



---

# DOKUMENTATION ZUM INFORMATICUP

---

IntelliTank



Gottfried Wilhelm Leibniz Universität Hannover

Von Axel Claassen, Burak Kadioglu und Sebastian Drath

# Inhalt

Einführung .....	3
Architekturbeschreibung.....	3
Konzept.....	3
Schnittstelle .....	4
Struktur.....	7
Funktionen.....	8
Installationsanleitung .....	10
Voraussetzungen .....	10
Einrichtung .....	13
Inbetriebnahme.....	14
Betreiberhandbuch .....	15
Hauptfenster .....	15
Route – Ansicht .....	19
Vorhersagezeitpunkt – Ansicht .....	21
Preisentwicklung .....	23
Validierung – Ansicht.....	26
Validierung .....	28
Troubleshooting .....	32
Warnungen.....	32
Fehlermeldungen .....	34
Ausblick .....	36

## Einführung

Dieses Projekt ist im Rahmen des „InformatiCups“ 2018 entstanden. Das Ziel dabei war es, ein Programm zu entwickeln, das zum einen eine Tankstrategie berechnet, um auf einer Route über mehrere Tankstellen die günstigste Art zu tanken ausgibt. Außerdem sollte es möglich sein, anhand der gegebenen Preishistorie, Preise bis zu einen Monat in die Zukunft bestimmen zu können. Dadurch wäre es möglich, Fahrwege im Vorfeld zu planen, um so effizient wie möglich fahren zu können.

## Architekturbeschreibung

### Konzept

In diesem Kapitel wird beschrieben, welche Entscheidungen während der Projektphase getroffen wurden und warum diese so getroffen wurden.

Zunächst mussten wir uns für eine Programmiersprache entscheiden. Da nicht klar ist, welches Betriebssystem die letztendlichen Nutzer verwenden, wäre es vorteilhaft, eine möglichst große Menge an Betriebssystemen abzudecken. Deswegen und aufgrund der Tatsache, dass alle Gruppenmitglieder Erfahrungen mit dieser Programmiersprache haben, wurde Java bzw. JavaFX gewählt.

Die Anforderungen an das Programm wurde von Seiten des Veranstalters vergleichsweise knappgehalten. Dies wiederum eröffnete viel Spielraum in der Eigengestaltung. Zu Beginn des Projekts kristallisierten sich während der Brainstorming-Phase zwei Schwerpunkte heraus, in die man das Projekt ausrichten könnte: „*Präzision*“ und „*Visibility*“.

„*Präzision*“ als Schwerpunkt umfasst eine Mischung aus Forschungsarbeit (welche Faktoren Einfluss auf den Benzinpreis haben könnten), sowie eine Optimierung des Fahrtweges.

„*Visibility*“ hingegen hat den Schwerpunkt auf benutzerfreundlicher Darstellung.

Schlussendlich entschieden wie uns, den Schwerpunkt auf eine Benutzerfreundliche Darstellung zu legen. Einerseits, weil Präzision ein Bereich ist, in dem es theoretisch keine Grenze gibt. Während des Brainstormings fielen uns immer neue Einflussfaktoren ein, die möglicherweise eine erhöhte Präzision zur Folge haben könnten: Genaue Fahrtrouten mit Google Maps, aktuelle Staumeldungen, Mindest-/Höchstgeschwindigkeit der Strecken und ein dazu passender Benzinverbrauch sind nur einige Beispiele. Konsequenterweise hätte man auch Tankstellen, an denen wegen des erhöhten Preises nicht getankt wird, vorbeifahren können, um weiter Benzin zu sparen. Da aber laut Aufgabenstellung jede Tankstelle der Route besucht werden muss, würde dadurch ein großer Teil der Optimierungsarbeit untergehen.

Ein weiterer Punkt, der für „Visibility“ spricht, ist, dass es keine Anforderung für eine benutzerfreundliche Bedienung gab. Dies war einer der Punkte, die unsere Gruppe als besonders wichtig erachtet hat, da wir riesige Datenmengen zur Verfügung hatten, aber keine Möglichkeit sie darzustellen. Hätten wir in diesem Punkt nur das nötigste gemacht, hätte das Programm in dem Routendokument nur die vorhergesagten Preise ergänzt. Um nun aber diese Route abzufahren, hätte man jeweils die ID der Tankstelle nehmen müssen, und in einer anderen Datei schauen müssen, wie die Adresse der jeweiligen Tankstelle ist.

In Anbetracht der uns zur Verfügung stehenden Zeit, sowie die Tatsache, dass unsere Gruppe nur aus drei aktiven Mitgliedern bestand, entschieden wir uns, ein realistisches Ziel zu setzen: Eine übersichtliche GUI, die Routen inklusive Tankstrategie und Vorhersagepunkte grafisch darstellt, sowie Graphen für historische Preise an den jeweiligen Tankstellen darstellen. Als Framework für die GUI wurde *JavaFX* verwendet.

Im Bereich Präzision haben wir uns auf einen möglichst akkuraten Vorhersagealgorithmus fokussiert. Außerdem haben wir einige Einflussfaktoren wie zum Beispiel Ferien in den jeweiligen Bundesländern mit eingebunden.

Dadurch haben wir eine gute Balance zwischen „Visibility“ und Präzision erreicht.

## Schnittstelle

Sämtliche Daten des Programms werden im Ordner „Eingabedaten“ bereitgestellt. Wie die im Folgenden beschriebenen Eingabedaten beschafft und zur Verfügung gestellt werden, ist im Kapitel [Installationsanleitung](#) beschrieben.

Dafür existiert zunächst eine csv-Datei, die alle Tankstellen in Deutschland enthält (*Tankstellen.csv*). Das Format dieser Datei entspricht den Vorgaben und enthält beispielsweise eine ID, den Namen, die Adresse, sowie die Position der Tankstelle (Längen- und Breitengrad).

Die Formatierung einer Tankstellenzeile sieht dabei wie folgt aus:

```
<ID>;<Name>;<Marke>;<Straßenname>;<Hausnummer>;<Postleitzahl>;<Ort>;<Breitengrad>;<Längengrad>
```

Außerdem enthält der Ordner „Benzinpreise“ je eine csv-Datei pro Tankstelle, in der die historischen Benzinpreisänderungen aufgelistet sind. Hierbei handelt es sich um ein Datum, gefolgt von einem Preis, der ab diesem Zeitpunkt bis zur nächsten Änderung gilt. Auch diese Eingabedateien entsprechen der Formatierung aus den Vorgaben.

Die Formatierung einer Benzinpreisänderung als Zeile innerhalb einer Datei mit der ID der Tankstelle im Namen sieht dabei wie folgt aus:

```
<Datum>;<Preis>
```

Im Ordner „Fahrzeugrouten“ sind die importierten Routen hinterlegt. Wenn diese über die GUI importiert werden, werden die Dateien in diesen Ordner verschoben und können geöffnet werden. Näheres zum Öffnen der Routen wird im Kapitel [Betreiberhandbuch](#) beschrieben. Die Routendateien sind wie folgt formatiert:

<Datum an dem die Tankstelle besucht wird>;<Tankstellen-ID>

In der ersten Zeile einer Routendatei befindet sich das Tankvolumen als einzelne ganze Zahl.

Im Ordner „Vorhersagezeitpunkte“ sind Dateien mit einzelnen Vorhersagezeitpunkten für verschiedene Tankstellen hinterlegt. Es können unabhängig voneinander beliebige Tankstellen angegeben werden, für die die Preise zum angegebenen Datum vorhergesagt werden. Zusätzlich wird ein Datum angegeben, bis zu dem die historischen Benzinpreise, auf dessen Basis die Vorhersage getätigt wird, als bekannt angenommen werden.

<Datum bis zu dem Preise bekannt sind>;<Vorhersagedatum>;<Tankstellen-ID>

Ein Datum liegt dabei bei allen bisher beschriebenen Dateien im Format

YYYY-MM-DD hh:mm:ss+HH

vor. Dabei entsprechen Y=Jahr, M=Monat und D=Tag, während h=Stunde, m=Minute, s=Sekunde und H=Zeitverschiebung, die durch verschiedene Zeitzonen entstehen. Damit wäre das Abgabedatum dieses Projekts wie folgt aufgebaut:

2017-01-21 23:59:59+01

Zusätzlich zu den genannten Eingabedaten werden für die entwickelte Vorhersage außerdem die Feriendaten für jedes Bundesland verwendet. Dafür wird zunächst eine Zuordnung von Postleitzahlen zu Bundesländern eingelesen. Die Eingabedatei dazu heißt *Postleitzahl2Bundesland.csv*. Da die Postleitzahlen immer in Blöcken zu einzelnen Bundesländern zugeordnet werden können, findet man in dieser Datei jeweils eine untere Grenze eines Postleitzahlenbereichs, gefolgt von einer oberen Grenze und dem Bundesland. So werden beispielsweise die Postleitzahlen von 01001-01936 dem Bundesland Sachsen zugeordnet. Die genaue Formatierung sieht wie folgt aus:

<untere Postleitzahl>;<obere Postleitzahl>;<Bundesland>

Die Bundesländer sind dabei mit ihrem üblichen, vollständigen Namen gespeichert. Die Daten wurden von der Seite ([hier](#)<sup>1</sup>) bezogen und aufbereitet.

---

<sup>1</sup> <https://cebus.net/de/plz-bundesland.htm>

Da nun das Bundesland zu einer Tankstelle bekannt ist, kann bei der Vorhersage abgefragt werden, ob am Ort einer Tankstelle zu einer bestimmten Zeit Schulferien waren oder nicht. Diese Schulferiendaten werden im Ordner „Ferien“ bereitgestellt. Es stehen die Schulferien für die Jahre 2013-2020 zur Verfügung. Dabei sind die Ferien für jedes Jahr in einer separaten txt-Datei gespeichert. In der Datei für ein Jahr sind alle Ferien aller Bundesländer enthalten. Die Datei besteht aus 16 Blöcken, für jedes Bundesland einer. Diese beginnen mit einer Zeile mit dem Namen des Bundeslandes, zu dem die folgenden sechs Zeilen mit Feriendaten gehören. Sie stehen in der Reihenfolge für:

- Winterferien
- Osterferien
- Pfingstferien
- Sommerferien
- Herbstferien
- Weihnachtsferien

Die Ferien sind entweder als einzelner Tag, als Bereich oder als Kombination mehrerer Tage bzw. Bereiche angegeben, da beispielsweise zu den Pfingstferien, sowohl der Brückentag an Pfingsten, als auch der Brückentag an Himmelfahrt gezählt werden. So werden die Ferien je nachdem, ob es sich um einzelne Tage, Bereiche oder eine Kombination daraus handelt, wie folgt beschrieben:

Einzelner Tag	Bereich	Beliebige Kombination durch Schrägstrich getrennt
DD.MM.	DD.MM.-DD.MM.	DD.MM./DD.MM.-DD.MM.

Wenn der jeweilige Ferientyp im Bundesland nicht existiert, wird ein Strich (-) in der jeweiligen Zeile gesetzt. Beispielsweise gibt es in Baden-Württemberg keine Winterferien. Die Feriendaten wurden von der Seite ([hier](#)<sup>2</sup>) für das jeweilige Jahr bezogen und aufbereitet. Die Links zu den anderen Jahren sehen dementsprechend anders aus.

Weiterhin werden Trainingsdaten, die für die Vorhersage erstellt werden, gespeichert, damit nicht immer wieder neu komplett vorhergesagt werden muss. Diese werden nur wiederverwendet, wenn die zugrundeliegenden Daten für die Vorhersage nicht verändert wurden. Dazu zählen nicht die Ferien, die Bundeslandzuordnung, sowie die Preise und Tankstellen, sondern lediglich die Routen, sowie Vorhersagezeitpunkte. Sollten die Preise für einzelne Tankstellen geändert werden, so müssen die Trainingsdaten manuell gelöscht werden. Änderungen der Feriendaten, Preisdaten, Bundeslandzuordnungsdaten, sowie Tankstellendaten sind für unser Programm nicht vorgesehen und außerdem auch

---

<sup>2</sup> [https://www.schulferien.org/Schulferien\\_nach\\_Jahren/2017/schulferien\\_2017.html](https://www.schulferien.org/Schulferien_nach_Jahren/2017/schulferien_2017.html)

nicht erforderlich. Näheres zur Vorhersage, zur Speicherung und Wiederverwendung der Trainingsdaten ist in Kapitel [Funktionen](#) erläutert.

Im Ordner „Ausgabedaten“ werden die Vorhersagen, sowie die Tankstrategie für eine Route ausgegeben.

