wordt voor de dictionary kluisDict, elk element ‘kluis’ genoemd en doorgelopen.
5. Kijken of de value van kluis leeg is.
6. False
   1. Gaat door met de volgende kluis.
7. ValueError
   1. beginSchermTitel geeft de waarde: ‘Geen geldige invoer’ weer.
8. True
   1. De kluis de waarden huidige tijd en OV-nummer geven.
   2. readFile leegmaken. (False): return.
   3. for-loop wordt gemaakt voor de dictonary KluisDict, elk element ‘item’ genoemd en doorgelopen. (False): End
   4. If statement: ‘Als de waarden in item iets zijn’
      1. True: Schrijf je de kluis (item) en de inhoud van de kluis (item) naar readFile, en vervolgens ga je naar het volgende item.
      2. False: dan gaat hij verder naar het volgende item.
   5. Return
9. Als alle kluizen wel een waarde hebben, dan wordt er ‘Geen kluizen vrij’ op het beginscherm weergegeven, gevolgd door een return.
10. End.

**KluisOpenen()**

1. Start.
2. Je stopt de uitkomst van de functie dictionary() in kluisDict.
3. In de for-loop wordt voor de dictionary KluisDict, elk element ‘kluis’ genoemd en doorgelopen.
4. For-loop ‘voor kluis in kluisDict’, checkt of de waarden van kluis iets is, en het opgegeven OV zit in de KluisDict.
5. Als dit waar is, dan ‘wordt de kluis geopend’. (Vervolgens wordt dit gereturned, en slaat de volgende stappen over tot eind.).
6. Als er een ValueError optreedt, (als het opgegeven OV geen integer is), dan komt er ‘geen geldige invoer’ te staan. (Vervolgens wordt gereturned, en slaat de volgende stappen over tot het einde.).
7. Als er geen waardes in de kluis staan bij het opgegeven OV, wat anders is dan een integer, wordt op het beginscherm weergegeven dat het OV nummer onbekend is. Vervolgens wordt gereturned.
8. End.

**KluisVrijgeven()**

1. Start.
2. Je stopt de uitkomst van de functie dictionary() in KluisDict.
3. Open de file ‘FietsStalling.txt’ en sla deze op in de variabele readFile.
4. For-loop ‘voor kluis in KluisDict’, kluis wordt een element van KluisDict, kijkt voor elk kluisje de volgende statement:
5. Of (if-statement) het OV in de kluis voorkomt.
6. True
   1. Kluis wordt leeggemaakt
   2. BeginSchermTitel laat zien dat de ‘kluis is vrijgegeven’.
   3. ReadFile wordt leeggemaakt.
   4. For-loop wordt aangemaakt: ‘voor item in KluisDict’ item wordt een element van de dictionary KluisDict, en wordt doorlopen.
   5. If-statement kijkt (voor elk item): ‘of de waarde van het item iets is’.
      1. True
         1. Het item zelf (kluisnummer) en de inhoud van het item (datum/tijd en OV nummer) schrijven naar readFile.
         2. Gaat door naar het volgende item in KluisDict
      2. False
         1. Gaat door naar het volgende item in KluisDict
   6. Return statement (gaat uit de for-loop).
7. False
   1. Gaat gelijk door naar een nieuwe kluis in KluisDict
8. ValueError
   1. Op het beginSchermTitel wordt: ‘Geen geldige invoer’ geprint.
   2. Return statement (gaat uit de for-loop).
9. Als er geen fiets is gestald op je OV, dan wordt er op het beginSchermTitel: ‘Dit OV nummer is onbekend’ geschreven.

**kluisInfo()**

1. Start.
2. Je stop de uitkomst van de functie dictionary() in de variabele KluisDict.
3. In de for-loop wordt voor de dictionary KluisDict, elk element ‘kluis’ genoemd en doorgelopen.
4. If-statement, kijken of het opgegeven OV in de kluis zit.
5. True
   1. BeginSchermTopTitel krijgt de uitkomst van de fietsstaltijd(), en geeft dit weer.
   2. BeginSchermTitel geeft de huidige kosten weer/krijgt de uitkomst van de prijs().
   3. Return statement
6. False
   1. De for-loop kijkt naar de volgende kluis.
7. ValueError
   1. BeginSchermTitel krijgt de waarde ‘Geen geldige invoer’, en geeft dit weer.
   2. Return statement en de volgende kluis begint.
   3. Wanneer het OV nummer niet bekend is, (het opgegeven OV nummer niet in de dict voorkomt) dan wordt er op het BeginSchermTitel: ‘Dit OV nummer is onbekend’ weergegeven.
   4. Return statement.
8. End.

**prijs(begintijd)**

1. Start.
2. Leest de starttijd in, en zet dit in een goed formaat.
3. Leest de huidige tijd in, en zet dit in een goed formaat.
4. Het berekende verschil (in hele dagen) van de huidige tijd en de starttijd, wordt toegekend aan de variabele: ‘dagVerschil’.
5. Het berekende verschil (in seconden tot 1 dag) van de huidige tijd en de starttijd, wordt toegekend aan de variabele: ‘secondeVerschil’.
6. Het verschil in aantal hele dagen, wordt omgezet in een aantal hele minuten, en dit aantal wordt toegekend aan de variabele: ‘dagMin’.
7. Het verschil in aantal hele seconde, wordt omgezet in een aantal hele minuten, en dit aantal wordt toegekend aan de variabele: ‘secMin’.
8. De variabelen dagMin en secMin worden bij elkaar opgeteld en in de variabele ‘minuten’ gestopt, nu weet je hoeveel minuten je fiets in totaal staat gestald in de kluis.
9. De uitkomst van: ‘minuten \* 0,00833333333’ (50 cent per uur) komt in ‘totaalPrijs’ te staan.
10. De variabele totaalPrijs wordt gereturned.