

Noctua Benutzerhandbuch

Gabriel Pawlowsky, Josef Sochovsky

18. April 2013

Inhaltsverzeichnis

1	Einleitung	3
2	Installation	4
2.1	Start der Installation	4
3	Programm	7
3.1	Schreiben eines Algorithmus	7
3.1.1	Erzeugen eines Algorithmus-Files	8
3.1.2	Entscheidungen	10
3.2	Historische Daten	10
3.3	Backtestingsoftware	11
4	minimale Systemvoraussetzungen	17

KAPITEL 1

Einleitung

Im Verlauf des Maturaprojekts des 5. Jahrgangs der Abteilung IT am TGM wurde von dem Projektteam Nagy Peer, Pawlowsky Gabriel und Sochovsky Josef unter der Betreuung von Prof. Mag. Hans Brabenetz und Dr. Helmut Vana eine Software geschrieben. Der Sinn und Zweck dieser Software ist das Testen eines Algorithmus über eine Menge an gegebenen Daten. Ein solcher Algorithmus ist eine automatisierte Entscheidungslogik, anhand von historischen Aktiendaten soll dieser Entscheidungen generieren. Eine Entscheidung kann kaufen und verkaufen.

Die Software zeigt Qualitätsmerkmale des Algorithmus über die gegebenen Daten, zeigt diese an und simuliert somit einen Börsenablauf.

KAPITEL 2

Installation

2.1 Start der Installation

Die Installation wird durch das ausführen der setup.exe gestartet. Danach öffnet sich der Installer.

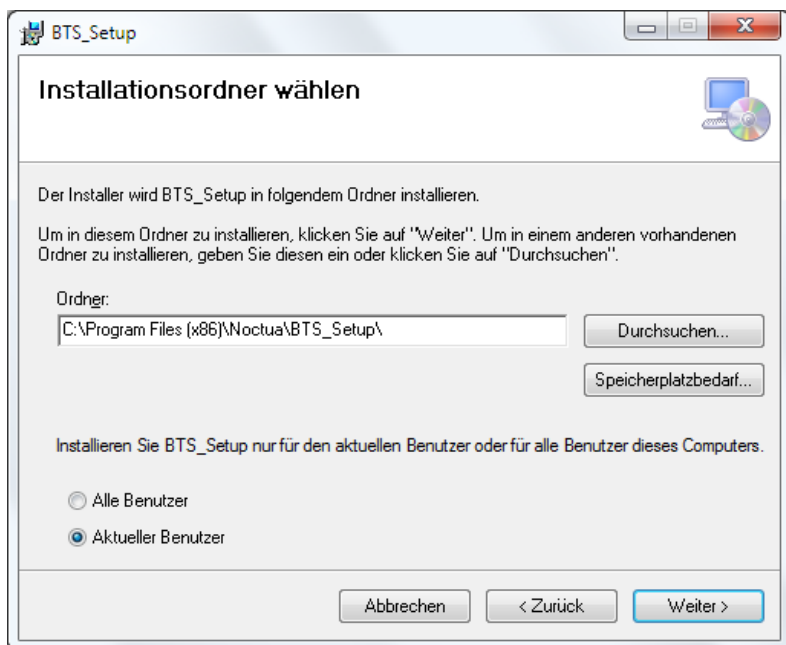


Abbildung 2.1: Konfigurieren der Installtion

Nach der korrekten Konfiguration des Setups kann die Installation gestartet werden.