Im Unterordner „Tankstrategien“ befinden sich die Ausgabedateien von Vorhersagen für Routen. Die Dateien sind genauso wie die Eingabedateien benannt. Diese werden automatisch erstellt, wenn eine Route fertig geladen und vorhergesagt wurde. Das Format entspricht dem Eingabeformat für Routen und wurde nur um zwei Spalten erweitert. Außerdem wird die erste Zeile mit der Tankkapazität nicht mehr ausgegeben. Die beiden neuen Spalten pro Tankstopp setzen sich zusammen aus dem vorhergesagten Preis für das angegebene Datum, sowie der Benzinmenge die hier getankt werden sollte, um möglichst günstig zu fahren.

Im Unterordner „Vorhersagen“ befinden sich die Ausgabedateien von Vorhersagen für die Vorhersagezeitpunkte. Auch diese Dateien werden nach Fertigstellung der Vorhersage automatisch erstellt und sind genauso benannt wie die Eingabedateien. Diese Ausgabedateien wurden lediglich um eine Spalte erweitert, in der der vorhergesagte Preis steht.

Wenn beispielsweise aufgrund fehlender Daten kein Preis vorhergesagt werden konnte, wird in den Ausgabedateien eine -1 geschrieben.

## Struktur

Für dieses Projekt wurde das Model-View-Controller Pattern (MVC) eingesetzt, damit einzelne Teile ausgetauscht werden können. Im Folgenden werden die Datenstrukturen im Modell kurz beschrieben. Zunächst gibt es für jede Tankstelle ein *GasStation* Objekt, welches die importierten Attribute sowie die Liste an historischen Preisen enthält. Ein solches *Price* Objekt besteht wiederum aus dem eigentlichen Preis und der Zeit, zu dem dieser Preis wirksam wurde. Ferner gibt es die Klassen *RefuelStop* und *PredictionPoint*, die einen Tankstopp respektive Vorhersagezeitpunkt repräsentieren. Zusätzlich zu den importierten Attributen besitzen beide Klassen eine Liste von den vorhergesagten Preisen. Diese werden im Gegensatz zu den historischen Preisen, die in einem *GasStation* Objekt enthalten sind, in diesen Klassen gesetzt, da die vorhergesagte Preisliste abhängig vom Zeitpunkt ist, ab dem vorhergesagt wird. Da beide über ein *GasStation* Objekt, eine Liste mit (vorhergesagten) *Price* Objekten und eine Vorhersagezeit verfügen, sowie ähnliche Aufgaben erfüllen, implementieren beide das gemeinsame Interface *IPredictionStation*.

Die Klassen *Route* und *PredictionPointList* fassen die Vorhersagezeitpunkte respektive Tankstopps zusammen und entsprechen den importierten Routen und Vorhersagezeitpunkten. Auch diese Klassen besitzen ähnliche Methoden und Aufgaben, sodass auch hier ein gemeinsames Interface namens *IPredictionStationList* implementiert wird.

Sowohl *IPredictionStationList* als auch *IPredictionStation* erweitern zudem das Interface *Validateable*, sodass es möglich ist, sowohl Routen und Vorhersagezeitpunktlisten, als auch einzelne Tankstopps und Vorhersagezeitpunkte zu validieren.

Diese Beschreibung bietet nur eine grobe und vereinfachte Übersicht über die Datenstruktur. In einigen der beschriebenen Klassen sind noch einige weitere Attribute enthalten, die zum Beispiel der Ergebnisse der Tankstrategie speichern. Weitere Informationen sind in der beiliegenden Java Dokumentation beziehungsweise auch im Quellcode zu finden.

## Funktionen

In diesem Kapitel werden interne Features des Programms beschrieben. Alle im Vordergrund befindlichen Funktionen werden im Kapitel „Betreiberhandbuch“ näher erläutert. Auf zwei Punkte wird hier besonders eingegangen: Die Vorhersage der Tankdaten, sowie die Tankstrategie-Algorithmen.

Für die Tankstrategien wurden zwei verschiedene Algorithmen verwendet, die im nachfolgenden Abschnitt behandelt werden.

### Tankstrategie 1: „Fixed path gas station problem“

Der Algorithmus basiert auf den Forschungsergebnissen von Samir Khuller, Azarakhsh Malekian und Julián Mestre. Dort wurde ein Algorithmus entwickelt, der effizient die günstigste Art zu tanken auf einer festgelegten Route berechnet. Weitere Informationen zu dem Paper finden Sie [hier](#)<sup>3</sup>.

### Tankstrategie 2: Naive Tankstrategie

Diese Tankstrategie steht repräsentativ für die Art, wie Menschen normalerweise tanken würden: Sobald der Tank knapp wird und er nicht bis zur nächsten Tankstelle reicht, wird vollgetankt. Außerdem wird überprüft, wie weit das Ziel noch entfernt ist und sollte es in Reichweite sein, wird nur so viel getankt wie nötig, um dort anzukommen. Dieser Zusatz ist zwar nicht besonders realitätsnah, aber dadurch lässt sich besser vergleichen, wie viel besser die intelligente Tankstrategie im Vergleich zu der naiven Strategie ist, wenn beide dieselbe Menge Benzin tanken.

### Vorhersage

Unser Programm bietet die Möglichkeit Preise auf Basis historischer Preise vorherzusagen. Die Preise liegen dabei als Kombination aus dem eigentlichen Preis und dem Datum, an dem der Preis geändert wurde vor.

Um die Vorhersage so genau wie möglich zu gestalten, werden die Preise für jede Tankstelle separat vorhergesagt. Dadurch werden lokale Besonderheiten bei der

---

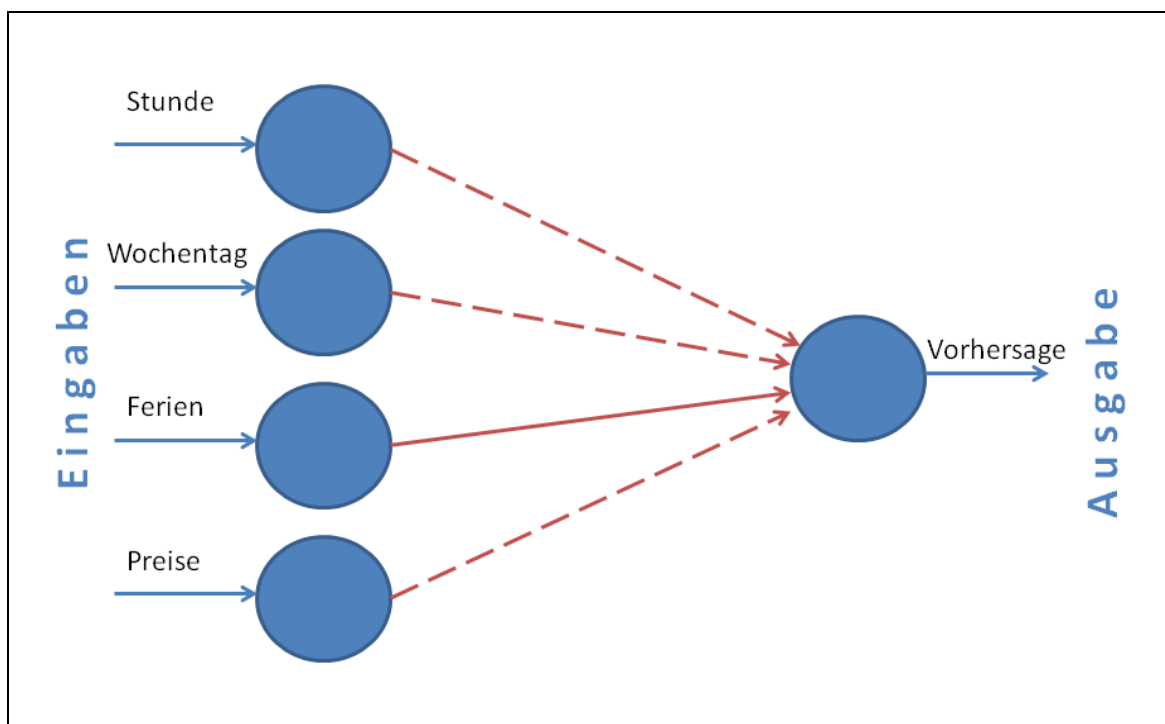
<sup>3</sup> <https://www.cs.umd.edu/~samir/grant/gas-j.pdf>



Preisentwicklung einzelner Tankstellen direkt berücksichtigt. Als Basis für unsere Vorhersage wurden zunächst der Wochentag, die Stunde eines Tages sowie die Preise der letzten 5 Stunden verwendet. Um die Vorhersage später zu verbessern, wurden zusätzlich die Feriendaten des Bundeslandes, in dem die jeweilige Tankstelle liegt, verwendet. Zusätzlich zu den Feriendaten pro Bundesland und Jahr musste dafür eine Zuordnung von den Postleitzahlen einer Tankstelle zu einem Bundesland erfolgen. Der Import dieser Daten wird im Abschnitt [Schnittstelle](#) beschrieben.

Die Wochentage und Stunden werden der Vorhersageeinheit als Vektoren der Länge 7 beziehungsweise 24 übergeben. Dabei ist jeweils der Eintrag gesetzt, der für den aktuellen Tag beziehungsweise die Stunde steht.

Für die Vorhersage wurde ein einlagiges Perzeptron verwendet. Dabei handelt es sich um ein künstliches neuronales Netz. Der Aufbau eines solchen Perzeptrons ist in der folgenden Abbildung dargestellt:



Die blauen Punkte stehen für die Ein- und Ausgabeneuronen. Die roten Pfeile stehen für Gewichtungen der Eingabewerte. Dabei bestehen die gestrichelten Pfeile aus mehreren Gewichten und wurden hier vereinfacht zusammengefasst. Das Gewicht für die Stunden besteht beispielsweise aus 24 einzelnen Gewichten, für jede Stunde eines. Diese Gewichte werden während des Trainierens mit den historischen Daten so angepasst, dass man aus den Eingaben eine möglichst genaue Vorhersage bestimmen kann. Die Trainingsdaten bestehen dabei aus den Preisen bis zu einem bestimmten Datum, ab dem vorhergesagt werden soll. Die mögliche Vorhersagezeit

ist auf fünf Wochen begrenzt, da die Daten sonst zu ungenau werden. Die Ergebnisse der Vorhersage einer Tankstelle können wie in Kapitel [Betreiberhandbuch](#) beschrieben über die grafische Benutzeroberfläche eingesehen werden. Die Güte der Ergebnisse wird im Kapitel [Validierung](#) bewertet.

Nach dem Training werden die generierten Gewichtungen gespeichert, um in Zukunft schneller eine Vorhersagen abrufen zu können. Die Trainingsdaten werden allerdings nur wiederverwendet, wenn sich die Parameter für die Vorhersage nicht verändert haben. Das heißt, dass sowohl die Tankstelle als auch das Datum, ab dem vorhergesagt wird, identisch sein müssen. Die Trainingsdaten für eine Route werden beispielsweise unter dem Pfad

*Eingabedaten/Fahrzeugrouten/Trainingsdaten/<Routenname>.train*

Als weitere Verbesserung wurde ein mehrlagiges Perzeptron implementiert. Dieses besitzt eine weitere Schicht von Neuronen und dadurch mehr Gewichte. Da sich die Ergebnisse nicht deutlich verbessert haben und der Zeitaufwand für die Berechnung größer war, haben wir uns entschieden, dass wir weiterhin das einlagige Perzeptron beibehalten. Dies stellt einen optimalen Kompromiss aus Geschwindigkeit und Genauigkeit dar.

## Installationsanleitung

In diesem Kapitel wird erklärt, wie die Einrichtung des Programms funktioniert. Dies ist unterteilt in 3 Unterkapitel: Voraussetzungen, Einrichtung und Inbetriebnahme.

### Voraussetzungen

Dieses Programm wurde vorrangig für Windows entwickelt, deshalb empfehlen wir Ihnen, einen Windows Computer/Laptop mit dem Betriebssystem Windows Vista oder neuer zu verwenden. Das Programm unterstützt durch die Verwendung von Java zusätzlich die Möglichkeit auf macOS und Linux basierten Systemen zu laufen, wird aber offiziell nicht unterstützt.

Unabhängig vom Betriebssystem, das Sie verwenden, benötigen Sie eine aktuelle Version des Java Runtime Environments. Die neueste Version finden Sie auf der Offiziellen Oracle Seite. Klicken Sie [hier](#)<sup>4</sup>!

---

<sup>4</sup> <http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html>

[Overview](#)
[Downloads](#)
[Documentation](#)
[Community](#)
[Technologies](#)
[Training](#)

## Java SE Runtime Environment 8 Downloads

Do you want to run Java™ programs, or do you want to develop Java programs? If you want to run Java programs, but not develop them, download the Java Runtime Environment, or JRE™.

If you want to develop applications for Java, download the Java Development Kit, or JDK™. The JDK includes the JRE, so you do not have to download both separately.

