Projekt: Eventmanagment

Akteure: Andy Schleising, Valerii Drobiazgho

Eventmanagment ist eine Java Applikation, die es dem Benutzer Ermöglicht mehrere Events zu verwalten.

//Einleitung

Der Benutzer kann das Programm direkt ansteuern, und somit ohne Benutzerlogin die Events anlegen und verwalten.

So kann das Event an verschiedenen Orten und Bundesländern angelegt werden, mit verschiedenen Gegebenheiten, zb. : Hallengröße , Aussteller und die einhergehenden Autos(Ware), Mitarbeiter vor Ort separiert, in Intern und Externen Mitarbeiter.

//Unsere Design Entscheidung

Die Entscheidung wurde aufgrund von Personalen Entwicklungen getroffen, so dass der Aufwand nicht weiter steigt trotz schwindender Mitarbeiter.

Des Weiteren haben wir uns entschlossen den Leitsatz von „EINFACHHEIT“ zu folgend und wenig komplizierter Dinge einzubauen, die schwerlich zu implementieren sind und zu erläutern sind.

//Funktionen des Programms

Es kann nachträglich Veränderungen stattfinden, die beinhalten folgendes:

- Anzahl an Austellern
- Verfügbare Autos zur Besichtigung
- Stellplätze für Austeller können erweitert werden
- Autos können Klassifiziert werden (Modell, Zulassung, Halter, Kilometer, PS, maxKmH)

//Beschreibungen der Klassen von der Applikation

Es existieren „13“ Klassen,

Employee, EventCreate, EventName, EventOutput, GlobalClass, Main, Place, Price, Read, Warehouse, Write, Car, Event.

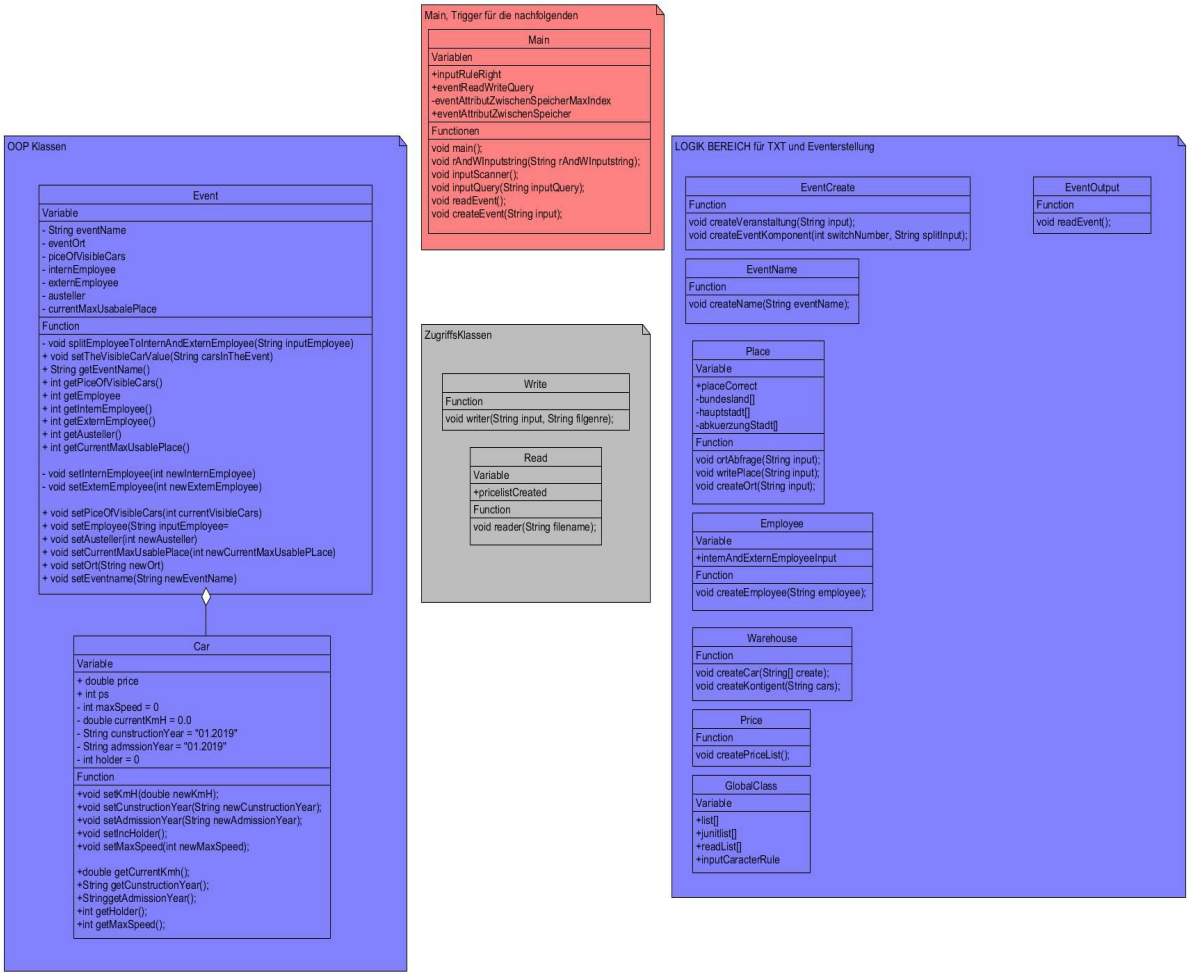
Von diesen werden „Main, EventCreate, Employee, EventName, GlobalClass, Place, Price, Warehouse, Car, Event“ als Speicherung für die Daten genutzt