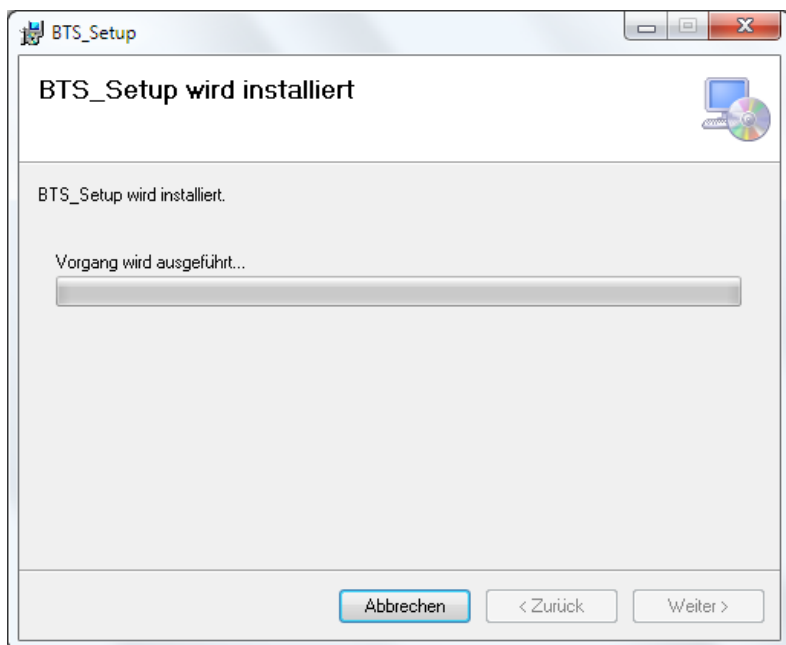


Abbildung 2.2: Installationsvorgang

Nach diesem Vorgang ist die Backtesting-Software vollständig installiert und ausführbar.

KAPITEL 3

Programm

3.1 Schreiben eines Algorithmus

Um die Software zu bedienen wird ein in F-Sharp geschriebener Algorithmus benötigt. Dieser muss als DLL-Datei zur Laufzeit in die Software eingebunden werden. Eine solche Datei wird mittels Microsoft Visual Studio erzeugt werden, damit die Software funktionieren soll.

Folgende Eigenschaften muss die Datei erfüllen:

- Name der Methode: `startCalculation`
- Übergabeparameter 1: eine Liste aller historischen Daten
- Übergabeparameter 2: eine Liste der Signale für die Entscheidungen (bei Übergabe ist diese Liste leer)

Mit dem Rückgabewert der Methode wird nicht gearbeitet, sondern mit der vom Algorithmus verarbeiteten Signalliste.

Listing 3.1: Datentyp des 1. Übergabeparameters

```
System.Collections.Generic.List<System.Tuple  
<System.DateTime, decimal, decimal, decimal, decimal>>
```

Listing 3.2: Datentyp des 2. Übergabeparameters

```
System.Collections.Generic.List<int>
```

Außerdem muss sich der Algorithmus im Namespace „Algorithm“ und im Modul „DecisionCalculator“ befinden.

3.1.1 Erzeugen eines Algorithmus-Files

Öffnen Sie Microsoft Visual Studio und erzeugen sie ein neues Projekt.

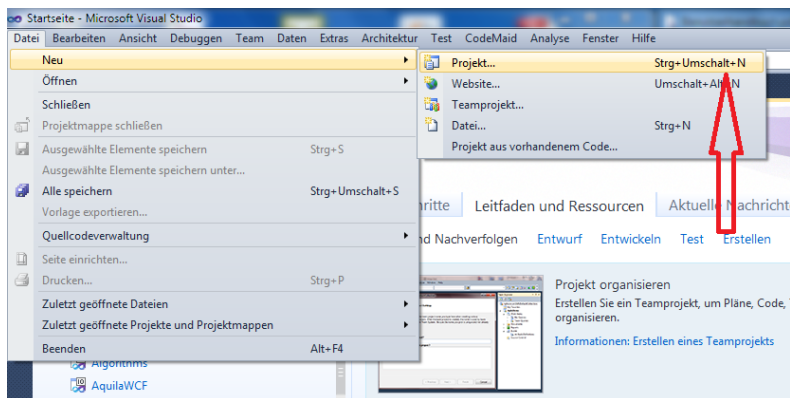


Abbildung 3.1: Erzeugen eines neuen Projekts

Erstellen Sie es als F-Sharp-Bibliothek.