[JRE 8u161 Checksum](#)  
[JRE 8u162 Checksum](#)

### Java SE Runtime Environment 8u161

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

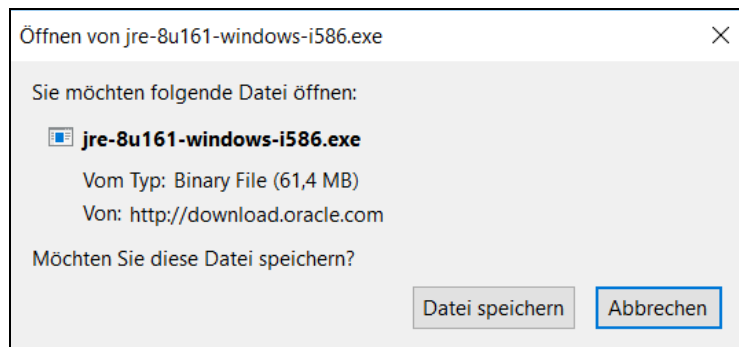
☐ Accept License Agreement
 ☒ Decline License Agreement

Product / File Description	File Size	Download
Linux x86	63.4 MB	<a href="#">jre-8u161-linux-i586.rpm</a>
Linux x86	79.29 MB	<a href="#">jre-8u161-linux-i586.tar.gz</a>
Linux x64	60.4 MB	<a href="#">jre-8u161-linux-x64.rpm</a>
Linux x64	76.35 MB	<a href="#">jre-8u161-linux-x64.tar.gz</a>
macOS	74.17 MB	<a href="#">jre-8u161-macosx-x64.dmg</a>
macOS	65.86 MB	<a href="#">jre-8u161-macosx-x64.tar.gz</a>
Solaris SPARC 64-bit	52.24 MB	<a href="#">jre-8u161-solaris-sparcv9.tar.gz</a>
Solaris x64	50 MB	<a href="#">jre-8u161-solaris-x64.tar.gz</a>
Windows x86 Online	1.78 MB	<a href="#">jre-8u161-windows-i586-iftw.exe</a>
Windows x86 Offline	61.35 MB	<a href="#">jre-8u161-windows-i586.exe</a>
Windows x86	64.56 MB	<a href="#">jre-8u161-windows-i586.tar.gz</a>
Windows x64 Offline	68.02 MB	<a href="#">jre-8u161-windows-x64.exe</a>
Windows x64	68.58 MB	<a href="#">jre-8u161-windows-x64.tar.gz</a>

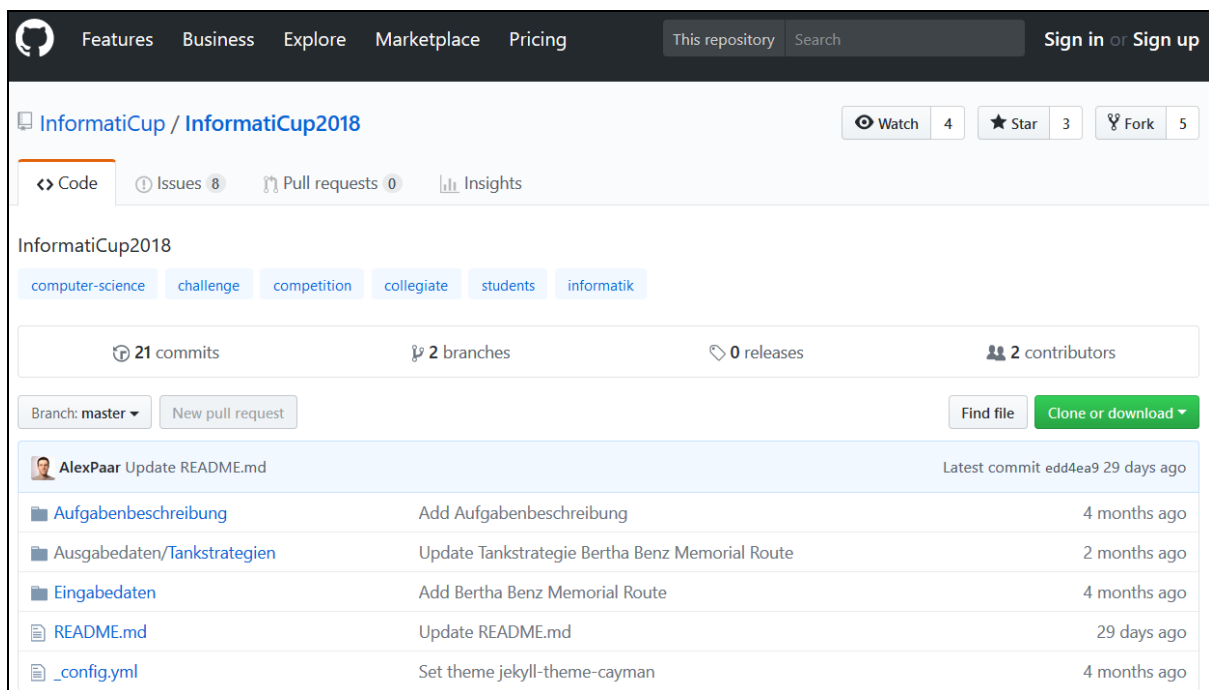
In der Mitte der Webseite sollte eine ähnliche Ansicht wie diese erscheinen. Hier wird die „Java SE Runtime Environment 8u161“ angezeigt. Bei Ihnen ist es möglicherweise eine neuere Version, was aber kein Problem darstellen sollte.

Klicken Sie zunächst auf den leeren Kreis „Accept License Agreement“ nachdem Sie die Lizenzvereinbarung zur Kenntnis genommen haben. Danach laden Sie die Windows Offline-Version herunter, je nachdem welche Windows Version Sie besitzen. (Um herauszufinden, welche Windows Version Sie verwenden, drücken Sie einfach die Windows-Taste und die Pause-Taste gleichzeitig. Es erscheint nun ein Fenster mit allen wichtigen Angaben zu Ihrem Computer. Unter dem Abschnitt „System“ und „Systemtyp“ finden sie die gesuchte Information. Sollten Sie ein 32-Bit-System haben, laden Sie die x86 Version herunter, haben Sie ein 64-Bit-System, laden Sie die x64 Version herunter).

Nachdem Sie ihre Version ausgewählt haben, sollte sich ein Download-Fenster öffnen.



Klicken Sie auf „Datei speichern“. Nachdem der Download abgeschlossen ist, klicken Sie doppelt auf die von Ihnen heruntergeladene Datei, um den Installationsprozess zu starten. Folgen Sie nun die Anweisungen des Installers, um es zu installieren. Des Weiteren benötigen Sie die Daten mit Tankpreisen, die für das Programm notwendig sind. Besuchen Sie hierfür die GitHub Seite des InformatiCups. Klicken Sie [hier](https://github.com/InformatiCup/InformatiCup2018)<sup>5</sup>.



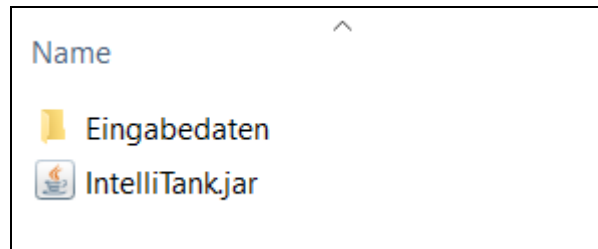
Klicken Sie auf den grünen Button „Clone or download“ und anschließend auf „Download ZIP“. Dieser Download kann einige Zeit in Anspruch nehmen. Entpacken Sie die Dateien anschließend an einem beliebigen Ort, an dem Sie es schnell wiederfinden.

<sup>5</sup> <https://github.com/InformatiCup/InformatiCup2018>

Als letztes benötigen Sie das Programm selbst. Dies sollte aber im Regelfall dieser Dokumentation beiliegen, welches Sie in dem Ordner „IntelliTank“ finden.

## Einrichtung

Um das Programm starten zu können, müssen zunächst die Tankstellendaten mit dem Programm kombiniert werden.



Öffnen Sie dazu zunächst den Programmordner. Dort sollten sich zwei Elemente befinden: Ein Ordner namens „Eingabedateien“ und eine ausführbare Datei „IntelliTank.jar“. Gehen Sie nun zu dem Ort, an dem Sie Ihre zuvor heruntergeladenen Tankpreis-Dateien entpackt haben und kopieren Sie den Ordner „Eingabedateien“ per Rechtsklick auf den Ordner. Gehen Sie wieder zurück zu Ihrem Programmordner, der die zwei Elemente enthielt. Drücken Sie mit Rechtsklick auf eine freie Stelle in dem Ordner und wählen Sie „Einfügen“ aus dem Kontextmenü aus. Nun werden alle nötigen Dateien in den Ordner kopiert. Dieser Vorgang kann einige Minuten dauern. Ist der Vorgang abgeschlossen, kann das Programm in Betrieb genommen werden.

Wenn Sie alles richtig gemacht haben, sollte der Ordner mit dem Programm so aussehen:

- Eingabedaten
  - Benzinpreise
    - 1.csv
    - ...
  - Fahrzeugrouten
    - Bertha Benz Memorial Route.csv
    - ...
  - Ferien
    - 2013.txt
    - ...
  - Vorhersagezeitpunkte
    - Meine Tankstellen.csv
    - ...
  - Postleitzahl2Bundesland.csv
  - Tankstellen.csv
- IntelliTank.jar

## Inbetriebnahme

Um das Programm auszuführen, klicken Sie hierfür doppelt auf die Datei „IntelliTank.jar“. Kurz darauf sollte das Hauptfenster des Programms erscheinen. Alternativ kann das Programm über die Konsole gestartet werden. Dafür muss zunächst mit dem Befehl

```
cd <Pfad zur jar-Datei>
```

in das Verzeichnis mit der Datei „IntelliTank.jar“ gewechselt werden. Daraufhin kann das Programm mit

```
java -jar IntelliTank.jar
```

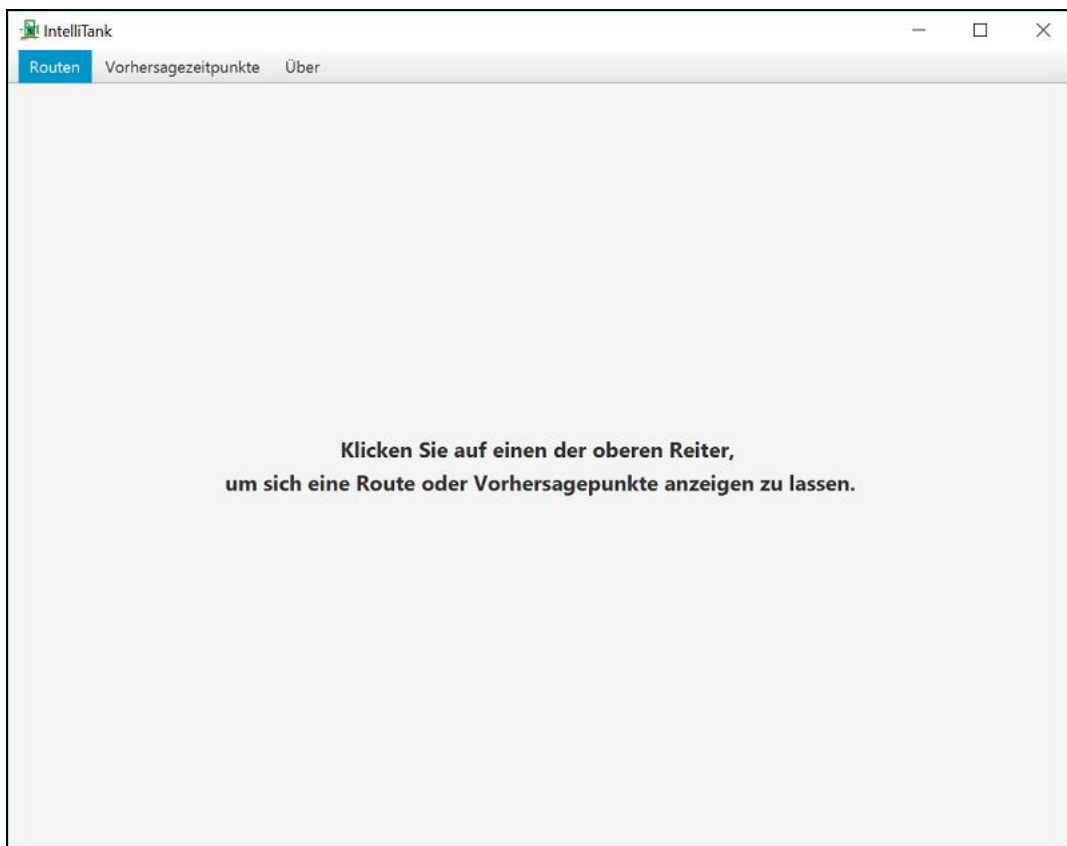
gestartet werden.

