10 MATSE-SE Projekt: Fuhrpark-Verwaltung

|  |  |
| --- | --- |
|  | **Ziel**: Sie entwickeln in Projektgruppen eine Fuhrpark-Verwaltungs-Software. Diese wird die Verwaltung von Mitarbeitern, Fahrzeugen sowie die entsprechenden Fahrzeugbuchungen umfassen. |

**„In einem erfolgreichen Projekt müssen nicht nur Alle am gleichen Strang ziehen, sondern auch in die gleiche Richtung.“**

# Zielstellung des Projekts

In diesem Projekt werden Sie über drei Wochen zentrale Inhalte aus Lernfeld 3 und 4 selbstständig erarbeiten und anwenden. Ein „einsatzfähiges Softwareprodukt“ als Projektergebnis ist nicht wichtig! Zentral ist, dass Sie sich intensiv mit den Inhalten beschäftigen und diese selbst erarbeiten. „Copy und Paste“ führt zu schnellen Ergebnissen, nicht aber zu einem tieferen Verständnis, das wir hier erreichen möchten.

# Projektauftrag

Entwickeln Sie ein Programm, das die Verwaltung eines Fuhrparks ermöglicht.

* Ein Unternehmen hat einen Fuhrpark der den Mitarbeiter\*innen zur Verfügung steht.
* Mitarbeiter\*innen können für einen bestimmten Zeitraum und einen zu nennenden Zweck ein Fahrzeug ausleihen.
* Für unterschiedliche Zwecke (Stadtfahrt, Langstreckenfahrt, Transport von Arbeitsmaterial) werden unterschiedliche Fahrzeugtypen (z.B. PKW, LKW, Motorrad) bereitgestellt.
* Die Mitarbeiter-Daten sowie die Daten des Fuhrparks und Fahrzeugbuchungen werden in einer XML-Datei gespeichert.

# Vorbereitende Aufgaben

Bearbeiten Sie die Dokumentation (Projektziele, Projektrollen, Milestones) **nur einmal je Projektgruppe**!

|  |  |  |
| --- | --- | --- |
| **Nr.** | **Projekt-Vorbereitungen** | **Erledigt** |
| 1 | Erhalt der Projektunterlagen |  |
| 2 | Nehmen Sie sich ausreichend viel Zeit die folgenden Arbeitsschritte zu lesen. Zu den neuen Fachbegriffen und Fachkonzepten haben Sie zahlreiche Hilfsangebote im [Moodle Kurs (Abschnitt Projektarbeit)](https://moodle.hems3.de/course/view.php?id=365#section-5). |  |
| 3 | Verschaffen Sie sich einen Überflick über die Ausbaustufen mit den jeweiligen Anforderungen. |  |
| 4 | Lesen Sie die Beschreibung der Ausbaustufen. Setzen Sie sich gemeinsam ein vorläufiges Ziel das für Sie realistisch erscheint. |  |
| 5 | Teilen Sie die unterschiedlichen [Arbeitsrollen ein (siehe Abschnitt Projektrollen)](#_Projektrollen_2). |  |
| 6 | Füllen Sie für eine Zeitplanung das Raster „[Milestones](#_Milestones_1)“ aus. |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **Technische Vorbereitungen** | **Erledigt** | **Geprüft** |
| 1 | Wählen Sie **eine** Entwicklungsumgebung (IDE) z.B. Eclipse, Net Beans, IntelliJ. Achten Sie darauf, dass die IDE **Java FX** unterstützt!  **Tipp**: Diese Wahl sollte gemeinschaftlich in der Gruppe getroffen werden. |  |  |
| 2 | Einigen Sie sich auf die Form des Dateiaustauschs (Cloud, Git-Hub/Repository, Server, E-Mail) |  |  |

Bearbeiten Sie die folgenden Aufgaben, dokumentieren Sie ihre Entscheidungen im Kapitel [Zieldefinition](#_Zieldefinition).

