

Konzeptpapier: Programmierung einer „Eisenbahn“



Inhalt

Dokumenthistorie	3
1. Gesamtüberblick	4
2. Anforderungsanalyse	4
3. Umsetzung und Architektur	4
3.1 Festlegung der Umsetzung	4
3.2 Systemanforderungen	6
4. Daten	6
5. Steuerung	7
6. Protokollierung	8
7. Qualitätssicherung	8

Dokumenthistorie

Nr.	SW Version	Dokument Version	Datum	Verfasser
1	Release_v0.1	Konzeptpapier_v0.1	07.10.2020	Niklas Dausch
2	Release_v0.2	Konzeptpapier_v0.2	08.10.2020	Niklas Dausch
3	Release_v0.3	Konzeptpapier_v0.3	09.10.2020	Niklas Dausch
4	Release_v0.3.1	Konzeptpapier_v0.3.1	10.10.2020	Niklas Dausch
5	Release_v0.3.2	Konzeptpapier_v0.3.2	10.10.2020	Niklas Dausch
6	Release_v1.0	Konzeptpapier_v1.0	11.10.2020	Niklas Dausch

1. Gesamtüberblick

Ziel dieses Projektes ist die Programmierung einer Eisenbahn. Deren Konfiguration, bzw. Zusammenstellung soll manuell erfolgen. Die erstellten Konfigurationen werden gespeichert und können zu einem späteren Zeitpunkt wiederverwendet werden.

Nach erfolgreicher Konfiguration des Zuges wird eine Zugfahrt modelliert. An den Haltepunkten kann die Zahl der Passagiere angepasst werden, sowie Informationen zur Belegung und Konfiguration abgefragt werden. Nach Erreichen des letzten Haltepunktes endet das Programm.

2. Anforderungsanalyse

Die Kundenanforderungen gestalten sich wie folgt:

- a) Manueller Zusammenbau der Eisenbahn
- b) Ein Zug besteht aus einer Lok und x Wagen
- c) Die Wagen weisen folgende Spezifikationen auf:
 - a. Wagen-Typ 1: Länge: 2 Meter, Fenster: 4, Achsen: 2
 - b. Wagen-Typ 2: Länge: 3 Meter, Fenster: 6, Achsen: 3
 - c. Wagen-Typ 3: Länge: 4 Meter, Fenster: 8, Achsen: 4
- d) Speichern und Wiederabrufen der Konfigurationen
- e) Die Eisenbahn soll umherfahren
- f) Passagiere können ein- und aussteigen
- g) Die Konfiguration und der Status müssen abrufbar sein.
- h) Keine graphisch aufwendige Lösung ist erforderlich.
- i) Bearbeitungszeit: 5 Arbeitstage.

Die Lieferbedingungen sind wie folgt:

- a) Konzeptpapier
- b) C++ Code
- c) Executable

3. Umsetzung und Architektur

Im folgenden Abschnitt erfolgt die Erläuterung der Umsetzung. Damit einher gehen die Klärung der Kundenanforderungen und deren konzeptionelle Implementierung.

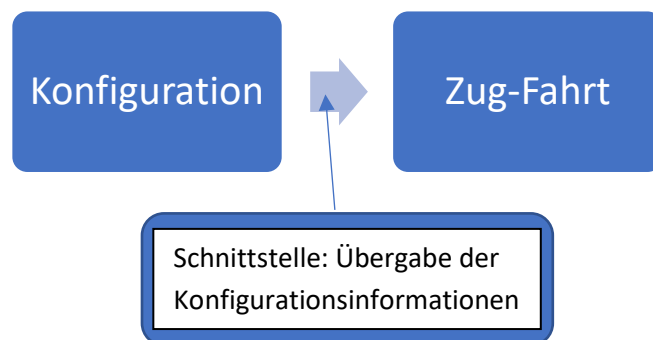
3.1 Festlegung der Umsetzung

Zur Realisierung des Projekts „Eisenbahn“ wird eine Agile Softwareentwicklung gewählt.

Auf Grund der teilweise nicht eindeutig spezifizierten Kundenanforderungen ist die agile Herangehensweise sowohl für den Kunden als auch für den Auftragnehmer von Vorteil.