## Betreiberhandbuch

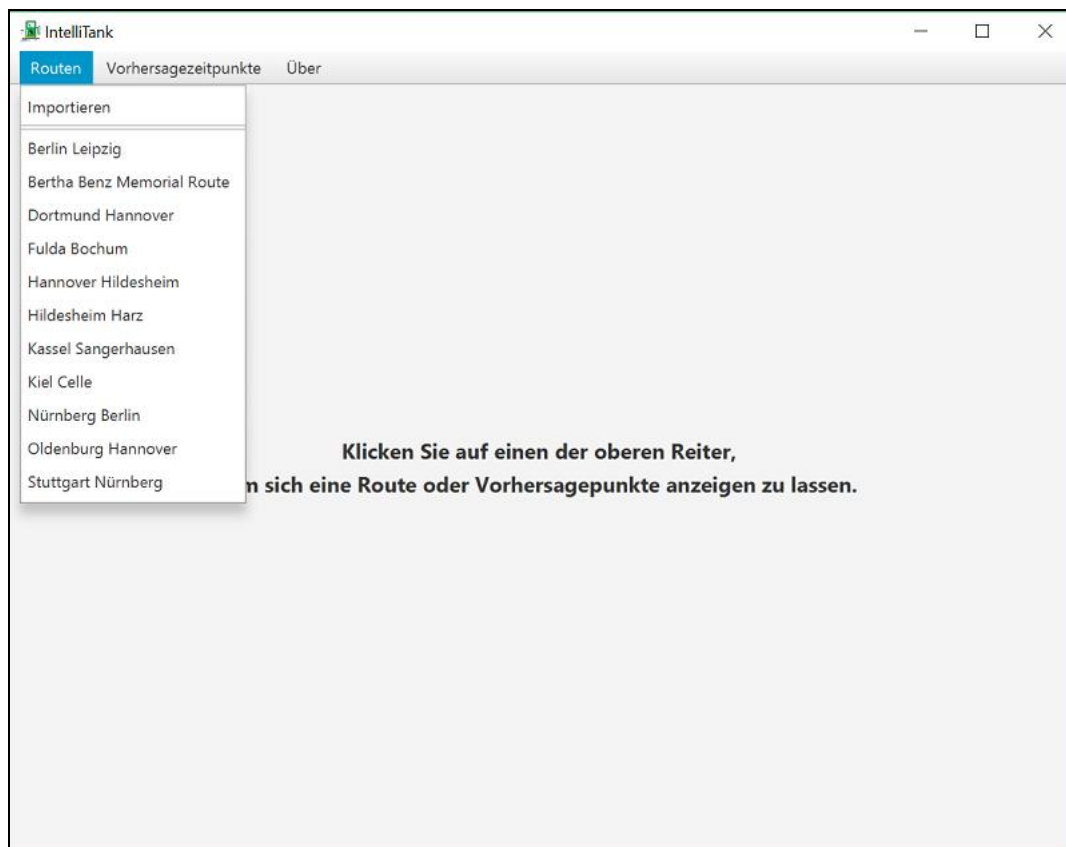
### Hauptfenster

Sobald das Programm gestartet wurde, wird keine Route geladen. Der Grund hierfür ist die Dauer des Ladens einer Route. Damit sich der Benutzer des Programms eine Route nach seinem Interesse laden kann.

Oben auf dem Hauptfenster ist eine Menüleiste abgebildet, die folgende Menüs enthält: „Routen“, „Vorhersagezeitpunkte“ und „Über“. Nach dem Starten des Programms öffnet sich das folgende Fenster:



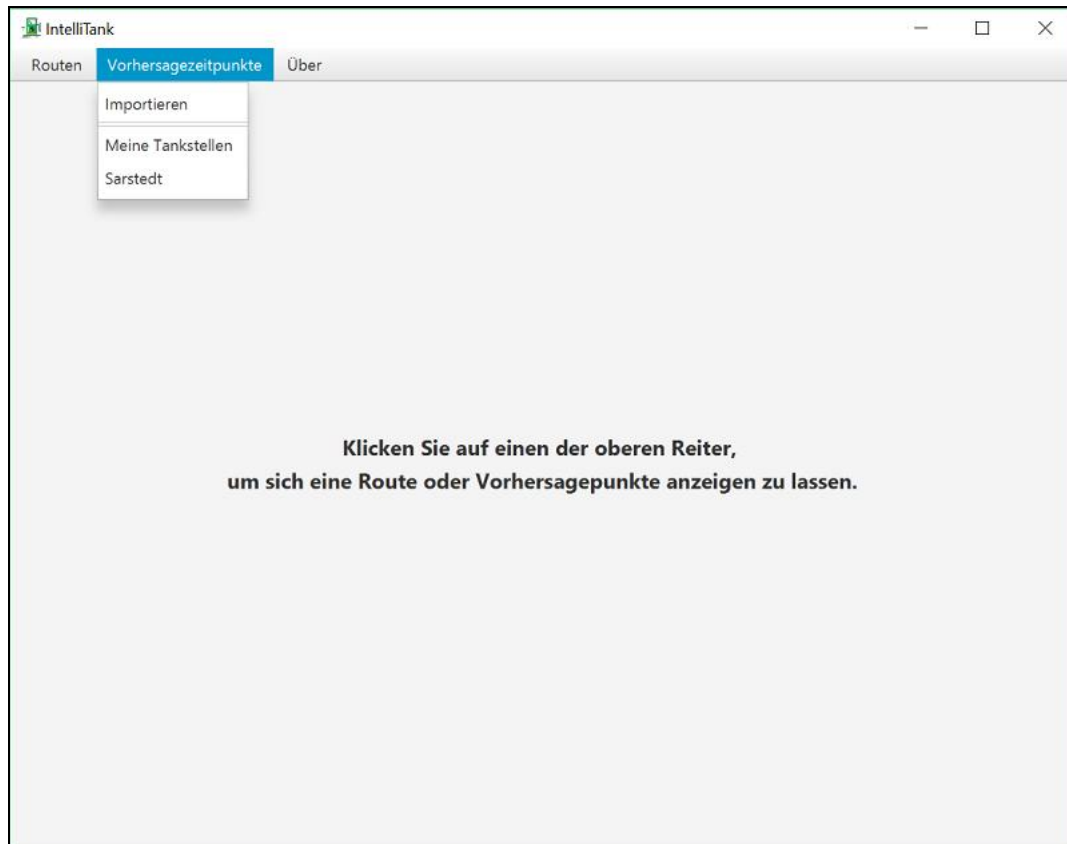
Wenn der Benutzer auf „Routen“ klickt, öffnet sich Folgendes:



Der Benutzer kann hierbei externe Routen in das Programm laden, indem er auf „Importieren“ klickt. In dem Fall werden die ausgewählten Routen in den im Abschnitt [Schnittstelle](#) erwähnten Routenordner kopiert. Die Routen, die sich in diesem Ordner befinden, werden direkt im Menü unterhalb des Imports angezeigt. Diese lassen sich mit einem Mausklick laden. Beim Laden werden die Preise für die Route vorhergesagt und es wird die Tankstrategie berechnet. Im Anschluss an das Laden der Route wird die Routenansicht geöffnet (siehe [Route – Ansicht](#)). Wenn die importierte Datei fehlerhaft ist, wird beim Laden eine Fehlermeldung ausgegeben.

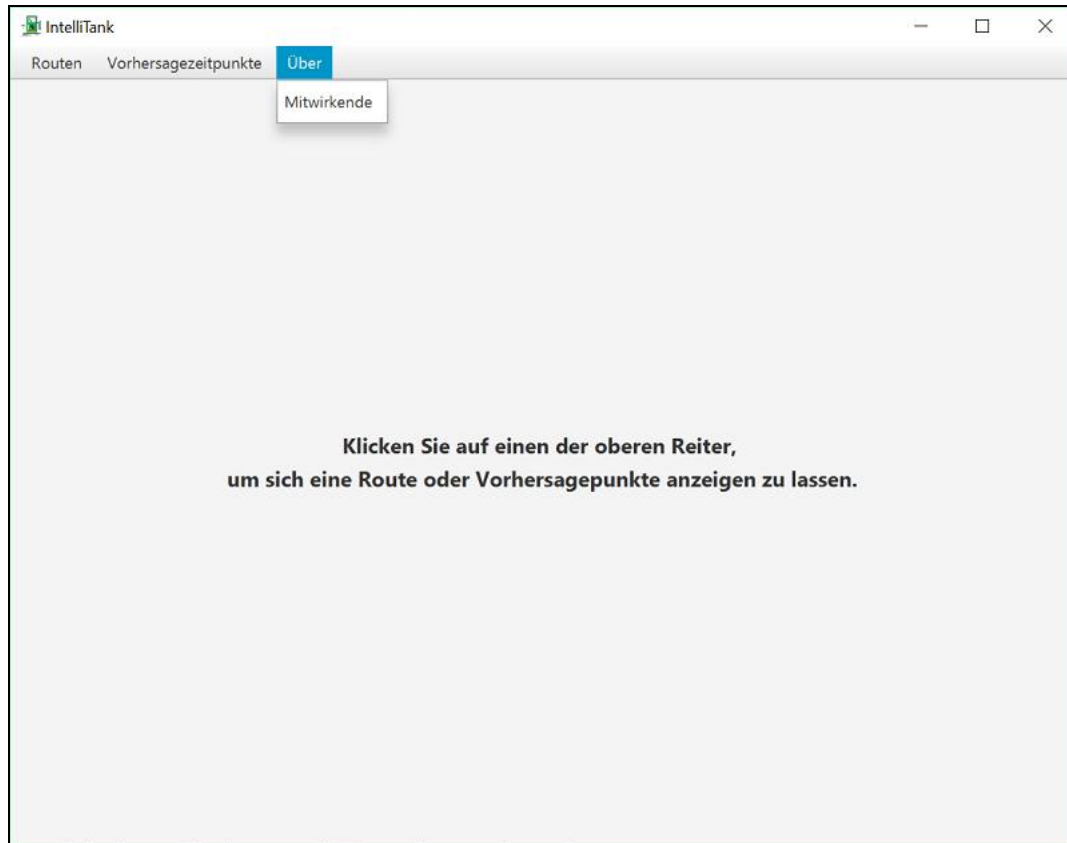


Beim Öffnen des Menüs „Vorhersagezeitpunkte“ wird folgendes angezeigt:

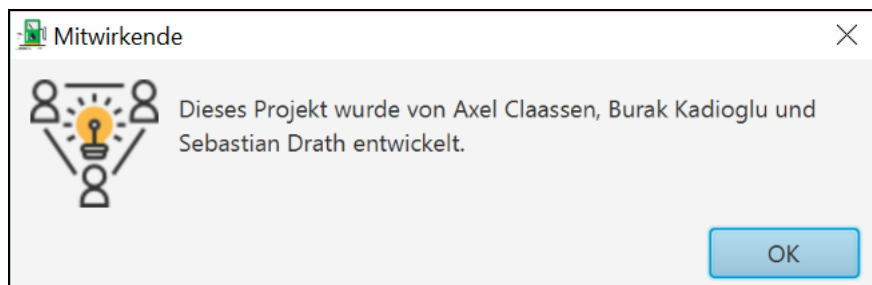


Der Benutzer kann sowohl neue als auch bereits vorhandene Vorhersagezeitpunkte darstellen lassen. Um neue Vorhersagezeitpunkte zu importieren, muss der Benutzer auf „Importieren“ klicken. Vorhandene Vorhersagezeitpunkte lassen sich darstellen, indem auf den entsprechenden Namen geklickt wird. Mit dem Klick auf den Namen wird das Laden der Vorhersagezeitpunkte gestartet. Wenn die importierte Datei fehlerhaft ist, wird beim Laden eine Fehlermeldung ausgegeben. Nachdem die Vorhersagezeitpunkte geladen sind und die Preise vorhergesagt wurden, wird die Vorhersagepunktansicht geöffnet (siehe [Vorhersagezeitpunkt – Ansicht](#)).