# Ausbaustufen

## Funktionale Anforderungen

**Die folgende Beschreibung der Anforderungen ist nicht in allen Punkten präzise und umfasst nicht alle Details, dies ist gewollt! Treffen Sie auf Basis ihres Wissens die richtigen Entscheidungen.**

### Ausbaustufe: Adressbuch erstellen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Anforderung** | **Bereich[[1]](#footnote-1)** | **Erledigt** | **Geprüft** |
|  | Erstellen Sie ein Adressbuch mit einer grafischen Nutzeroberfläche unter Verwendung von **Java FX 8**. Nutzen Sie hierzu das [Tutorial von code.makery](http://code.makery.ch/library/javafx-8-tutorial/). | LF |  |  |
|  | Speichern Sie die Kontaktdaten in einer **XML Datei**. | LF 3 |  |  |
|  | Halten Sie stets, auch in folgenden Ausbaustufen das **Model-View-Control Modell** ein. | LF 4 |  |  |
|  | Ausbaustufe auf Funktionsfähigkeit getestet? |  |  |  |

**Hinweis**: Alle Schritte dieser Ausbaustufe sind im [Tutorial](http://code.makery.ch/library/javafx-8-tutorial/) verfügbar!

### Ausbaustufe: Verwaltung von Fahrzeugen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Anforderung** | **Bereich** | **Erledigt** | **Geprüft** |
|  | Erweitern Sie das Programm so, dass neben Kontakten auch **Fahrzeuge** in das Programm ein- und ausgegeben werden können. Dazu soll die Java FX GUI verwendet werden. | LF 3 |  |  |
|  | Erstellen Sie eine **XML-Datei**, in der die Fahrzeuge **persistent** gespeichert werden können. | LF 3 |  |  |
|  | Ausbaustufe auf Funktionsfähigkeit getestet? |  |  |  |

### Ausbaustufe: Implementierung eines Buchungssystems

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Anforderung** | **Bereich** | **Erledigt** | **Geprüft** |
|  | Erstellen Sie eine GUI, über die Mitarbeiter (Kontakte) jeweils Fahrzeuge buchen können. | LF 3 |  |  |
|  | Speichern Sie diese Buchungen in einer XML-Datei. | LF |  |  |
|  | Prüfen Sie, dass zu einem Zeitpunkt das Fahrzeug nicht bereits durch einen anderen Mitarbeiter gebucht ist. | LF 4 |  |  |
|  | Ausbaustufe auf Funktionsfähigkeit getestet? |  |  |  |

### Ausbaustufe: Algorithmische Optimierung

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Anforderung** | **Bereich** | **Erledigt** | **Geprüft** |
|  | Implementieren Sie eine **Suchfunktion** (lineare Suche), die es ermöglicht nach einem bestimmten Fahrzeug zu suchen. | LF 4 |  |  |
|  | Ermöglichen Sie es die Kontakte und Fahrzeuge sortiert ausgeben zu lassen. Nehmen Sie die **Sortierung** mit einem selbst geschriebenen Sortieralgorithmus (Insertion-Sort, oder Selection-Sort). | LF 4 |  |  |

### Ausbaustufe: Optimierung der Datenmodellierung

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Anforderung** | **Bereich** | **Erledigt** | **Geprüft** |
|  | r | LF 3 |  |  |
|  | Passen Sie die GUI so an, dass gezielt ein bestimmter Fahrzeugtyp (PKW, LKW etc.) gebucht werden kann. | LF 3 |  |  |
|  | Ausbaustufe auf Funktionsfähigkeit getestet? |  |  |  |

### Ausbaustufe – Ergänzungen und eigene Anforderungen

Ab dieser Ausbaustufe können Sie eigene, abweichende Features für ihre Software definieren. Diese werden alternativ zu den folgenden Ausbaustufen bewertet.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Anforderung** | **Bereich** | **Erledigt** | **Geprüft** |
|  |  |  |  |  |
|  |  |  |  |  |
|  | Ausbaustufe auf Funktionsfähigkeit getestet? |  |  |  |

### Ausbaustufe: Algorithmische Vertiefung

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Anforderung** | **Bereich** | **Erledigt** | **Geprüft** |
|  | Berechnung und Ausgabe der am **häufigsten ausgeliehenen** Fahrzeugtypen und Fahrzeuge. | LF 4 |  |  |
|  | Berechnung und Ausgabe der **durchschnittlichen Ausleihzeit**, insgesamt und nach Fahrzeug-Typen aufgelistet. | LF 4 |  |  |
|  | Berechnung und Ausgabe der Mitarbeiter mit den **meisten Leihtagen**. | LF 4 |  |  |
|  | Ausbaustufe auf Funktionsfähigkeit getestet? |  |  |  |

### Ausbaustufe: Vertiefung Datenstrukturen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Anforderung** | **Bereich** | **Erledigt** | **Geprüft** |
|  | Verwenden Sie zur Repräsentation/Speicherung der Kontaktdaten einen **binären Suchbaum**. | LF 4 |  |  |
|  | Optimieren Sie den binären Suchbaum zu einem **balancierten** binären Suchbaum (**AVL-Baum**) | LF 4 |  |  |
|  | Ausbaustufe auf Funktionsfähigkeit getestet? |  |  |  |

### Ausbaustufe: Datenbanken

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Anforderung** | **Bereich** | **Erledigt** | **Geprüft** |
|  | Speichern Sie die Daten in einer **MySQL** Datenbank (z.B. XAMPP)  **Hinweis**: Hierfür sind gute Vorerfahrungen mit MySQL Datenbanken notwendig! | LF |  |  |
|  | Ausbaustufe auf Funktionsfähigkeit getestet? |  |  |  |

## Nicht funktionale Anforderungen

### Ausbaustufe Dokumentation (Basisanforderungen)

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **Anforderung** | **Erledigt** | **Geprüft** |
|  | Schriftliche Dokumentation der **Arbeitsrollen**. |  |  |
|  | Dokumentation der **Milestones.** |  |  |
|  | Dokumentation der **Änderungen** von **Milestones** (bei Bedarf). |  |  |
|  | Einigen Sie sich darauf welche Programme Sie zur **Programmierung** und zur **Dokumentation** verwenden wollen (astah community, yEd, Visio). Alle Gruppenmitglieder sollten das Programm auf dem PC haben! |  |  |
|  | Darstellung von mindestens vier Klassen der Model-Ebene (Datenrepräsentation: Fahrzeuge, Kontakte, Buchung) als UML Klassendiagramm. |  |  |
|  | Nennung von Problemen im Projekt und wie diese gelöst wurden. |  |  |

### Anforderungsstufe – Dokumentation (Vertiefung)

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **Anforderung** | **Erledigt** | **Geprüft** |
|  | Stellen Sie den Ablauf einer Fahrzeug-Buchung als UML-**Klassendiagramm** dar. |  |  |
|  | Stellen Sie den Ablauf einer Fahrzeug-Buchung als UML-**Sequenzdiagramm** dar. |  |  |

### Präsentation der Projektergebnisse

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **Anforderung** | **Erledigt** | **Geprüft** |
|  | Es wird deutlich dargelegt welche **Ausbaustufe** und welche einzelnen Features realisiert wurden. |  |  |
|  | **Probleme** in der Entwicklung (fachliche als auch gruppeninterne) werden genannt und die angewendete Lösungsstrategie kurz erläutert. |  |  |
|  | **Aufbau** und **Strukturierung** (Sachliche Gliederung, Inhalte werden logisch und nachvollziehbar dargelegt, zielorientierte Darlegung der Ergebnisse) |  |  |
|  | Sprachliche Gestaltung (Ausdrucksweise, Satzbau, Stil) |  |  |
|  | Zielgruppengerechte Darstellung (Medieneinsatz, Visualisierung, Körpersprache. |  |  |

Die Anforderungen an die Projektpräsentation basiert auf den Kriterien zur [Bewertung von Abschlussprojekten im IT-Bereich der IHK Darmstadt](https://www.darmstadt.ihk.de/produktmarken/aus_und_weiterbildung_channel/pruefungen/downloads/Muther/IT_Hinweise_Projektarbeit/2551050).

# Tipps zur Vorgehensweise in Projekten

1. Verschaffen Sie sich einen Überblick über die **funktionalen** und **nicht** **funktionalen** Anforderungsbereiche.
2. Achten Sie auf eine sinnvolle **Modellierung** der Klassen sowie der verwendeten Datenstrukturen. Dies gilt insbesondere ab Ausbaustufe II.
3. Achten Sie darauf die **Dokumentation** kontinuierlich zu pflegen, nicht erst am Schluss!
4. Gehen Sie **arbeitsteilig** vor. Das Projekt beinhaltet viele Teilaufgaben, die sinnvollerweise unabhängig voneinander gelöst werden. Es ist nicht notwendig, dass alle Gruppenmitglieder an den gleichen Teilaufgaben arbeiten!
5. Setzen Sie sich **realistische Ziele** (z.B. Realisierung der Ausbaustufe III), hierbei kann die [SMART-Formel](https://de.wikipedia.org/wiki/SMART_(Projektmanagement)) helfen.
6. Unterteilen Sie das Ziel in realistische **Milestones** (Teilziele).
7. Haben Sie keine Angst das **Gesamtziel** oder **Milestones** zu **ändern**!
8. Fokussieren Sie sich zunächst auf eine **funktionsfähige Basisversion**. Die weiteren Ausbaustufen sind teilweise erst dann (sinnvoll) umsetzbar, wenn das Gesamtprogramm zumindest rudimentär funktioniert.
9. Nutzen Sie den umfangreichen **Zeitrahmen** bereits **am Anfang** aus!
10. Nutzen Sie **Hilfsangebote**! Fragen sie die Lehrkraft oder tauschen sie sich mit anderen Projektgruppen aus! Die Projektgruppen stehen nicht im Wettbewerb zueinander.

# Stammblatt-Projektarbeit

## Gruppenmitglieder

Tragen Sie die Gruppenmitglieder ein (maximal 4 Personen pro Gruppe!). Ergänzen Sie später die Zuweisung der Projektrollen, siehe folgenden [Abschnitt Projektrolle](#_Projektrollen_3)n.

|  |  |  |
| --- | --- | --- |
|  | Gruppenmitglied | Projektrolle(n) |
| 1 | Erik Bong | Guru, Organisator |
| 2 | Felix Schwarz | Protokollant, Feelgood-Manager |
| 3 | Micky Koch | Spec-Checker |
| 4 | - | - |

## Projektrollen

In einem Projekt fallen viele unterschiedliche Aufgaben an. Für bestimmte Aufgaben ist es sinnvoll eine Person zu definieren, die immer als Ansprechpartner gelten kann! Grundsätzlich ist wechselseitiges informieren über den aktuellen Projektstand sinnvoll. Dabei sollten **positive Entwicklungen** und **mögliche Konflikte und Risiken** mitgeteilt werden.

**Organisator**: Der Organisator achtet auf die Einhaltung der gesetzten Ziele und passt den Zeitplan und ggf. die angestrebten Ausbaustufen, in **Absprache mit den anderen Gruppenmitgliedern** an.

**Guru**: Der Guru sollte der erfahrenste Programmierer sein, an den sich alle Gruppenmitglieder wenden können, sollten Probleme auftauchen.

**Protokollant**: Der Protokollant notiert am Ende eines jeden Arbeitstages den Arbeitsfortschritt mit **wenigen** Sätzen. Es wird notiert, welche Aufgaben noch offen sind und welche erledigt wurden. Ergänzend sollten (mögliche) Probleme benannt notiert werden. Hierzu befragt der Protokollant alle Gruppenmitglieder und teilt die Ergebnisse mit. Hierzu sollte die Vorlage [Milestones](#_Milestones_2) genutzt werden.

**Spec-Checker**: Ein wichtiger Teil der Software-Entwicklung ist die Modellierung. Diese ermöglicht es gemeinsame Schnittstellen zu verwenden. Der Spec-Checker achtet darauf, dass diese gruppeninternen Schnittstellen fachgerecht modelliert und von Allen eingehalten werden.

**Feelgood-Manager**: Der Feelgood-Manager achtet darauf, dass die Stimmung gut und produktiv bleibt und versucht insbesondere negative/pessimistische Stimmungen zu erkennen und zu beheben.

**Weitere Rollen**: Definieren Sie ggf. weitere Rollen, die Sie als wichtig empfinden.

## Zieldefinition

Notieren Sie unter „Zieldefinition 1“ ihr Gesamtziel in Forme in Form einer Ausbaustufe, die Ihnen als Gruppe als realistisch erscheint. Passen Sie im Projektverlauf ggf. das angestrebte Ziel an, notieren Sie das neue Gesamtziel unter „Zieldefinition 2“ etc.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Angestrebte Ausbaustufen | | Zielanpassung | | |
|  | **funktionale Anforderungen** | **nicht funktionale Anforderungen** | **Anpassung der Zieldefinition** | **Datum und Grund der Änderung** | **Kommentar** |
| Zieldefinition 1: | 8 (9 optional) |  |  |  |  |
| Zieldefinition 2: |  |  |  |  |  |
| Zieldefinition 3: |  |  |  |  |  |

## Milestones

Milestones (dt. Meilensteine) sind Zwischenziele, die eine Zeitangabe verfügen. Milestones unterstützen eine realistische Zeitplanung. Große Ziele wirken schnell unrealistisch und bieten keinen Anhaltspunkt an dem die Arbeit begonnen werden kann. Modularisieren (unterteilen) Sie daher das Projektziel in Zwischenschritte und überlegen Sie sich eine sinnvolle Bearbeitungsreihenfolge. Passen Sie bei Bedarf die Meilensteine (Ziel und/oder Zeitpunkt) an. So können diese z.B. früher oder später erreicht werden oder unter Umständen vollständig entfallen.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Zieldatum | Milestone | Ziel erreicht am | Verantwortliche\*r | Problem/Verzögerung, weil... | Lösungsidee |
|  | 15.03.2017 | * Konstituierung der Gruppen * Einlesen in die Projektanforderungen * Milestones ausfüllen * Zieldefinition dokumentieren * Arbeitsbeginn Adressbuch |  | Alle |  |  |
|  | 15.03.2017 | Ausbaustufe I |  | Alle |  |  |
|  | 15.03.2017 | GUI-Grundgerüst | 15.03.2017 | Micky |  | Nach Tutorial, und grobe GUI-Modellierung im Team |
|  | 15.03.2017 | Klassen(Attribute und Gette rund Setter) | 16.03.2017 | Felix | Länger als geplant |  |
|  | 16.03.2017 | CompareTo() | 16.03.2017 | Felix |  | Doku von compareTo()  nachgeschaut |
|  | 15.03.2017 | Github Integration | 16.03.2017 | Erik | Ging nicht sofort | try and error im team |
|  | 16.03.2017 | GUI und modell verbinden |  | Micky |  |  |
|  | 16.03.2017 | Objekte in XML speichern | 16.03.2017 | Erik |  | Nach tutorial |
|  | 22.03.2017 |  |  |  |  |  |
|  | 22.03.2017 | Zwischenevaluation | 22.03.2017 | Alle |  |  |
|  | 23.03.2017 |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  | 29.03.2017 |  |  |  |  |  |
|  | 30.3.2017 |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  | 19.05.2017 | Finale Abgabe des Projekts |  |  |  |  |
|  | 22.5.2015 | Projektpräsentation |  | Alle |  |  |
|  | 23.05.2017 | Reflexion  Abschließende Evaluation |  | Alle |  |  |

# Bewertungshinweise

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **LF[[2]](#footnote-2)** | **Kriterium** | **Wertung**  0 bis 10 | **Kommentar** |
| **Arbeitsprozess (20%)** | | | | |
|  |  | Hoher Arbeitseinsatz |  |  |
|  |  | Aktive Projektarbeit, guter Lernzuwachs |  |  |
| **I Modellierung (10%)** | | | | |
|  | LF 3 | Einhaltung von Model-View-Control (MVC) |  |  |
|  | LF 3 | Sinnvolle Aufteilung der Klassen |  |  |
|  | LF 3 | Sinnvolle Wahl der Attribute, Methoden und Datenstrukturen |  |  |
| **II Programmierung (40 %)** | | | | |
|  | LF | Lauffähigkeit des Programms |  |  |
|  | L4 4 | Kommentierung des Programmcodes ist vorhanden |  |  |
|  | LF 3 | GUI mit Java FX   * Nutzergerechter Aufbau der GUI * Funktionalität |  |  |
|  | LF | Umfang der Ausbaustufen/Features |  |  |
|  | LF 4 | Sortieralgorithmen wurden sinnvoll eingesetzt   * Suchalgorithmen wurden sinnvoll eingesetzt * lineare Suche * binäre Suche |  |  |
|  | LF 4 | Datenstrukturen wurden sinnvoll eingesetzt   * Array, Arraylist * binärer Suchbaum / AVL-Baum |  |  |
|  | LF 4 | Persistente Speicherung wurde gelöst |  |  |
| **IV Dokumentation (15%)** | | | | |
|  | LF | Rollenverteilung u. Aufgabenverteilung |  |  |
|  | LF | Milestones   * Definition der Milestones * Zieladaption mit Begründung |  |  |
|  | LF 3 | UML-Klassendiagramme der Model-Ebene |  |  |
|  | LF | Umgang mit Problemen   * Probleme wurden antizipiert * Probleme wurden benannt * Lösungsstrategien wurden dokumentiert |  |  |
| **IV Präsentation (15%)** | | | | |
|  | LF | Aufbau und Strukturierung |  |  |
|  | LF | Sprachliche Gestaltung |  |  |
|  | LF | Zielgruppengerechte Darstellung |  |  |

1. **Legende**: „Bereich“ gibt das zugehörige Lernfeld an. Die Angabe **LF** entspricht LF 3 **und** LF4. [↑](#footnote-ref-1)
2. **Legende**: „Bereich“ gibt das zugehörige Lernfeld an. Die Angabe **LF** entspricht LF 3 **und** LF4. [↑](#footnote-ref-2)