Die Kundenanforderungen weisen einen hohen Grad an Uneindeutigkeiten. Zur Umsetzung ist deshalb eine Konkretisierung der unter Punkt 2 genannten Anforderungen notwendig:

- 1) Da es keine Spezifikationen zur graphischen Ausarbeitung gibt, wird eine Konsolenanwendung entwickelt. Für das umzusetzende Projekt ist dies im Bezug auf mehrere Punkte vorteilhaft:
 - a. Effektives Vorgehen wird hierdurch ermöglicht, da es die Problemstellung löst
 - b. Der Aspekt der Wirtschaftlichkeit ist erfüllt, da die Umsetzung gebrauchstauglich ist, kostengünstig und mehrfach verwendbar. Es sind keine Frameworks, die eine Lizenzgebühr erfordern zur Umsetzung und Anwendung nötig.
 - c. Durch die Vermeidung solcher Abhängigkeiten sind dahingehende Fehler weitestgehend ausgeschlossen.
 - d. Das Programm ist leicht verständlich, zudem robust, zuverlässig und leicht wartbar.
 - e. Voraussichtlich nur geringe Systemanforderungen notwendig.
- 2) Modularer Aufbau des Programmes durch die Aufteilung in einen Konfigurator und die Zugfahrt an sich.



- 3) Keine Information über die Lokomotive, diese wird nur als Bestandteil eines Zuges mit aufgenommen.
- 4) Die Angabe eines „umherfahrenden“ Zuges wird als Bewegung des Zuges von einer Haltestelle zur nächsten dargestellt. Die Bewegung wird in verbrauchten Zeiteinheiten dargestellt.)
- 5) Die Anzahl der gespeicherten Eisenbahnkonfigurationen wird nicht softwareseitig begrenzt.
- 6) Die Konfiguration der Eisenbahn ist nur am Beginn des Programms möglich.
- 7) Die Konfiguration der Eisenbahn ist nicht in den einzelnen Bahnhöfen möglich.
- 8) Nach der fertigen Konfiguration wird der Zug im ersten Bahnhof bereitgestellt.
- 9) Die Anzahl der Bahnhöfe wird vor Abfahrt abgefragt.
- 10) Die Belegung bezieht sich auf alle Wagons. Die Passagiere können sich im Zug frei von Wagon zu Wagon bewegen.
- 11) Die Abfrage der Konfiguration gibt die Wagenreihung und die Anzahl der Passagiere zurück.

- 12) Der Zug fährt von Bahnhof zu Bahnhof und nur am Bahnhof kann ein- und ausgestiegen werden.
- 13) Es gibt kein Limit für die Anzahl der ein- und aussteigenden Passagiere.
- 14) Der Status und die Konfiguration sind nur im Bahnhof abfragbar.
- 15) Der Zug verlässt den Bahnhof nur auf Befehlseingabe (Abfahrt).
- 16) Der Zug fährt bis zum letzten Bahnhof der Strecke oder bis der Abbruch-Befehl eingegeben wird.
- 17) Das Verhindern von Fehleingaben durch den Nutzer ist von hoher Bedeutung. Die Sicherheit der Eingaben ist sehr hoch zu bewerten, sodass keine Schäden durch Fehlverhalten des Benutzers entstehen.

3.2 Systemanforderungen

Folgende Anforderungen sind zu beachten:

Software:

- Windows 10 oder höher

Hardware:

- Prozessort Intel i3 oder höher
- Verfügbarer Speicherplatz

4. Daten

Bei den anfallenden Daten handelt es sich zum einen um die Benennung der Konfiguration, und zum anderen um die Konfiguration der Eisenbahn an sich.

Der Datenfluss gestaltet sich wie folgt:

Konfiguration -> Namensgebung -> Speichervorgang

Speicherabfrage der Namen -> Laden der Konfiguration

Datenentitäten:

- Die Namen werden in Form von <string> in einer Text-Datei auf dem verwendeten Rechner gespeichert.
- Die Konfiguration wird in Form von <int> in einer Txt-Datei auf dem verwendeten Rechner gespeichert.

Um die Datentypen in der korrekten Form abzuspeichern, wird bei der Eingabe der Daten auf den korrekten Typ geachtet. Der Nutzer wird so lange zur Korrektur der Eingabe aufgefordert, bis eine valide Eingabe erfolgt ist.