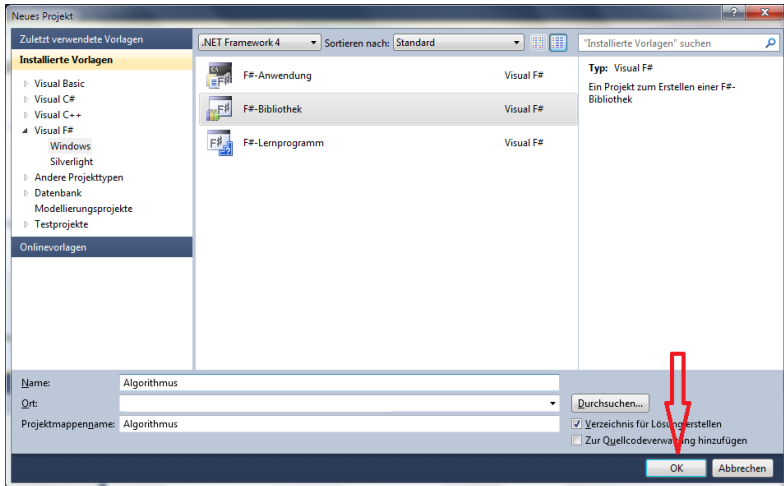


Abbildung 3.2: Erstellen Sie es als F-Sharp-Bibliothek

Halten Sie sich an die, im Kapitel „Schreiben eines Algorithmus“, genannten Richtlinien und implementieren die Methode `startCalculation` und kompilieren sie das Projekt.

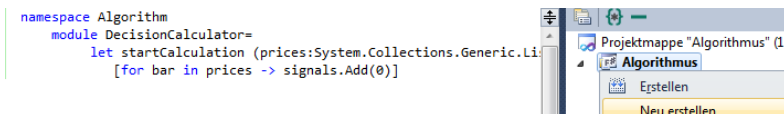


Abbildung 3.3: Kompilieren des Projekts

3.1.2 Entscheidungen

Folgende Entscheidungen sind zulässig:

3	starkes Kaufssignal
2	mittleres Kaufssignal
1	schwaches Kaufssignal
0	Alle Bestände werden gekauft oder verkauft
-1	schwaches Verkaufssignal
-2	mittleres Verkaufssignal
-3	starkes Verkaufssignal

3.2 Historische Daten

Um einen Algorithmus mit der Backtesting-Software testen zu können, müssen auch noch historische Aktien-Preisdaten in die Software eingebunden werden, über die der Algorithmus getestet wird. Diese können in Form einer Comma-separated values-Datei gespeichert und dessen Pfad in der Backtesting-Software angegeben werden. Dabei wurde ein sehr weit verbreitetes Format benutzt, dass bspw. auch jegliche Software des renomierten Aktiendatenbereitstellungs-Unternehmens eSignal exportieren kann. Außerdem werden in dieser Datei die historischen Daten in Form von Bars über einen bestimmten Zeitraum (z.B. Daily-Bar oder Minute-Bar) erwartet und nicht als einzelne Preiswerte. Das Format sieht in etwa so aus:

Listing 3.3: Aufbau der Comma-separated values-Datei

```
Bar,Date,Time,Open,High,Low,Close  
1,01/02/90,00:00,8.8125,9.375, 8.75,9.3125  
2,01/03/90,00:00,9.375, 9.50,9.375,9.375
```

3,01/04/90,00:00,9.375,9.6875,9.3125,9.40625

...

Zuerst steht also die Nummer des Bars, die allerdings nicht berücksichtigt wird. Darauf folgt das Datum in der Form MM/DD/YY und die Uhrzeit in der Form hh:mm. Zu guter letzt kommen nun nur noch die Werte Open, High, Low und Close des Bars. Diese Datei kann unendlich lange gemacht werden, es können also unendlich viele Bars noch nach unten hin ergänzt werden. Die erste Zeile der Datei wird allgemein nicht berücksichtigt, da diese meist die Überschriften beinhaltet. Sollte sich in der ersten Zeile also ein Bar befinden wird dieser ebenfalls ignoriert.