Bei einem Klick auf das Menü „Über“ kann der Benutzer anschließend auf „Mitwirkende“ klicken. Dies ist in der graphischen Benutzeroberfläche folgendermaßen dargestellt:

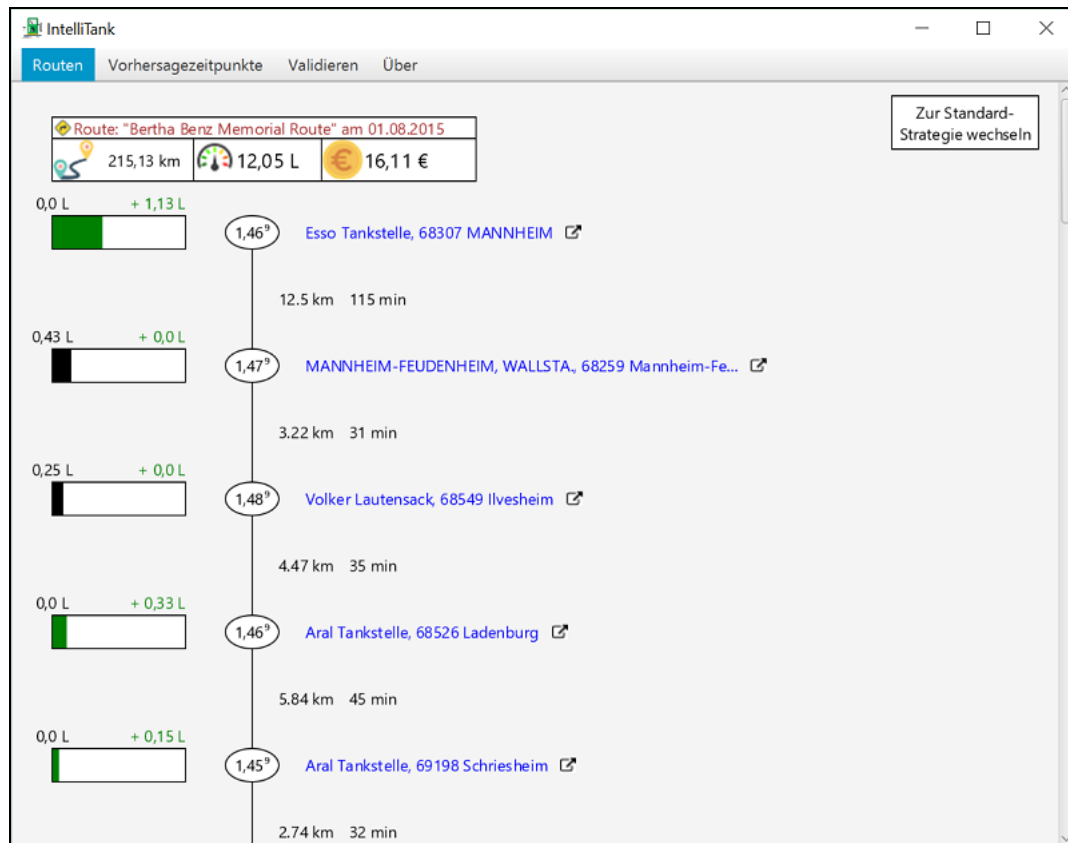


Mit dem Klick auf „Mitwirkende“ öffnet sich ein neues Fenster, indem näherbracht wird, welche Entwickler dieses Programm entwickelt haben:






## Route – Ansicht

In diesem Unterkapitel wird die Ansicht der Route mithilfe der Route „Bertha Benz Memorial Route“ beschrieben. Nachdem diese Route geladen wird, wird der Benutzer mit der folgenden Ansicht konfrontiert:



Auf dieser Ansicht sieht der Benutzer alle Informationen zur befahrenen Route. Im Folgenden wird jedes Element im Detail beschrieben:

Der Kasten oben links gibt dem Benutzer einige Fakten zur Route: Name und Datum der befahrenen Route, gefahrene Strecke in Kilometern, Verbrauch in Litern sowie der Preis in Euro:

Route: "Bertha Benz Memorial Route" am 01.08.2015		
	215,13 km	 12,05 L  16,11 €

In dem Fall wurden 215,13km gefahren und 12,05L getankt. Der gesamte Tankpreis beträgt 16,11€.

Nun folgt die Beschreibung zu der Verbindung zweier Tankstellen:



Das Rechteck beschreibt den aktuellen Tankstatus. Der schwarze Anteil steht dabei für die Benzinmenge, die sich bei Ankunft an dem Tankstopp noch im Tank befindet, während der grüne Teil die Menge an neu getanktem Benzin darstellt. Beim ersten Stopp wurden demnach 1,13 L getankt und beim zweiten befinden sich noch 0,43 L davon im Tank und es wird nichts neu getankt.

In den Ovalen wird der für diesen Tankstopp vorhergesagte Preis angezeigt. Wenn für den Tankstopp kein Preis vorhergesagt werden konnte, weil wie im Kapitel [Funktionen](#) beschrieben beispielsweise historische Daten fehlen, wird ein Preis auf Basis der Preise der anderen Tankstellen innerhalb der Route zu der Zeit des nicht vorhersagbaren Tankstopps geraten. Dabei werden die Preise der Tankstopps in Abhängigkeit zur Entfernung zum Tankstopp ohne Preis in den geratenen Preis mit einbezogen. Dieser geratene Preis wird dann in rot dargestellt.

Rechts neben dem Oval ist der Name der Tankstelle zu sehen. Die Ovale sind mit einer Linie verbunden, die den Abstand der Tankstellen darstellt. Rechts neben dieser Linie befindet sich der Abstand der Tankstellen in Kilometern und die Dauer der Fahrt von Tankstelle A nach B.

Der Button „Zur Standard-Strategie wechseln“ oben rechts auf der Routenansicht wechselt bei einem Mausklick zur naiven Tankmethode. Dabei ändern sich die Daten in dem Kasten oben links. Zudem ändert sich auch die getankte Benzin-Menge der einzelnen Tankstellen. Die verschiedenen Tankmethoden sind im Kapitel [Funktionen](#) beschrieben.

Mit einem Klick auf „Validieren“ in der Menüleiste kann die Vorhersage für die gesamte Route validiert werden. Die Vorhersage kann darüber anhand verschiedener Messwerte beurteilt werden. Die erscheinende Ansicht ist in Kapitel [Validierung – Ansicht](#) erläutert.

In diesem Unterkapitel wird die Darstellung der Vorhersagezeitpunkte beschrieben. Hierzu folgt ein Screenshot der Vorhersagezeitpunkte „Meine Tankstellen“:

Die Vorhersagezeitpunkte werden mithilfe einer Tabelle dem Benutzer dargestellt. In der Tabelle sind die Spalten „Nr.“, „Tankstelle“, „Bekannte Zeit“, „Vorhersagezeitpunkt“, „Preis“ und „Echter Preis“ enthalten. Durch diese Tabelle werden dem Benutzer auf eine einfache Art und Weise alle notwendigen Informationen zur Vorhersagezeitpunkte dargestellt. Die Spalte „Bekannte Zeit“ stellt dabei die Zeit dar, bis zu der historische Benzinpreise für die Vorhersage benutzt wurden. Wenn der echte Preis nicht vorhanden ist, weil die Vorhersagezeitpunkte beispielsweise in der Zukunft liegen, wird hier kein Preis angezeigt. Ebenso wird kein vorhergesagter Preis angezeigt, wenn beispielsweise aufgrund fehlender historischer Daten oder einem Punkt zu weit in der Zukunft keine Vorhersage gemacht werden konnte. Wenn keine Daten vorhergesagt wurden und keine historischen Daten vorliegen sieht die Spalte wie folgt aus:

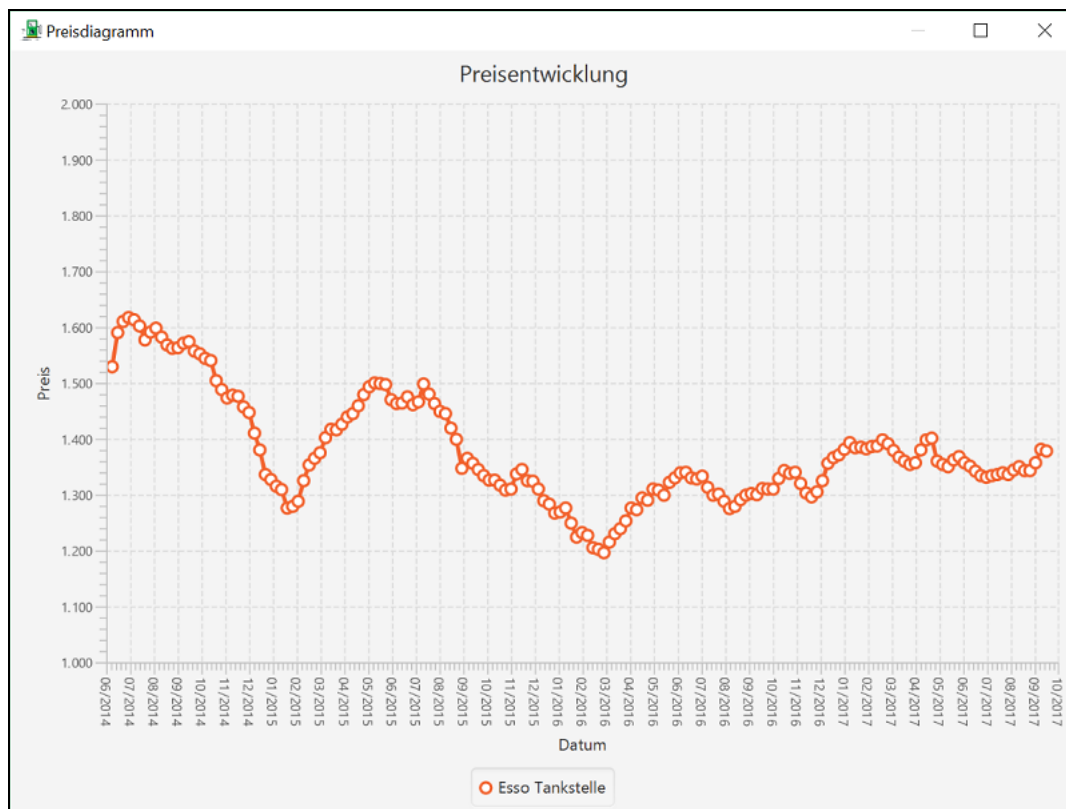
IntelliTank					
Routen Vorhersagezeitpunkte Validieren Über					
Sarstedt					
Nr.	Tankstelle	Bekannte Zeit	Vorhersagezeitpunkt	Preis	Echter Preis
1	JET SARSTEDT HILDESHEIMER STR. 114A	Di 10.02.2015 12:18	So 15.02.2015 21:18	-,-,-,-	-,-,-,-

Mit einem Klick auf „Validieren“ in der Menüleiste kann die Liste von Vorhersagezeitpunkten validiert werden. Die Vorhersage der Punkte kann darüber anhand verschiedener Messwerte beurteilt werden. Die erscheinende Ansicht ist in Kapitel [Validierung – Ansicht](#) erläutert.

## Preisentwicklung

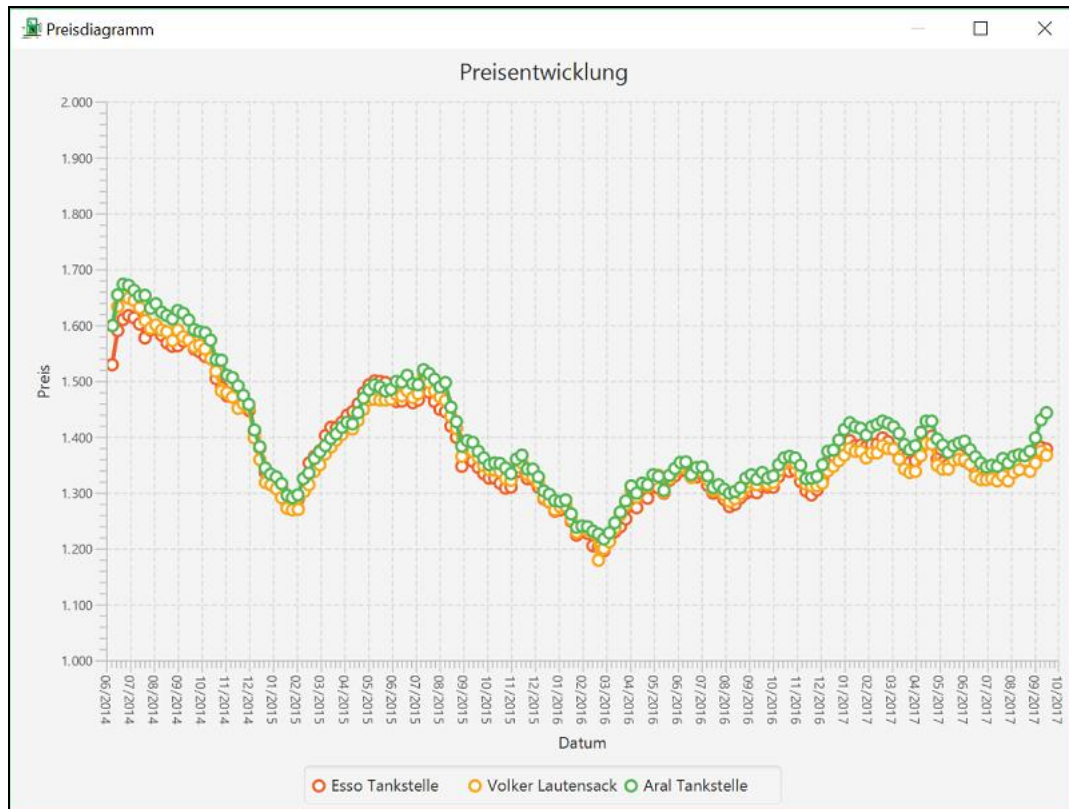
Mithilfe der vorhandenen graphischen Benutzeroberfläche können die historischen Preise einer beliebigen Tankstelle der Route graphisch angezeigt werden. Um dies zu verwirklichen, muss der Benutzer auf den Namen der Tankstelle klicken. Zu einer solchen Preisübersicht kann der Benutzer auch von der Vorhersagezeitpunktansicht gelangen, indem er auf die entsprechende Zeile klickt.