„Car, Event“ Klassen für die Verwendung von der „OOP“ Funktionen.

„Write und Read“ dient den zugriff auf die Festplatte zum lesen und schreiben einer Datei.

„EventOutput“ dient dazu mittels „Read“ die Gespeicherten Elemente auszugeben.

Hierzu ein einfaches Klassendiagramm.



Der Rot Gehighlighte Bereich ist der Kernel, die Main Klasse Triggert die in den Blau gekennzeichneten Bereiche. Links ist die OOP Ebene und Rechts die Logik. Der Graue Bereich zwischen den 3 Bereichen dient dem Lese und Schreibzugriff.

Links Blauer Bereich:

Folgende Klassen : Event, Car

Rechts Blauer Bereich:

Folgende Klassen: CreateEvent, Place, Employee, Warehouse, EventName, Price, GlobalClass und EventOutput

Der Graue Bereich:

Folgende Klassen: Write, Read

Und der Rote Bereich ist nur mit der Main Klasse vertreten.

//Warum die Speicherung in TXT Dateien ?

Die dienen dazu das eingegebene als Rücklagen Konsistent auf der Festplatte zu speichern und bei einem Neustart wieder im Programmcode ausführen zu können.

//Welche Pakete gibt es

Paket „oop“

Paket: „de.myrtana.fh“

//Warum benutzen wir die oben stehenden Pakete

oop :

Verwendung der Objectorieintierten Lösungen für Event.
Event und Car, werden über die Klassen abgebildet.

de.myrtana.fh :

Dies ist der Kernel von der Java Applikation.
Dient der Erstellung von Txt Dateien zur Speicherung der Parameter.
Hiermit werden die Daten verarbeitet, und die Klassen aufgerufen.

// Folgende Bibliotheken sind notwendig

JUnit: für Unit Test

Java.io.ioexception: für das Schreiben von Daten und das Abfangen von Fehlern

Java.util.Scanner: zum einlesen einer Eingabe von der Konsole

//Was ist für die Ausführungen/Installation zu beachten?

Es wird vorausgesetzt, dass Java Runtime Environment installiert ist (ab ver. JRE 1.7).

Der Benutzer muss Adminrechte haben oder hat den schreibzugriff auf die Festplatte, wegen der Erstellung der TXT Dateien.

Es muss die Eingaberegel beachtet werden:

Name < Ort < InternMitarbeiter , ExternMitarbeiter < Automodell-Stück , X-x^x <