3.3 Backtestingsoftware

Die Backtesting-Software ist nun also die Kernsoftware, mit der die Algorithmen über Daten-Files getestet werden können. Dazu sollten zuerst die allgemeinen Einstellungen getätigt werden. Diese befinden sich im Settings-Tab unter "Orders" und sehen so aus:

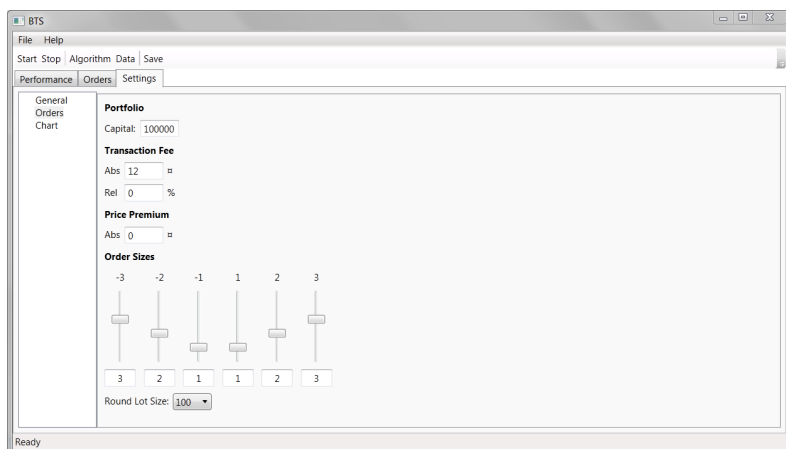


Abbildung 3.4: Order-Settings der Backtesting-Software

Hier kann zuerst das gewünschte Kapital eingestellt werden, dessen Handel die Backtesting-Software simulieren soll. Danach können Transaktionsgebühren absolut oder relativ, ein Preisaufschlag für Käufe und die Order-Größen eingestellt werden. Die Order-Größen geben an, wie viele Round Lots bei einem Signal von -3 bis +3 gekauft/verkauft werden sollen. Ein Round Lot ist hierbei die kleinste über einen Online-Broker erwerbende Menge an Aktien, die ebenfalls variieren und daher eingestellt werden kann.

Auf der Seite darunter, der "Chart"-Seite, können verschiedene Indikatoren aus der Combobox ausgewählt und hinzugefügt werden, die dann nach der Performancemessung in die Grafik im Orders-Tab eingezeichnet werden. Hier können außerdem

für jeden Indikator die entsprechenden notwendigen Parameter und Farben ausgewählt werden.

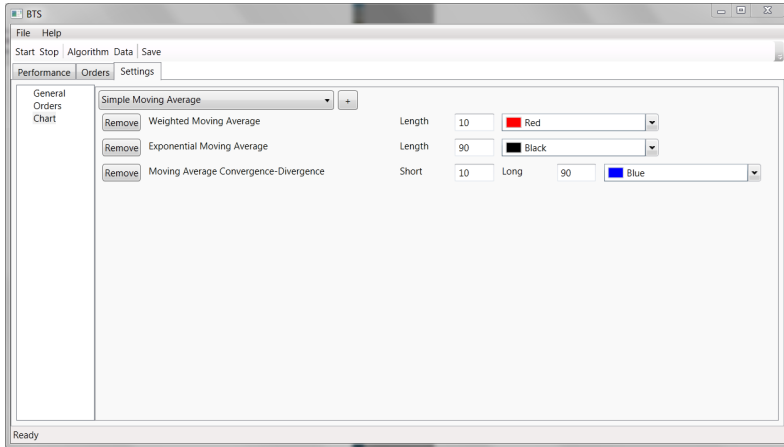


Abbildung 3.5: Chart-Settings der Backtesting-Software

Weiters können nun unter der Seite "General" die eigentlich wichtigsten Informationen festgelegt werden. Diese sind die Pfade zur Algorithmus- und zur Daten-Datei, die in den vorherigen Punkten erläutert wurden. Diese Pfade können übrigens auch durch den "Algorithm"- und den "Data"-Button in der Menüleiste hinzugefügt werden. Darunter kann noch eine Zeitspanne ausgewählt werden. Der Algorithmus wird dann nur über alle Daten getestet, die im Daten-File im gewählten Zeitraum vorkommen.