Die Preise der einzelnen Tankstellen sind in dem Zeitintervall [2012, 2017] registriert. Es ist hierbei wichtig zu erwähnen, dass das Zeitintervall bei den Tankstellen sehr unterschiedlich vertreten ist. Es gibt beispielsweise Tankstellen, deren Preise zwischen 2014 und 2017 registriert sind, wohingegen es auch Tankstellen gibt, die zum Beispiel nur Preise von 2013 und 2015 aufweisen. Hierbei könnte die Tankstelle entweder neu eröffnet worden sein oder es wurde erst ab einem bestimmten Zeitpunkt der betrachtete Kraftstoff Super E5 angeboten. Die Preisentwicklung einer Tankstelle sieht folgendermaßen aus:



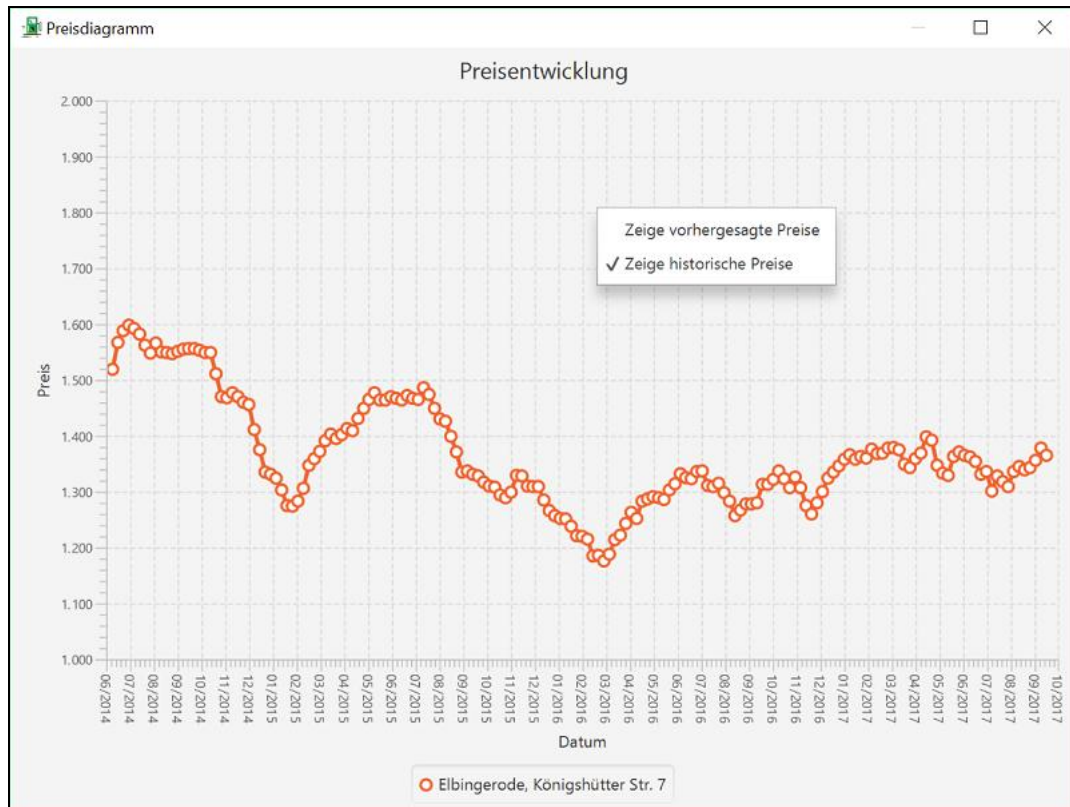
Auf der x-Achse sind die Daten zu denen die einzelnen Preise aufgenommen wurden, abgebildet. Die y-Achse enthält den Wert der Preise. Hier ist zudem zu erkennen, dass diese Tankstelle die Preise zwischen 2014 und 2017 aufweist. Um eine präzise Kurve zeichnen zu können wurde hier jeweils ein Punkt pro Woche gesetzt. Dieser Punkt entspricht dem durchschnittlichen Preis innerhalb dieser Woche.

Falls der Benutzer weitere Preisentwicklungen anzeigen lassen möchte, so kann er mehrere Preisentwicklungen in einem Fenster darstellen lassen, indem er weitere Routen- beziehungsweise Vorhersagezeitpunktelemente anklickt. Die Idee hierbei war, dass der Benutzer die Preise zwischen den einzelnen Tankstellen besser vergleichen kann. Nun folgt beispielhaft die Anzeige der Preisentwicklung mit drei Tankstellen:

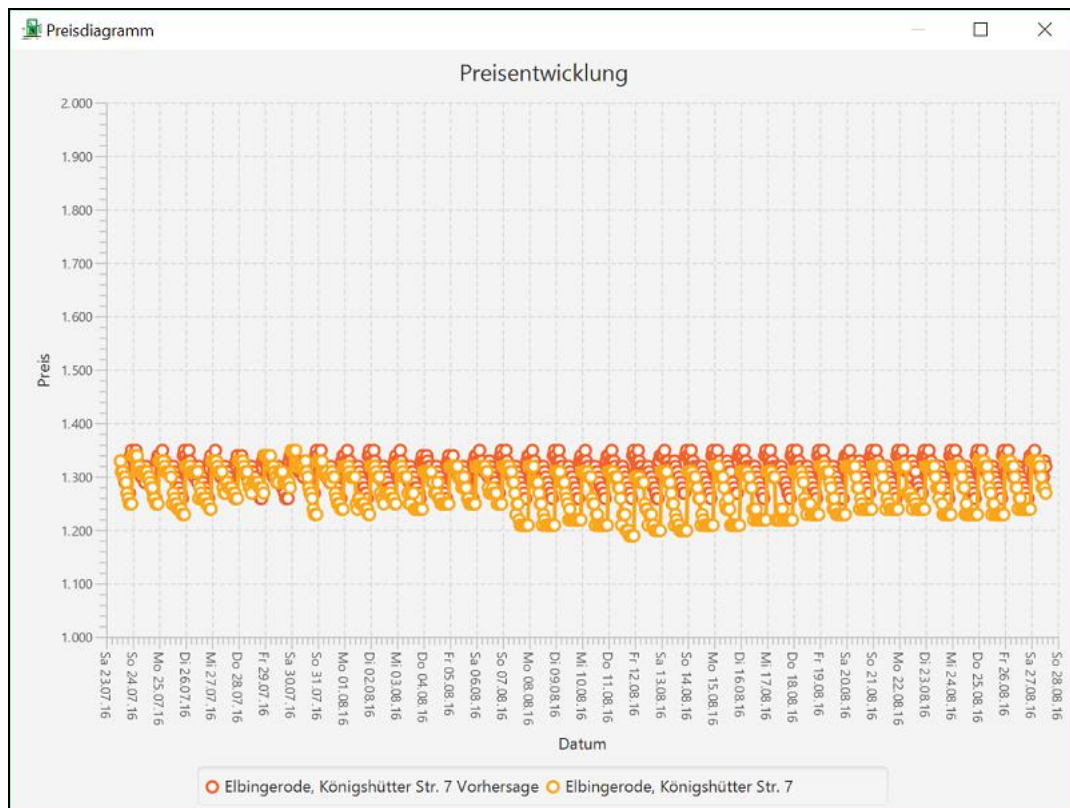


Ein weiteres Feature, das in der Preisentwicklung enthalten ist, ist die Anzeige der vorhergesagten Preise. Diese kann der Benutzer anzeigen lassen, indem er mit der rechten Maustaste auf einer beliebigen Stelle im Graphen klickt und auf „Zeige vorhergesagte Preise“ klickt. Auf diese Weise kann zwischen den historischen und vorhergesagten Preisen umgeschaltet werden.





Sobald der Benutzer auf „Zeige vorhergesagte Preise“ geklickt hat, erscheint folgende Ansicht:



Hier ist anzumerken, dass der orangene Graph die vorhergesagten Preise darstellt, wohingegen der gelbe Graph die entsprechenden historischen Preise darstellt. In dem Fall wird dem Benutzer die Genauigkeit der Vorhersage deutlich.

## Validierung – Ansicht

Ausgehend von der Routenansicht beziehungsweise der Vorhersagezeitpunktansicht kann eine Validierung geöffnet werden. Wenn diese über die Menüleiste geöffnet wird, wird diese für die Vorhersage aller Elemente der Route bzw. der Vorhersagezeitpunktliste erstellt. Es ist auch möglich die Validierung einzelner Elemente zu öffnen. Dafür muss in der Routenansicht ein mit einem Rechtsklick auf einen Tankstellennamen eine Validierung geöffnet werden. In der Vorhersagezeitpunktansicht geschieht das entsprechend mit einem Rechtsklick auf die jeweilige Zeile.

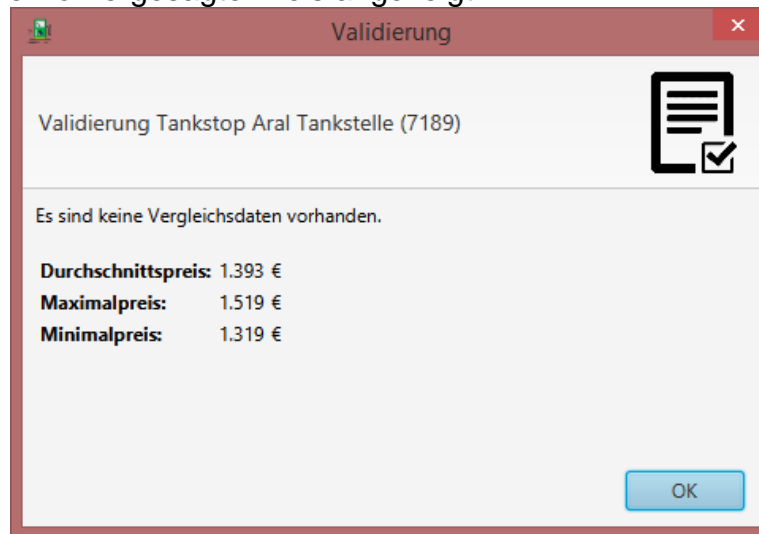
Im Folgenden wird das Fenster der Validierung erläutert:



Auf der Abbildung erkennt man die Validierung für einen einzelnen Tankstopp. Hier werden Daten zur Vorhersage des jeweiligen Tankstopps angezeigt. Entsprechend sieht die Validierung für einen Vorhersagezeitpunkt aus. Zunächst ist die Durchschnittsabweichung der Vorhersage über den gesamten Zeitraum der Vorhersage angegeben. Daneben ist die Abweichung nur an der für den Tankstopp relevanten Zeit angegeben. Das heißt der Preis dieser Tankstelle weicht über den gesamten Zeitraum um durchschnittlich knapp 3 Cent ab und der spezielle Zeitpunkt an dem getankt wurde ist exakt vorhergesagt (0 Cent Abweichung). Darunter ist die maximale Abweichung wieder erst für den gesamten Zeitraum und danach für den angefragten Zeitpunkt aufgelistet. Mit diesen Werten kann beurteilt werden wie genau exakt Preise vorhergesagt werden. Einen weiteren Anhaltspunkt dafür bieten die Preisdiagramm, die in Abschnitt [Preisentwicklung](#) beschrieben sind.

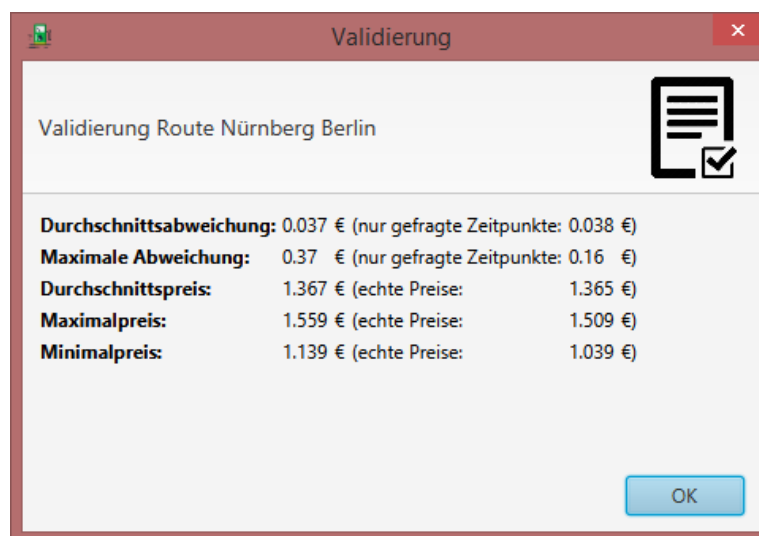
In den unteren drei Zeilen ist der Durchschnitt, das Maximum sowie das Minimum der vorhergesagten Preise im Zeitraum abgebildet. Dahinter sieht man die entsprechenden Daten aus den echten Preisdaten des Zeitraums. Mithilfe dieser Werte kann beurteilt werden wie genau der Preisrahmen vorhergesagt wurde.

