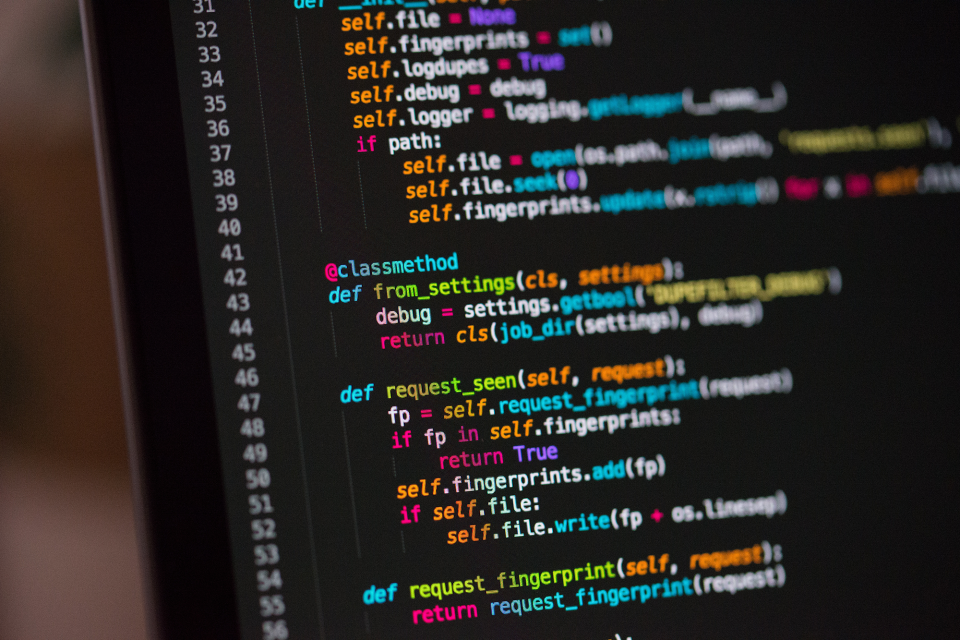
## 



FirmenParkplatz Projekt

19.06.2020

**─**

Joshua Kuck, Lukas Kirmaier, Yannik Werner, Elias Achtnich und Malte Konze

Schuljahr: 2020

Klasse: 12 - 5 Pi Leistungskurs

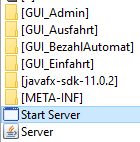
Fach: Prakt. Informatik eGK

Version: 1.0

Auftraggeber: HEMS, J. Mang

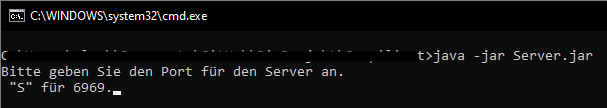
# Installationsanleitung und Setup

1. Entpacken der compiliert.rar Datei.

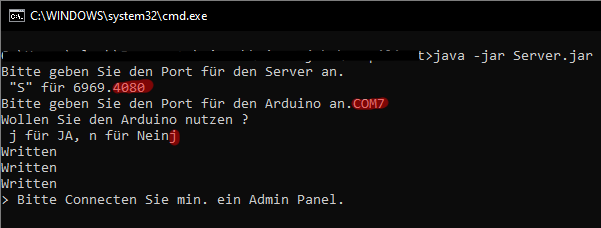


So sollte das Entpackte Archiv aussehen.

1. Führen Sie die Start Server.bat aus. Danach Sollte sich folgendes Konsolenfenster öffnen: (falls nicht, stellen Sie sicher dass Sie java runtime 11 installiert haben)

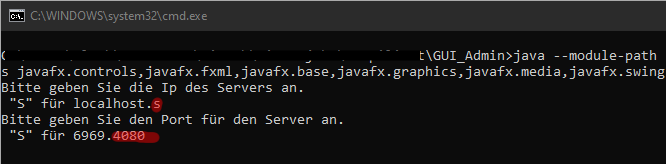


1. Folgen Sie den Anweisungen und geben Sie den gewünschten Port den ihr Server belegen soll ein.
2. Geben Sie nun den Port des Arduinos an. Falls Sie keine Arduino Simulation wünschen,können Sie das im nächsten Schritt eingeben und ihre Portangabe ist somit irrelevant.

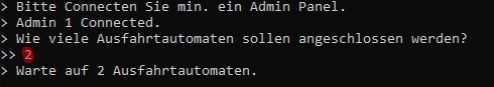


Rot die Benutzereingabe für die Verwendung von einem Arduino.

1. Öffnen Sie nun den Ordner “GUI\_Admin” und führen Sie die Start.bat aus.
2. Geben Sie die Ip des Servers an und den Port. Falls Sie das Programm auf einem System laufen lassen wollen geben Sie “s” für localhost ein oder den Standard Port an.



1. Geben Sie die Anzahl des nächsten Guis an, dass ihnen der Server ausgibt.



1. Ab Schritt 4 kann das Prozedere für jedes vom Server geforderte Gui wiederholt werden.
2. Falls Sie nun alle 4 GUIs und den Server gestartet haben und die 4 Guis ihnen Angezeigt werden haben Sie das System erfolgreich gestartet.

# 

# 

# 

# 

# Bedienungsanleitung und Funktionsweise

## Server

Der Server benötigt im Regelfall keine weitere Benutzer Interaktion.