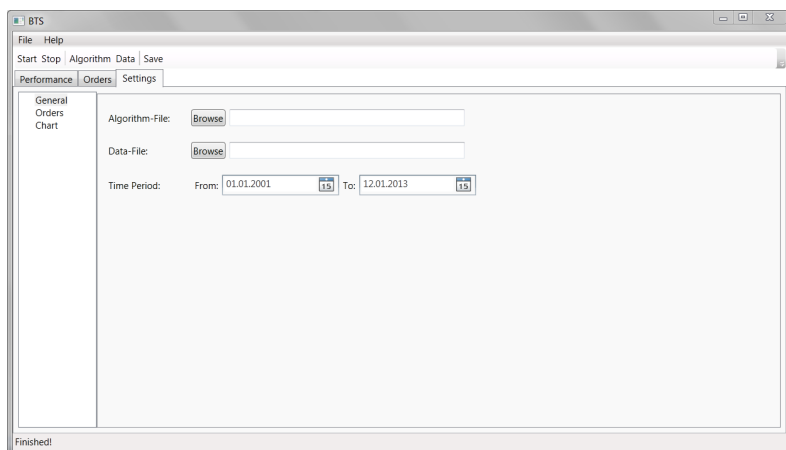


Abbildung 3.6: General-Settings der Backtesting-Software

Nachdem alle Einstellungen getroffen wurden, kann der Test gestartet werden.

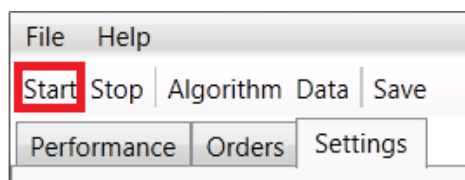


Abbildung 3.7: Start-Button der Backtesting-Software

Anschließend werden auf dem Performance-Tab der Backtesting-Software die allgemeinen Performance-Daten des Algorithmus über das spezifizierte Daten-File ausgegeben. Nähere Erklärungen zu diesen Werten können durch Halten der Maus über einen der Texte als Tooltip angezeigt werden.

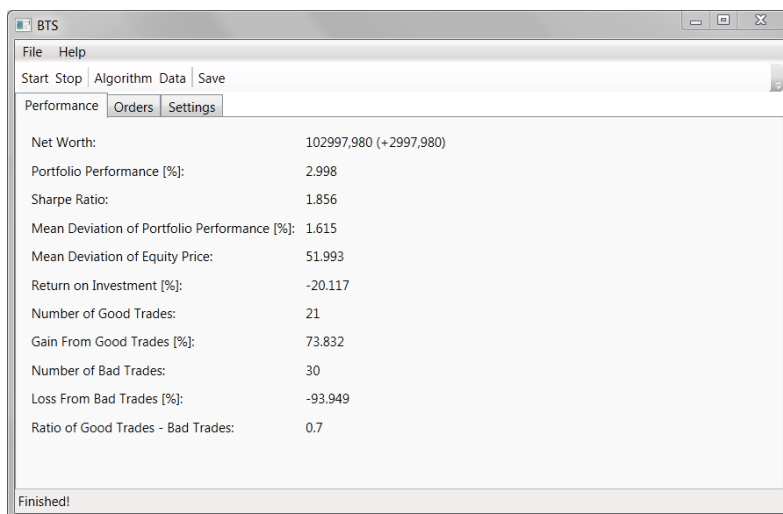


Abbildung 3.8: Start-Button der Backtesting-Software

Die meisten Informationen über den Verlauf des Tests können nun auf dem Orders-Tab gefunden werden. Hier wird zuerst oben ein Chart der Aktien-Preisdaten mit allen gewählten Indikatoren angezeigt. Durch Rechtsklick mit der Maus kann im Chart hinausgezoomt und mit der linken Maustaste hineingezoomt werden. Die Pfeile im Chart zeigen an, zu welchen Zeitpunkten der Algorithmus unter realen Bedingungen über den gewählten Zeitabschnitt Signale ausgegeben hätte. Die unterschiedlichen Farben und Farbschattierungen zeigen dabei ein Kauf- (Grün) oder ein Verkauf-Signal (Rot) und dessen Stärke an.