Falls für den vorhergesagten Zeitraum keine Daten verfügbar sind, aber eine Vorhersage erstellt wurde, weil die Daten beispielsweise bis zum Start des Vorhersagezeitraums verfügbar waren, werden nur der minimale, maximale und durchschnittliche Vorhergesagte Preis angezeigt.



In diesem Fall wurde eine Vorhersage, die in die Zukunft reicht getätigt, sodass nur die abgebildeten Werte verfügbar sind.

Routen und Vorhersagezeitpunktlisten können ebenso validiert werden:



Hier wurde die Durchschnittsabweichung über alle enthaltenen Tankstellen und Zeiträume gemittelt. Der dahinter angegebene Wert „nur gefragte Zeitpunkte“ entspricht der mittleren Abweichung an den Zeiten, für die der Preis bestimmt werden sollte.

Wenn weder historische Daten verfügbar sind noch eine Vorhersage getätigt werden konnte, kann keine Validierung durchgeführt werden.

## Validierung

In diesem Abschnitt werden die vorhergesagten Preisdaten mit den echten Preisen verglichen. Um dies zu ermöglichen muss der vorhergesagte Abschnitt in einem zeitlichen Bereich liegen, in dem historische Preisdaten verfügbar sind. Das heißt, bei einer Route in der Zukunft ist keine Bewertung der Vorhersage möglich. Um die Güte einer Vorhersage zu betrachten wurden folgende Werte betrachtet:

- **Durchschnittliche Abweichung:** Dieser Wert gibt die durchschnittliche Abweichung der echten Preise zu den vorhergesagten Preisen.
- **Maximale Abweichung:** Dieser Wert gibt die maximale Abweichung der echten Preise zu den vorhergesagten Preisen an.
- **Durchschnittlicher Preis:** Dieser Wert entspricht dem durchschnittlichen Preis, der über den gesamten Vorhersagezeitraum von 5 Wochen vorhergesagt wurde.
- **Maximaler Preis:** Dieser Wert entspricht dem maximalen Preis, der über den gesamten Vorhersagezeitraum von 5 Wochen vorhergesagt wurde.
- **Minimaler Preis:** Dieser Wert entspricht dem minimalen Preis, der über den gesamten Vorhersagezeitraum von 5 Wochen vorhergesagt wurde.

Zusätzlich zu diesen Werten können die vorhergesagten Preise anhand eines Diagramms der Preisentwicklung untersucht sowie mit den echten Preisdaten des Zeitraums verglichen werden. Im Folgenden werden einige solcher Diagramme aufgeführt, um die Vorhersagen zu bewerten. Dabei sind einige Beispiele davon Spezialfälle, die nur sehr selten auftreten und andere sind durchschnittliche bis gute Fälle! Die gelbe Kurve steht für die echten Preise, während die orangefarbene Kurve die vorhergesagten Preise darstellt. Diagramm 1 und 3 sind nur ein Ausschnitt des vorhergesagten Zeitraums von 5 Wochen, da sich der Verlauf ähnlich weiter fortsetzt.

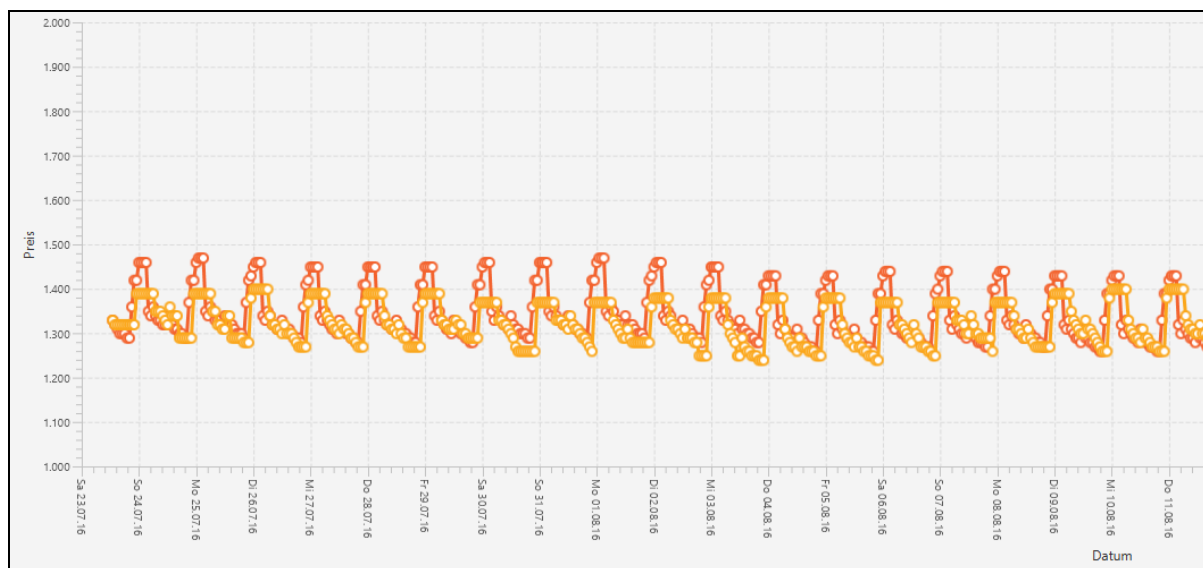


Diagramm 1: Tankstelle 4470 (SEESSEN, BORNHAUSER STR | Shell | 38723 | Seesen) (aus Route „Hildesheim Harz“)

Im Diagramm 1 erkennt man einen Fall, der häufig zu beobachten ist. Die Preise wurden relativ gut vorhergesagt. Allerdings kann man erkennen, dass die Preise in den nächtlichen Stunden relativ stark ansteigen, obwohl viele Tankstellen ihre Preise über die Nacht hinweg nicht ändern. Um diesem Problem entgegenzuwirken, wurden die Gewichte für die nächtlichen Stunden zu Blocks zusammengefasst, sodass nicht mehr jede Stunde ein eigenes Gewicht aufweist, welches durch Trainingsdaten trainiert werden muss.

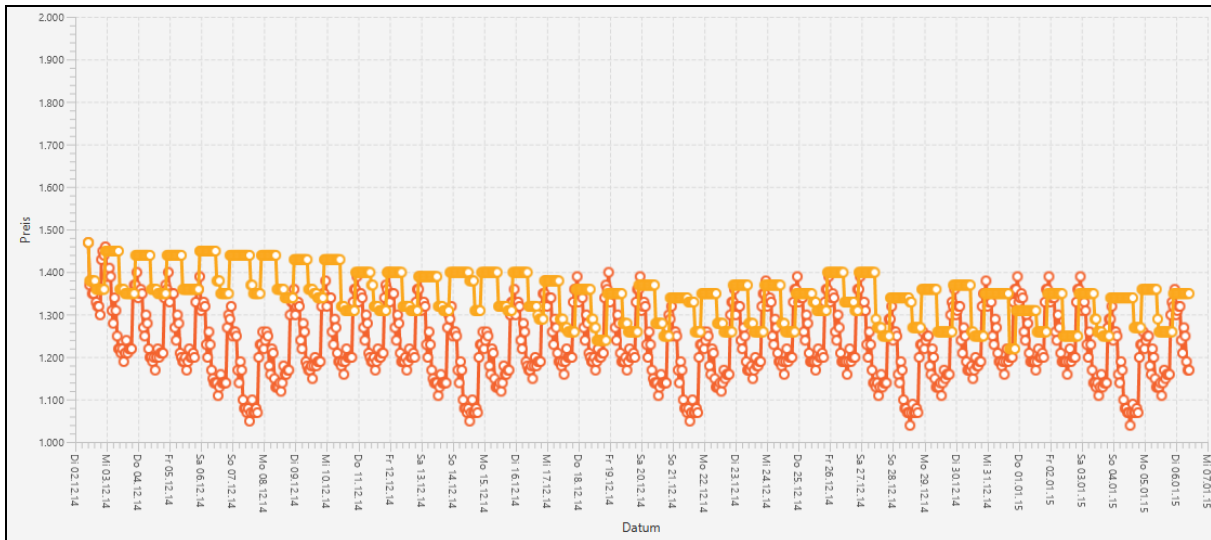


Diagramm 2: Tankstelle 1083 (TOTAL HAMBURG | TOTAL | 22761 | HAMBURG) (aus Route "Kiel Celle")

Im Diagramm 2 erkennt man, dass gelegentlich der Preistrend falsch eingeschätzt wird. Dies liegt daran, dass die Preise direkt im Zeitraum vor dieser Route abfallen, sodass angenommen wird, dass dieser Trend zunächst anhält. Es kommt auf die zugrundeliegenden Daten an, wie damit umgegangen wird. In den meisten Fällen werden die Preisdaten jedoch gut vorhergesagt, wie in Diagramm 1 und 3 zu sehen ist.

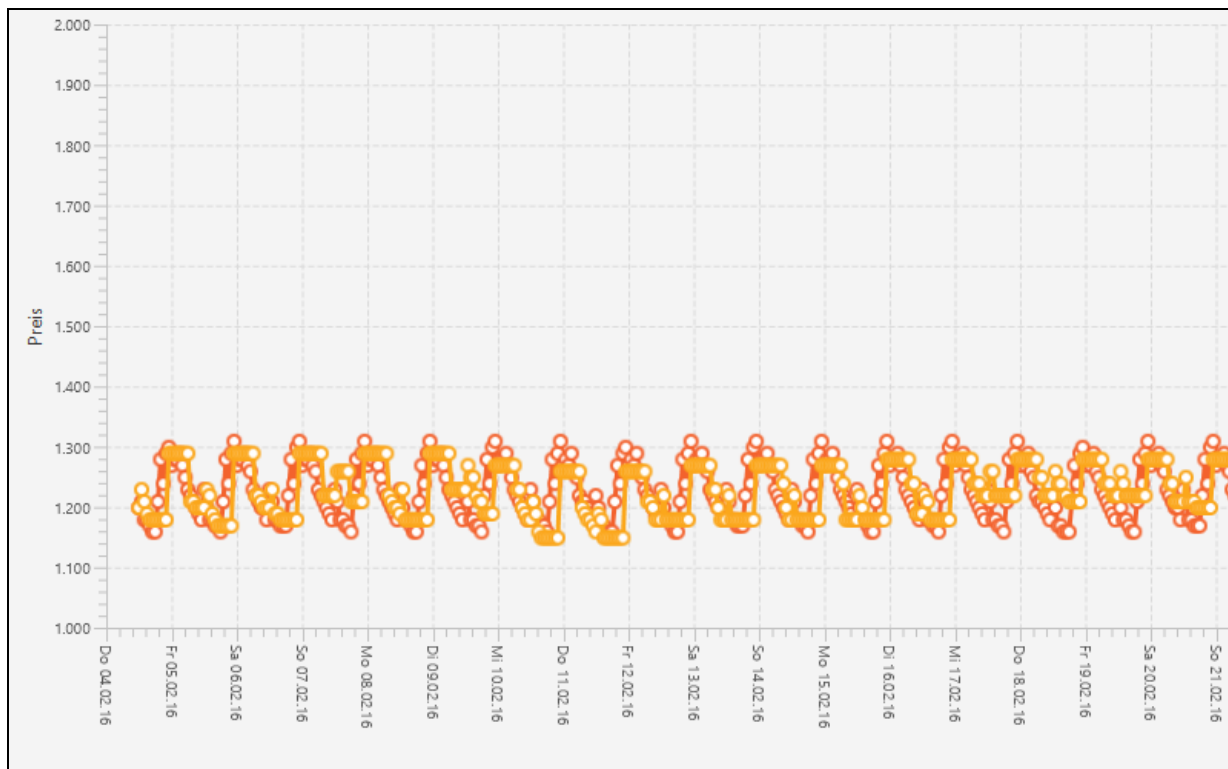


Diagramm 3: Tankstelle 8866 (Aral Tankstelle | ARAL | 36137 | Großenlütder) (aus Route Fulda Bochum")

In Diagramm 3 kann man erkennen, dass die Preise sehr genau so vorhergesagt werden, wie es in echt ebenfalls der Fall ist.

Insgesamt haben die 3 vorgestellten Tankstellen in diesem Zeitraum eine durchschnittliche Abweichung von 3,5 Cent (Diagramm 1), 12 Cent (Diagramm 2) sowie 3 Cent (Diagramm 3).

