



**HTL anichstraße**

---

# **ETF-Backtesting**

SWP-Projekt

Betreuer: Rubner

Zeitraum: September bis Jänner/Feber

Mitarbeiter: Florian Staudacher

Klasse: 5AHWII 2020/21

# Einleitung

## Ziel dieses Dokuments

Bei diesem Projekt geht es darum einen ETF (TLT) über die letzten 10-20 Jahre nach einer bestimmten Strategie backzutesten.

**TLT:** Staatsanleihen (T-Bonds) sind von der US-Bundesregierung ausgegebene Staatsanleihen mit einer Laufzeit von mehr als 20 Jahren. T-Bonds wird bis zur Fälligkeit regelmäßig verzinst. Zu diesem Zeitpunkt erhält der Eigentümer auch einen Nennbetrag in Höhe des Kapitals.

**Strategie:** Als erstes tritt die Frage auf wie wir an das Traden rangehen. Hier verwenden wir eine Strategie, bei der man jeden Freitag (oder Donnerstag falls Freitag börsenfrei ist) die maximale Anzahl an Positionen kauft und diese dann am nächsten Börsentag wiederverkauft.

**Dokumentation:** Der Verlauf des ganzen Projekts sollte auf Github dokumentiert werden.

### Bonus:

- I. Erfassen wann Zeittechnisch am Abend der Kauf am ehesten gelingt, also wann am abend der beste Kaufzeitpunkt ist.
- II. Stop lose bei verschiedenen % einbauen

## Zweck dieses Dokuments

- Die Trading Startegie verstehen und anwenden lernen
- Die Trading Strategie nach ihrer Sinnhaftigkeit überprüfen
- Herausfinden ob es eine zeitliche Abhängigkeit für den Optimalenkaufpunkt gibt

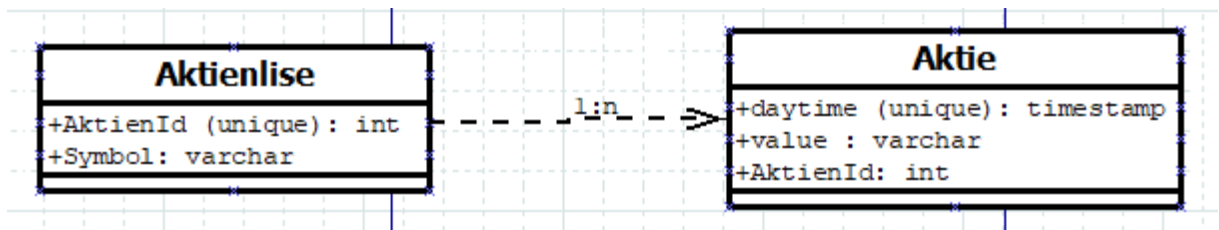
## Programmlicher Ablauf

### Basis

1. Daten per CSV oder API importieren und in brauchbare Datenstruktur abspeichern
2. Datenstruktur so erweitern, dass die Tradingstrategie auf sie angewendet werden kann, das heißt den jeweiligen Tag ermitteln (Freitag, Donnerstag etc)
3. Trading Strategie anwenden, das heißt überprüfen ob der letzte Börsenaktive Tag einer Woche ist, an diesem Tag wird dann gekauft buyStock() und am nächsten Börsenaktiven Tag der nächsten Woche wird wieder verkauft sellStock(). Das Ganze wird mit dem closingValue des Stocks durchgeführt
4. Als nächster Punkt kommt es zur Visualisierung, diese wird in JavaFX erstellt und es handelt sich dabei um 3 verschiedene Arten die Daten darzustellen, diese 3 Darstellungsarten werden dann in ein Flowpane eingetragen.
  - 4.1 Liniendiagramm, das die Darstellung der Trading Strategie an einem Beispieldepot den Verlauf der Effizienz der Tradingstrategie darstellen soll.
  - 4.2 Balkendiagramm, das die Chance für positive oder Negative Trades darstellen soll.
  - 4.3 Tabelle, diese dient zur Darstellung der Trading Historie, dabei sollten positive Werte mit der Farbe Grün hervorgehoben werden und negative mit der Farbe Rot.

### Bonus

1. Über die Alphavantage API sollen Daten einer bestimmten Aktie in einem bestimmten Zeitintervall ausgelesen werden. Diese Daten werden somit in mehreren Csv files gespeichert.
2. Durch ein Datenbankschema sollten danach die files abgesichert werden sodass bei bereits verwendeten Aktien der Zugriff auf das Programm auch ohne Internet gewährleistet werden kann.



3. Eine Connection zu dieser Datenbank sowie Klassen für die jeweiligen Tabellen werden in Java erstellt.
4. Methoden zum Abruf der wichtgen Daten, sowie erstellen weiterer Tabellen und das Befüllen der Tabellen aus den CSV files werden implementiert.
5. Minimalwert der ausgewählten Aktie ermitteln
6. Häufigkeit der Minimalwerte zum jeweiligen Zeitpunkt graphisch via Barchart darstellen.

7. API auf das Basisprojekt anwenden, sodass die Trading Strategie auf jede Aktie angewendet werden kann.
8. Statt input über console einen grafischen Input erstellen

## Technologien

Hier eine Auflistung der Technologien, welche für die Umsetzung benötigt werden.

- Als Programmiersprache wird JAVA verwendet
- Entwicklungsumgebung ist die Eclipse Enterprise Edition
- Graphic Bibliothek stellt JAVA FX dar
- CSV Datei aktuell direkt von der NASDAQ verwendet (Daten), wird im Bonus Teil zu einer API geändert
- Alpha Vantage
- Apache Commons library

## Meilensteine

<b>Vorbereitungsphase</b>	
Freigabe Pflichtenheft	28.09.2020
Start der Umsetzung	30.09.2020
<b>Implementierung des Codes der Basiseigenschaften des Projekts</b>	
Importieren der Daten und Abspeicherung in brauchbare Datenstruktur	07.10.2020
Erweiterung der Datenstruktur um Trading Startpunkt	14.10.2020
Implementierung der Tradingstrategie im Programm	21.10.2020
Visualisierung eines Beispieldepots via Linechart	05.11.2020
Visualisierung der Erfolgchance via Barchart	12.11.2020
Visualisierung der Trading Historie via Tabelle	19.11.2020
<b>Bonus</b>	
Änderung des Datenbezugs von CSV zu API	02.12.2020
Implementierung eines Stoplose Mechanismus	09.12.2020
Visualisierung eines Wolkendiagramms für optimalen Kauf/- Verkaufszeitpunkt	23.12.2020
<b>Test und Verbesserung</b>	
Test und Analyse	04.01.2021
Verbesserung des Codes	13.01.2021
<b>Vermutliche Abgabe des Projekts</b>	20.01.2021