In der unteren Hälfte des Bildschirms wird zu jedem dieser Signale ausgegeben welchen Effekt das jeweilige Signal auf das Kapital hätte und welcher Gewinn oder Verlust dadurch zum

damaligen Preis entstanden wäre. Position bzw. der Transaction Price geben dabei an, wieviele Round Lots durch dieses Signal gekauft worden wären und wieviel die Umsetzung des Signals kosten würde. Gain/Loss bezieht sich auf die prozentuelle Veränderung des Investitionskapitals der expliziten Order und Portfolio Performance auf die prozentuelle bzw. absolute Veränderung des gesamten gewählten Kapitals. All diese Werte werden auch kumulativ, also aufaddiert, angezeigt.



Abbildung 3.9: Start-Button der Backtesting-Software

Zu guter Letzt können unter "File" -> "Export" die berechneten Performance-Daten lesbar als Text-File (.txt) exportiert werden. Außerdem kann der aktuelle Zustand der Backtesting-Software inklusive aller Settings und berechneter Daten außer der Charts (da sonst die Aktiendaten mitgespeichert werden müssten) gespeichert und zu einem späteren Zeitpunkt erneut geladen werden.

KAPITEL 4

minimale Systemvoraussetzungen

Folgende Voraussetzungen müssen vom System erfüllt werden:

Hardwareanforderungen

Prozessor	1 GHz
RAM	512 MB
Festplattenspeicher (32 Bit)	850 MB
Festplattenspeicher (64 Bit)	2 GB

Betriebssystemanforderungen

Unterstützte Clientbetriebssysteme sind:

- Windows 8 (32-Bit- und 64-Bit-Editionen)
- Windows 7 (32-Bit- und 64-Bit-Editionen)
- Windows Vista (32-Bit- und 64-Bit-Editionen)

Backtesting-Software Als BTS wird die im Laufe dieses Projekts erstellte Backtesting-Software bezeichnet, die einen Algorithmus in Form einer DLL-Datei auf seine Performance über historische Aktien-Preisdaten testen kann.. 10–16, 20

Bar Im Aktienhandel versteht man unter einem Bar die Zusammenfassung aller Preisänderungen eines Instruments über einen festgelegten Zeitraum. Ein Bar wird durch die Werte Open (erster Wert), High (höchster Wert), Low (niedrigster Wert), Close (letzter Wert) innerhalb eines bestimmten Zeitraums definiert.. 10, 11

Comma-separated values Das Dateiformat CSV steht für englisch Comma-separated values (seltener Character-separated values) und beschreibt den Aufbau einer Textdatei zur

Speicherung oder zum Austausch einfach strukturierter Daten. Die Dateinamenserweiterung lautet .csv.. 10

DLL Dynamic Link Library (DLL) bezeichnet allgemein eine dynamische Programmbibliothek, meist bezieht sich der Begriff jedoch auf die von Microsoft für die Betriebssysteme Microsoft Windows und OS/2 verwendete Variante..

7

F-Sharp F# ist eine funktionale Programmiersprache von Microsoft für das .NET-Framework.. 7, 9, 20

Abbildungsverzeichnis

2.1	Konfigurieren der Installtion	5
2.2	Installationsvorgang	6
3.1	Erzeugen eines neuen Projekts	8
3.2	Erstellen Sie es als F-Sharp-Bibliothek	9
3.3	Kompilieren des Projekts	9
3.4	Order-Settings der Backtesting-Software	12
3.5	Chart-Settings der Backtesting-Software	13
3.6	General-Settings der Backtesting-Software	14
3.7	Start-Button der Backtesting-Software	14
3.8	Start-Button der Backtesting-Software	15
3.9	Start-Button der Backtesting-Software	16

Listings

3.1	Dateityp des 1. Übergabeparameters	8
3.2	Dateityp des 2. Übergabeparameters	8