In der folgenden Tabelle sind die durchschnittlichen Abweichungen der 5 vorhergesagten Wochen für alle Tankstellen aus ausgewählten Routen zu sehen:

Route	Durchschnittliche Abweichung	Durchschnittlicher Preis (vorhergesagt/echt)	Maximaler Preis (vorhergesagt/echt)	Minimaler Preis (vorhergesagt/echt)
<b>Fulda Bochum</b>	0,047 €	1,250 € / 1,236 €	1,449 € / 1,439 €	1,049 € / 1,109 €
<b>Nürnberg Berlin</b>	0,037 €	1,367 € / 1,365 €	1,559 € / 1,509 €	1,139 € / 1,039 €
<b>Kiel Celle</b>	0,065 €	1,354 € / 1,407€	1,569 € / 1,669 €	1,219 € / 1,259 €

Man kann erkennen, dass die Preise oft sehr genau mit wenigen Cents Abweichung getroffen werden. Die durchschnittliche Abweichung einer Route liegt noch deutlich niedriger, wenn nur die Zeitpunkte betrachtet werden, an denen an einer Tankstelle getankt wurde und nicht wie in der Tabelle der gesamte Vorhersagezeitraum aller Tankstellen gemittelt wird. Die durchschnittliche Abweichung der Route Kiel - Celle an den gefragten Zeitpunkten beträgt beispielsweise nur 1,8 Cent. Auch der preisliche Rahmen im Vorhersagezeitraum konnte gut getroffen werden. So liegen die minimalen, maximalen und durchschnittlichen Preise der Vorhersagen meistens sehr nah an den echten Werten.

Die genannten Messwerte und Diagramme können innerhalb des Programms, wie im Kapitel [Betreiberhandbuch](#) beschrieben abgefragt werden.



## Troubleshooting

In diesem Abschnitt wird auf einzelne Fehlermeldungen und ihre Bedeutung eingegangen. Außerdem werden nähere Informationen zur Ursache und Behebung aufgelistet. Dieses Kapitel unterteilt sich in Warnungen und Fehlermeldungen.

### Warnungen

„Warnung 201: Fehler beim Lesen der Datei Tankstellen.csv. Möglicherweise werden Daten fehlerhaft dargestellt.“

- Diese Warnung tritt auf, wenn es Probleme gab, den Inhalt der Tankstellen.csv Datei zu lesen. Dieses Problem tritt für gewöhnlich nur auf, wenn die Datei bearbeitet wurde. Besuchen Sie erneut die Stellen in der Datei, die Sie bearbeitet haben und stellen Sie sicher, dass jede Zeile dieselbe Anzahl an Semikolons hat. Falls Sie die Datei nicht geändert haben, laden Sie die Datei erneut herunter. Alle Informationen dazu finden sie unter dem Kapitel [Installationsanleitung](#).

„Warnung 202: Für die ausgewählte Tankstelle konnte keine Vorhersage durchgeführt werden.“

- Diese Warnung tritt auf, wenn ein Preisdiagramm für eine Tankstelle geöffnet wird, in dem die vorhergesagten Preise angezeigt werden sollen, oder wenn der Anzeigemodus auf „Zeige vorhergesagte Preise“ gestellt wird und die Tankstelle keine vorhergesagten Daten besitzt, weil beispielsweise keine bzw. nicht ausreichend historische Preisdaten vorhanden sind oder die Vorhersagezeit zu weit in der Zukunft liegt.

„Warnung 203: Für eine oder mehrere der ausgewählten Tankstellen konnte keine Vorhersage durchgeführt werden“

- Diese Warnung tritt in den selben Fällen, wie die Warnung 202 auf. Allerdings wird diese Warnung angezeigt, wenn mehrere Tankstellen angezeigt werden und eine oder mehrere davon keine vorhergesagten Daten besitzen.

„Warnung 204: Die Datei Postleitzahl2Bundesland.csv wurde nicht gefunden! Den Tankstellen kann kein Bundesland zugeordnet werden. Feriendaten können bei der Vorhersage nicht benutzt werden.“

- Diese Meldung wird angezeigt, wenn sich die Datei „Postleitzahl2Bundesland.csv“ nicht am richtigen Ort befindet oder einen anderen Namen trägt. Stellen Sie sicher, dass sich die Datei im Unterordner „Eingabedaten“ des Programmordners befindet. Sollte das Problem weiterhin bestehen, versuchen Sie das Programm erneut einzurichten. Alle Informationen finden Sie dazu im Kapitel [Installationsanleitung](#). Wird diese Warnung ignoriert, könnte dies zu einer ungenaueren Vorhersage der Tankpreise führen.



„Warnung 205: Falsches Format der Datei „Postleitzahl2Bundesland.csv“. Möglicherweise werden Daten fehlerhaft dargestellt.“

- Diese Warnung wird angezeigt, wenn es ein Problem beim Lesen der Datei „Postleitzahl2Bundesland.csv“ gab. Möglicherweise wurde die Datei geändert. Um dieses Problem zu lösen, laden Sie das Programm erneut herunter und ersetzen Sie die neue Datei „Postleitzahl2Bundesland.csv“ im „Eingabedaten“ Ordner mit der fehlerhaften. Andernfalls Installieren Sie das Gesamte Programm erneut. Alle Informationen dazu finden Sie unter dem Kapitel [Installationsanleitung](#).

„Warnung 206: Feriendaten wurden nicht gefunden und können bei der Vorhersage nicht benutzt werden.“

- Diese Warnung wird angezeigt, wenn sich die Dateien zu Ferien nicht im „Eingabedaten“ Ordner befinden. Stellen Sie sicher, dass sich im „Eingabedaten“ Ordner ein Ordner namens „Ferien“ befindet. Zusätzlich sollten sich dort 8 Dateien mit Jahreszahlen wiederfinden von 2013.txt bis 2020.txt. Sollte dies nicht der Fall sein, downloaden Sie das Programm erneut und ergänzen Sie die Fehlenden Dateien. Andernfalls Installieren Sie das Gesamte Programm erneut. Alle Informationen dazu finden Sie unter dem Kapitel [Installationsanleitung](#).

„Warnung 207: Die Dateinamen der Feriendaten entsprechen nicht dem richtigen Format. Möglicherweise werden Daten fehlerhaft dargestellt.“

- Diese Warnung wird angezeigt, wenn es Probleme gab, die Dateien im Verzeichnis Eingabedateien/Ferien/ zu lesen. Dort sollten sich acht Dateien mit Jahreszahlen wiederfinden von 2013.txt bis 2020.txt und keine weiteren. Sollte dies nicht der Fall sein, downloaden Sie das Programm erneut und ergänzen Sie die Fehlenden Dateien. Andernfalls Installieren Sie das Gesamte Programm erneut. Alle Informationen dazu finden Sie unter dem Kapitel [Installationsanleitung](#).

„Warnung 208: Falsches Format der Feriendaten. Möglicherweise werden Daten fehlerhaft dargestellt.“

- Diese Warnung wird angezeigt, wenn es Probleme gab, die Dateien im Verzeichnis Eingabedateien/Ferien/ zu lesen. Das bedeutet, dass die Namen der Dateien nicht dem Format „<Jahreszahl>.txt“ entsprechen oder weitere txt-Dateien im Ordner enthalten sind. Die sicherste und schnellste Lösung ist, das Programm erneut herunterzuladen und die fehlerhaften Dateien im Verzeichnis Eingabedateien/Ferien/ zu ersetzen. Sollte dies das Problem nicht lösen, installieren Sie das Programm komplett neu. Besuchen Sie hierfür das Kapitel [Installationsanleitung](#).

## Fehlermeldungen

„Error 301: Die Datei Tankstellen.csv wurde nicht gefunden! Das Programm konnte nicht gestartet werden.“

- Diese Fehlermeldung tritt auf, wenn die Datei „Tankstellen.csv“ nicht gefunden wurde. Das Problem kann auftreten, wenn Sie sich zum Beispiel nicht am richtigen Ort befindet oder der Name der Datei nicht exakt der gleiche ist. Stellen Sie sicher, dass sich die Datei im Unterordner „Eingabedaten“ des Programmordners befindet. Sollte das Problem weiterhin bestehen, versuchen Sie das Programm erneut einzurichten. Alle Informationen finden sie dazu im Kapitel [Installationsanleitung](#).

„Error 302: Die ausgewählte Route konnte nicht geladen werden. Datei möglicherweise fehlerhaft oder nicht mehr vorhanden.“

- Wenn die Routendatei, die geladen werden sollte, nicht vorhanden ist bzw. die Formatierung des Dateiinhalts fehlerhaft ist, wird diese Meldung angezeigt. Stellen Sie sicher, dass die Routendatei entsprechend der Beschreibung in Abschnitt [Schnittstelle](#) formatiert ist.

„Error 303: Vorhersagezeitpunkte konnten nicht geladen werden. Die Datei ist möglicherweise fehlerhaft oder nicht mehr vorhanden.“

- Wenn die Datei mit Vorhersagezeitpunkten, die geladen werden sollte, nicht vorhanden ist bzw. die Formatierung des Dateiinhalts fehlerhaft ist, wird diese Meldung angezeigt. Stellen Sie sicher, dass die Vorhersagezeitpunktdatei entsprechend der Beschreibung in Abschnitt [Schnittstelle](#) formatiert ist.

„Error 305: Die historischen Benzinpreise wurden nicht gefunden. Es kann keine Vorhersage getätigt werden.“

- Diese Fehlermeldung wird angezeigt, wenn der Ordner Benzinpreise fehlt oder leer ist. Stellen Sie sicher, dass sich dieser Ordner im „Eingabedaten“ Ordner befindet und richtig benannt wurde. Sollte dies nicht der Fall sein, wurden die Daten nicht eingefügt. Besuchen Sie hierfür das Kapitel [Installationsanleitung](#) für alle Informationen.

„Error 306: Die Preise konnten für keinen Tankstopp innerhalb der Route vorhergesagt werden. Möglicherweise konnten keine Preise importiert werden oder die historischen Daten liegen zu weit zurück.“

- Dies ist ein Problem, das bei selbst erstellten Routen auftreten kann. Es gibt Tankstellen, für die nicht ausreichend Daten vorhanden sind. Wählen Sie eine andere Route oder besuchen Sie andere Tankstellen, für die mehr Daten zur Verfügung stehen, um dieses Problem zu lösen.

„Error 307: Die Entfernung der Tankstopps in der importierten Route ist zu groß für die gewählte Tankkapazität.“

- Dieses Problem kann bei selbst erstellten Routen auftauchen. Wenn es Strecken innerhalb der Route gibt, die selbst mit vollem Tank nicht überwunden werden können, wird diese Fehlermeldung ausgegeben. Um das Problem zu beheben, gehen Sie in die Datei, die die von Ihnen aufgerufene Route enthält. Oben finden Sie die Kapazität des Tanks. Wählen Sie einen größeren Wert. Dies sollte das Problem beheben.

## Ausblick

Die Anforderungen des Projekts waren sehr knappgehalten, wodurch es sehr viel Spielraum in Bezug auf Weiterentwicklung gibt. Da wir unseren Schwerpunkt auf die „Visibility“ gelegt haben, kann das Projekt bezogen auf die „Präzision“ verbessert werden. Hier gibt es aufgrund der minimal beschriebenen Anforderungen viele Punkte, wie das Ergebnis einer Route präziser werden kann. Zum Beispiel wird zurzeit jede Route per „Luftlinie“ befahren. Hier könnte die „Präzision“ deutlich erhöht werden, wenn die Route mithilfe von „Google Maps“ befahren wird. So würden die Zahlen zur Gesamtstrecke, zum Verbrauch und zu den Fahrtkosten realistischer sein. Ein weiterer Faktor für die „Präzision“ der Ergebnisse ist die Geschwindigkeit. Da jede Strecke nicht mit derselben Geschwindigkeit befahren wird, würde dieser Faktor einen Einfluss zum Verbrauch haben. Daher könnte das Programm mit einem Mindest- und Höchstgeschwindigkeits-Feature der Strecken verbessert werden. Dazu müsste zudem der Benzinverbrauch angepasst werden.

Außerdem kann unsere Vorhersage auch auf andere Bereiche mit gleichen beziehungsweise ähnlichen Eingabegrößen ausgeweitet respektive umgestellt werden. So können beispielsweise die Preise für alle Kraftstoffe vorhergesagt werden und der Nutzer kann entscheiden welcher der für ihn interessante ist.

Die Tankstrategie lässt sich auch auf das Tanken von Strom für Elektroautos beziehen, da die Preise von Ladestation zu Ladestation anders sind und teilweise sogar kostenlos geladen werden kann. So kann mit dem Verfahren auch die günstigste Ladestrategie für eine Route gestimmt werden.

Ein weiteres Anwendungsgebiet für die von uns entwickelte Vorhersage könnte die Vorhersage von Flugpreisen sein, da sich auch diese abhängig von der Zeit und beispielsweise von Ferienzeiten verändern.

Letzten Endes wurde das Projekt so kreiert, dass es für viele Neuigkeiten offen ist. Man kann als Entwickler seiner Fantasie freien Lauf lassen.