## Admin GUI

## 

* 1. Die Konsole des Admins sie gibt Auskunft über den aktuellen Status des Programms und lässt dessen Aktivitäten überwachen.
  2. Simple Anzeige der Parkplatzsituation.
  3. Interaktionsfeld für Parkplätze. Es lassen sich Parkplätze hinzufügen, wobei die Id automatisch gesetzt wird. Auch kann man über die Id einen Entfernen oder den Status eines schon vorhandenen über die Id ändern.
  4. Eine Gesamtübersicht über alle Parkplätze mit ihrer Id, Status und ob ein Parkplatz vermietet ist. Die im Bereich 3 benötigte Id entspricht immer der aus der Tabelle.

## 

## Ein - und Ausfahrts GUI

## 

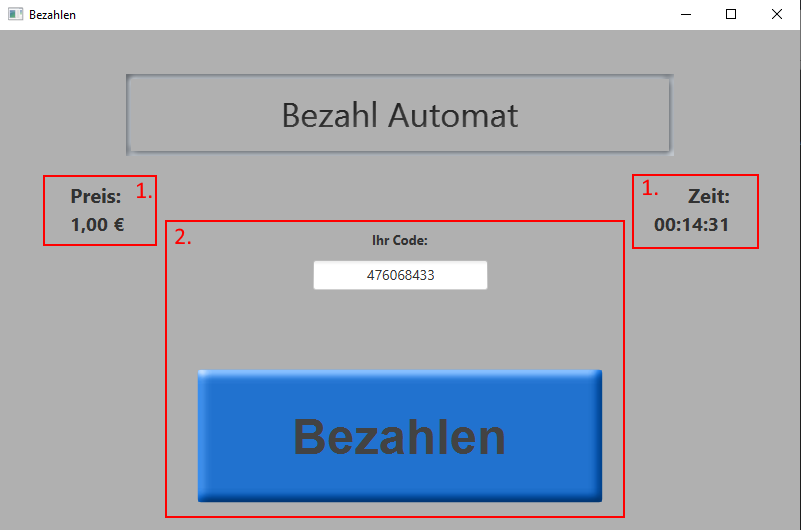
1. Der Button wenn man Einparken will.
2. Wenn man den Button drückt wird ein Code angezeigt der zum bezahlen und zum ausparken benutzt wird.

Der einzige Unterschied zur Ausfahrt ist das man dort den Code eingeben muss.

## 

## 

## Bezahl GUI



1. Anzeigen für Preis und die Parkdauer.
2. Erfüllt dieselbe Funktion wie bei der Ausfahrt.

# 

# 

# Technische Details

## Übersicht

Das ganze Projekt basiert auf einer Server - Client Architektur, das heißt das System kann ohne Probleme auf mehreren Systemen Laufen die über ein Netzwerk verbunden sind.

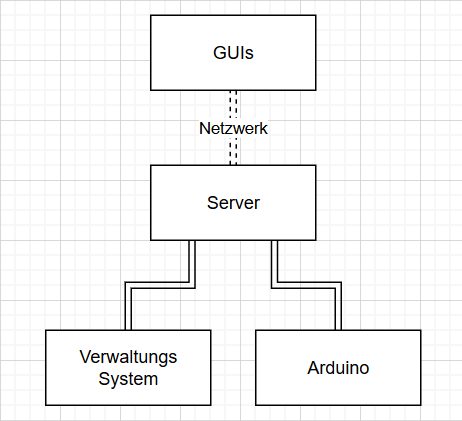
Der Server verwaltet die verschiedenen Oberflächen und verbindet diese mit dem eigentlichen Verwaltungssystem für die Parkplätze. Außerdem Steuert er den Arduino.

Die verbindung zum Arduino wird über eine Serielle Schnittstelle verwirklicht über die Codewörter hin und hergeschickt werden und so den Arduino mit dem Java Code verbindet. Diese Verbindung wurde soweit abstrahiert das man über einen einfachen Methodenaufruf eine bestimmte Aktion beim Arduino auslösen kann, Zb. öffne Schranke. Diese funktionen wurden als Bibliothek extrahiert und dem kern Projekt hinzugefügt.

## Server

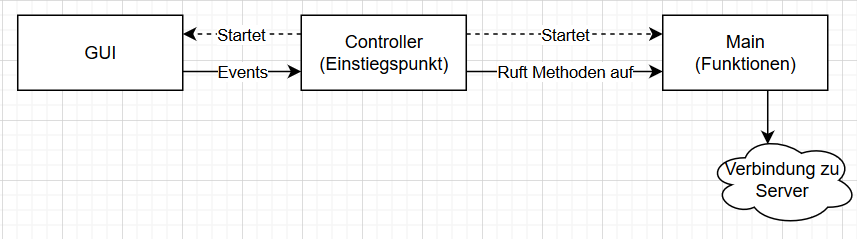
Der Server übernimmt die zentrale Verwaltungs und Verbindungs Rolle indem er die verbindungen zu den Oberflächen verwaltet und für jede einen eigenen Thread öffnet, sodass eine Parallele abarbeitung möglich wird.

Er verbindet eben diese Verbindungen mit den funktionen des eigentlichen Verwaltungssystem.

Zusammenfassend kann man sagen: Der Server steht im Mittelpunkt des Systems und verbindet die verschiedenen Teilkomponenten. 

## GUIs

Die GUIs sind im Kern alle gleich aufgebaut weshalb ich hier nur von GUIs reden werde. Sie basieren auf javafx und wurden nach dem Konzept gebaut die Funktionen aus dem eigentlichen GUI rauszuhalten. Das heißt es gibt die Controller Klasse die das Gui startet und die Events von diesem entgegennimmt und dann die zugehörige Funktion in einer dritten Klasse auslöst, die Main Klasse. Die main Klasse beinhaltet die Funktionen die das Gui haben soll.



# Klassendiagramm