Hinweise des Herstellers zur möglichen Weiterentwicklung:

- a) Löschen vorhandener Konfigurationen
- b) Überschreiben vorhandener Konfigurationen
- c) Personalisierung der Konfigurationen

5. Steuerung

- I) Programm Start
 - a. Die gelieferte EXE muss auf dem Windows10 Rechner gespeichert werden
 - b. Die Dateien sind unter: https://github.com/Nick-257/Zug_Aufgabe
 - c. Zum Start muss folgende Datei ausgeführt werden:
Eisenbahn_ConsolApp.exe
- II) Eingabeparameter Zugkonfiguration
 - a. Name des Lokführers (string)
 - b. Anzahl Waggons (int)
 - c. Waggon-Typ (int)
 - d. Speichern der Konfiguration → Ja oder Nein Option
 - e. Laden einer Konfiguration → Ja oder Nein Option
 - f. Name der Konfiguration (string) aus den verfügbaren Konfigurationen
- III) Eingabeparameter Zugfahrt
 - a. Anzahl der Bahnhöfe (unsigned): Negative Werte werden als Short-Cut zur höchsten Zahl (4 294 967 296) angesehen, ohne die ganze Zahl eintippen zu müssen.
 - b. Es ist immer die im Programm angezeigte Syntax zu verwenden
 - c. Abfahrt oder Abf (string) → Zug fährt zum nächsten Bahnhof
 - d. Abbruch oder Abb (string) → Das Programm wird abgebrochen, neuer Start möglich
 - e. Status oder S (string) → Anzeige der Wagons/Typ und Anzahl der Passagiere
 - f. Belegung oder B (string) → Anzeige der Passagiere im Zug
 - g. Personen oder P (unsigned) → führt zum Untermenü ein-, aussteigende Passagiere
 - h. Einsteigende und aussteigende Passagiere werden nur als ganze Zahl erfasst. Negative Werte werden als Short-Cut zur höchsten Zahl (4 294 967 296) angesehen, ohne die ganze Zahl eintippen zu müssen.
- IV) Programmablauf
 - a. Beim ersten Projektstart wird der Name des Zugführers eingegeben.
 - b. Der Zugführer wird begrüßt und muss seinen Zug zusammenstellen.

- c. Beim ersten Projektstart wird der Zug in seiner Konfiguration festgelegt. Die Festlegung erfolgt über die Abfrage der Wagon-Anzahl und anschließend der Zuordnung, welcher Wagon-Typ pro Wagon verwendet werden soll.
- d. Für jede neue Zugkonfiguration muss ein Name dafür vergeben werden.
- e. Der Zugführer kann eine bereits bestehende Zugkonfiguration aufrufen, wenn im vorherigen Programmaufruf eine Konfiguration gespeichert wurde, oder eine neue Zugkonfiguration eingeben.
- f. Nach Fertigstellung / Auswahl des Zuges wird vor Start zur Kontrolle die aktuelle Zugkonfiguration angezeigt.
- g. Nach der Zugkonfiguration wird die Anzahl der Bahnhöfe festgelegt.
- h. Nach Beendigung der Zugkonfiguration fährt der Zug vom Rangierbahnhof in den Start-Bahnhof,
- i. Der Zug bleibt im Start-Bahnhof stehen bis das Abfahrtssignal (Befehl Abfahrt) gegeben wird.
- j. Der Zug fährt dann ohne Unterbrechung bis zum nächsten Bahnhof.
- k. Der Zug hält in jedem Bahnhof an.
- l. In den Bahnhöfen kann jeweils ein- und ausgestiegen werden. Die Anzahl der ein- und aussteigenden Personen wird über ein Eingabefeld abgefragt. Im Start-Bahnhof kann nur eingestiegen werden, ein Aussteigen im Start-Bahnhof wird als nicht realistisch angesehen.
- m. In jedem Bahnhof kann immer der Status abgefragt werden, wobei der Status die Wagonkonfiguration und die gesamte Belegung enthält. Ist nur die Belegungsinformation gewünscht, dann kann über den Befehl: Belegung explizit dieser Status abgerufen werden.
- n. Der Zug fährt von Bahnhof zu Bahnhof bis zum Endbahnhof oder bis der Abbruch Befehl gegeben wird.
- o. Das Programm verabschiedet sich und kann danach neu gestartet werden.

6. Protokollierung

Verwendung von Github zur Versionskontrolle des Codes. Zusätzliches Führen eines Changelogs, um Änderungen nachverfolgen zu können.

7. Qualitätssicherung

Die Prüfung des Programmes erfolgt durch Debugging und manuellen Tests anhand von Testpersonen.

Bei allen Punkten der Entwicklung und Weiterentwicklung, muss wie unter Kapitel 3.1 Unterpunkt 17, der Sicherheitsaspekt berücksichtigt werden.

Des Weiteren kann ein Testprogramm entwickelt werden, das alle Eingabefunktionalitäten überprüfen kann.