

Gattringer, Gschwantner, Krottendorfer, Elpel

Projekt Börse

# Übersicht der Klassen

## Grundsätzliche Einteilung

* Client-GUI
  + JUhr
  + JSellPanel
  + JBuyPanel
  + JStart
  + Client
* Server-GUI
  + Server
  + JSpieler
* Server
  + Server
* Common
  + ExchangeRate
  + Game
  + HttpUtil
  + Item
  + Order
  + Player
  + Round
  + Stock

## Client-GUI

### JUhr

Diese Klasse ist für die Rundenzeiten zuständig und wird in der Klasse *Client* aufgerufen.

### JBuyPanel

Diese Klasse zeichnet ein JPanel für jeden Kaufen-Dialog und wird in der Klasse *Client* aufgerufen.

### JSellPanel

Diese Klasse zeichnet ein JPanel für jeden Verkauf-Dialog und wird in der Klasse *Client* aufgerufen.

### JStart

Diese Klasse zeichnet ein JFrame zum Verbinden mit dem Server und wird in der Klasse *Client* aufgerufen.

### Client

Diese Klasse zeichnet ein JFrame welches sozusagen das Spielbrett ist. Sie ruft alle anderen Klassen der Gruppe Client-GUI auf.

## Server-GUI

### Server

Diese Klasse zeichnet ein JFrame welcher 2 Tabs beinhaltet. Ein Tab beinhaltet eine Liste aller Waren und der andere Tab n-JSpieler Objekte.

### JSpieler

Diese Klasse zeichnet ein JPanel für jeden verbundenen Spieler. Sie wird in der Klasse *Server* aufgerufen.

## Server

### Server

Diese Klasse implementiert den gesamten Server. Der Server ist „passiv“ und reagiert lediglich auf Anfragen am Port 8000. Die Daten werden im JSON-Format übertragen. Wenn Daten an den Server übergeben werden geschieht dies mittels HTTP-POST, wenn Daten vom Server geholt werden geschieht dies mittels HTTP-GET.

## Common

### ExchangeRate

Diese Klasse berechnet den aktuellen Börsenkurs für jede Ware welche im Umlauf ist. Sie wird in der Klasse *Game* aufgerufen.

### Game

Diese Klasse stellt ein Spiel (Game) dar. Auf einem Server können n Spiele laufen. Diese Klasse ruft alle anderen Klassen der Gruppe Common auf.

### HttpUtil

Diese Klasse ist für die Verbindungen zum Server zuständig. Sie wird in der Klasse *Game* aufgerufen.

### Item

Diese Klasse stellt eine Ware (Item) dar. Sie wird in der Klasse *Game* aufgerufen.

### Order

Diese Klasse stellt eine Auftrag (Order) dar welcher von einem Spieler eingetragen wurde. Sie wird in der Klasse *Game* aufgerufen.

### Player

Diese Klasse stellt einen Spieler (Player) dar. Sie wird in der Klasse *Game* aufgerufen.

### Round

Diese Klasse stellt eine Runde (Round) dar. Sie wird in der Klasse *Game* aufgerufen.

### Stock

Diese Klasse stellt das Lager (Stock) einer Ware eines Spielers dar. Sie wird in der Klasse *Game* aufgerufen.

# 

# Simulation starten

## Als Client

Öffnen Sie die Datei „Client.jar“ mit einem Doppelklick.

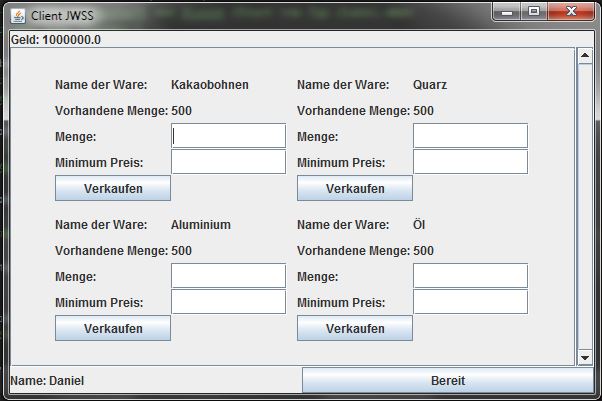
Es öffnet sich folgendes Fenster:



Geben Sie nun einen Namen ein. Das Feld **Server** können Sie leer lassen wenn Sie die gesamte Simulation (alle Clients + Server) auf Ihrem Rechner laufen lassen wollen, ansonsten fragen Sie bitte den Host des Servers um die Adresse und tragen Sie diese ein.

Im folgenden Bild sehen Sie ein Beispiel eines ausgefüllten Start-Fensters:



Nun befinden Sie sich im Spielfenster:

## Als Host

Öffnen Sie die Datei „Server.jar“ mit einem Doppelklick.

Nun müssen Sie Ihre IP-Adresse feststellen. Wir bitten Sie um Verständnis dass wir diesen Vorgang nicht weiters beschreiben.

Ihre Serveradresse setzt sich nun wie folgt zusammen:

http://10.0.0.2:8000

10.0.0.2 stellt die IP-Adresse dar. Bitte ersetzen Sie diese mit Ihrer IP-Adresse.

Wenn Sie eine Liste aller verfügbaren Waren sehen möchten oder wissen möchten welche und wie viele Spieler mit Ihrem Server verbunden sind öffnen Sie bitte die Datei „ServerGUI.jar“ mit einem Doppelklick.

Sie werden nun folgendes Fenster sehen:

