

Meilensteine und Ideen für ein Event-Management-System für Anfänger

Dieses Dokument enthält eine Schritt-für-Schritt-Anleitung zur Entwicklung eines **Event-Management-Systems**. Es bietet Anfängern eine klare Struktur und Entwicklungsmeilensteine, um das Projekt modular und verständlich zu gestalten.

Meilenstein 1: Projektstruktur und Basisklassen

Ziele:

1. Verstehen und Implementieren von **Grundstrukturen**.
2. Erste OOP-Prinzipien wie **Klassen** und **Methoden** anwenden.

Aufgaben:

- Erstelle eine Klasse `Kunde` mit folgenden Attributen:
 - `id` (int): Eindeutige ID des Kunden.
 - `name` (String): Name des Kunden.
 - `email` (String): E-Mail-Adresse.
- Schreibe Getter- und Setter-Methoden für jedes Attribut.
- Implementiere eine `toString()`-Methode, die die Kundendetails als String zurückgibt.
- Schreibe ein einfaches Hauptprogramm (`main()`), das:
 - Einen Kunden erstellt.
 - Die Kundendetails ausgibt.

Meilenstein 2: Einführung von Listen

Ziele:

1. Arbeiten mit **Listen** (`ArrayList`) zum Speichern mehrerer Objekte.
2. Daten aus einer Liste abrufen und anzeigen.

Aufgaben:

- Erweitere das Hauptprogramm, sodass mehrere Kunden in einer `ArrayList<Kunde>` gespeichert werden.
- Schreibe eine Methode `addKunde(Kunde kunde)` , um einen Kunden zur Liste hinzuzufügen.
- Schreibe eine Methode `listKunden()` , die alle Kunden in der Liste ausgibt.

Meilenstein 3: Mitarbeiter und Vererbung

Ziele:

1. Einführung von **Vererbung** und **Polymorphismus**.
2. Abstraktion durch ein gemeinsames Interface oder eine Basisklasse.

Aufgaben:

- Erstelle ein Interface `Person` mit folgenden Methoden:
 - `getId()`
 - `getName()`
 - `getEmail()`
- Lasse `Kunde` das Interface `Person` implementieren.
- Erstelle eine neue Klasse `Mitarbeiter` , die ebenfalls `Person` implementiert und zusätzlich ein Attribut `position` hat.

- Passe die bestehende Logik an, um sowohl Kunden als auch Mitarbeiter zu verwalten.

Meilenstein 4: Events und Orte

Ziele:

1. Einführung von **relationalen Objekten** (z. B. Ein Event hat Teilnehmer).
2. Arbeiten mit mehreren Klassen, die zusammenarbeiten.

Aufgaben:

- Erstelle eine Klasse `Ort` mit:
 - `id` (int): Eindeutige ID.
 - `name` (String): Name des Orts.
 - `capacity` (int): Maximale Teilnehmeranzahl.
- Erstelle eine Klasse `Event` mit:
 - `id` (int): Eindeutige ID.
 - `name` (String): Name des Events.
 - `ort` (Ort): Der zugehörige Ort.
 - `teilnehmer` (Liste von `Kunde`): Teilnehmer des Events.
- Schreibe Methoden in der `Event`-Klasse:
 - `addTeilnehmer(Kunde kunde)` : Fügt einen Teilnehmer hinzu.
 - `removeTeilnehmer(Kunde kunde)` : Entfernt einen Teilnehmer.
 - Prüfe, dass die Kapazität des Orts nicht überschritten wird.

Meilenstein 5: Validierung und Fehlerbehandlung

Ziele:

1. Einführung von **Validierungen** und **Exception Handling**.
2. Sicherstellen, dass das System robust und fehlerresistent ist.

Aufgaben:

- Implementiere eine Validierung für E-Mails:
 - E-Mail muss ein gültiges Format haben (z. B. `name@domain.com`).
 - E-Mail darf nicht doppelt vorkommen.
- Verwende **try-catch**, um Fehler wie ungültige Eingaben abzufangen.
- Füge eine benutzerdefinierte Exception `DuplicateEmailException` hinzu, die ausgelöst wird, wenn eine E-Mail bereits existiert.

Meilenstein 6: Benutzerinteraktion (Menü)

Ziele:

1. Einführung eines **textbasierten Menüs**.
2. Ermöglichen von Benutzerinteraktionen.

Aufgaben:

- Implementiere ein Menü mit folgenden Optionen:
 - i. Kunde hinzufügen.
 - ii. Mitarbeiter hinzufügen.
 - iii. Ort hinzufügen.
 - iv. Event erstellen.
 - v. Teilnehmer zu Event hinzufügen.
 - vi. Alle Events anzeigen.
 - vii. Alle Personen anzeigen.
 - viii. Programm beenden.
- Verarbeite Benutzereingaben mit einer `switch` -Anweisung.

Meilenstein 7: Erweiterte Funktionen

Ziele:

1. Hinzufügen von **Statistiken** und **Berichten**.
2. Verbesserte Suche und Filterfunktionen.

Aufgaben:

- **Statistiken:**
 - Zeige die Anzahl der Kunden, Mitarbeiter und Events an.
 - Zeige die durchschnittliche Teilnehmeranzahl pro Event an.
- **Suche:**
 - Suche nach Personen basierend auf deren Namen oder IDs.
 - Suche nach Events basierend auf Namen.

Zusätzliche Ideen

- **Speicherung:**
 - Speichere die Daten in einer Datei und lade sie beim Start des Programms.
- **Grafische Oberfläche (GUI):**
 - Verwende JavaFX oder Swing, um eine grafische Benutzeroberfläche zu erstellen.
- **Erweiterte Validierungen:**
 - Prüfe z. B., ob die Kapazität eines Orts beim Erstellen eines Events korrekt eingehalten wird.
- **Testabdeckung:**
 - Schreibe Unit-Tests für wichtige Funktionen, um sicherzustellen, dass das System korrekt funktioniert.

Beispiel für den Entwicklungsfortschritt

Meilenstein	Ergebnis
Meilenstein 1	Einzelner Kunde mit Name und E-Mail wird erfolgreich gespeichert.
Meilenstein 2	Mehrere Kunden können verwaltet und angezeigt werden.
Meilenstein 3	Einführung von Mitarbeitern und Vererbung.
Meilenstein 4	Events mit Teilnehmern und Kapazitätsprüfung implementiert.
Meilenstein 5	Validierungen und Fehlerbehandlung für E-Mails und IDs.
Meilenstein 6	Interaktives Menü für die Benutzerinteraktion.
Meilenstein 7	Erweiterte Statistiken und Suchfunktionen für Events und Personen.

Ziel

Nach Abschluss dieser Meilensteine hast du ein funktionales Event-Management-System entwickelt, das sowohl robust als auch modular ist. Dieses Projekt ermöglicht es dir, die grundlegenden und fortgeschrittenen Konzepte der Java-Programmierung zu erlernen und praktisch anzuwenden.