

Softwaretechnik II – Praktikum

Subsystem 4 – Zubereitung

Eine Dokumentation von:

J. Faßbender

J. Gobelet

L. Gobelet

E. Gödel

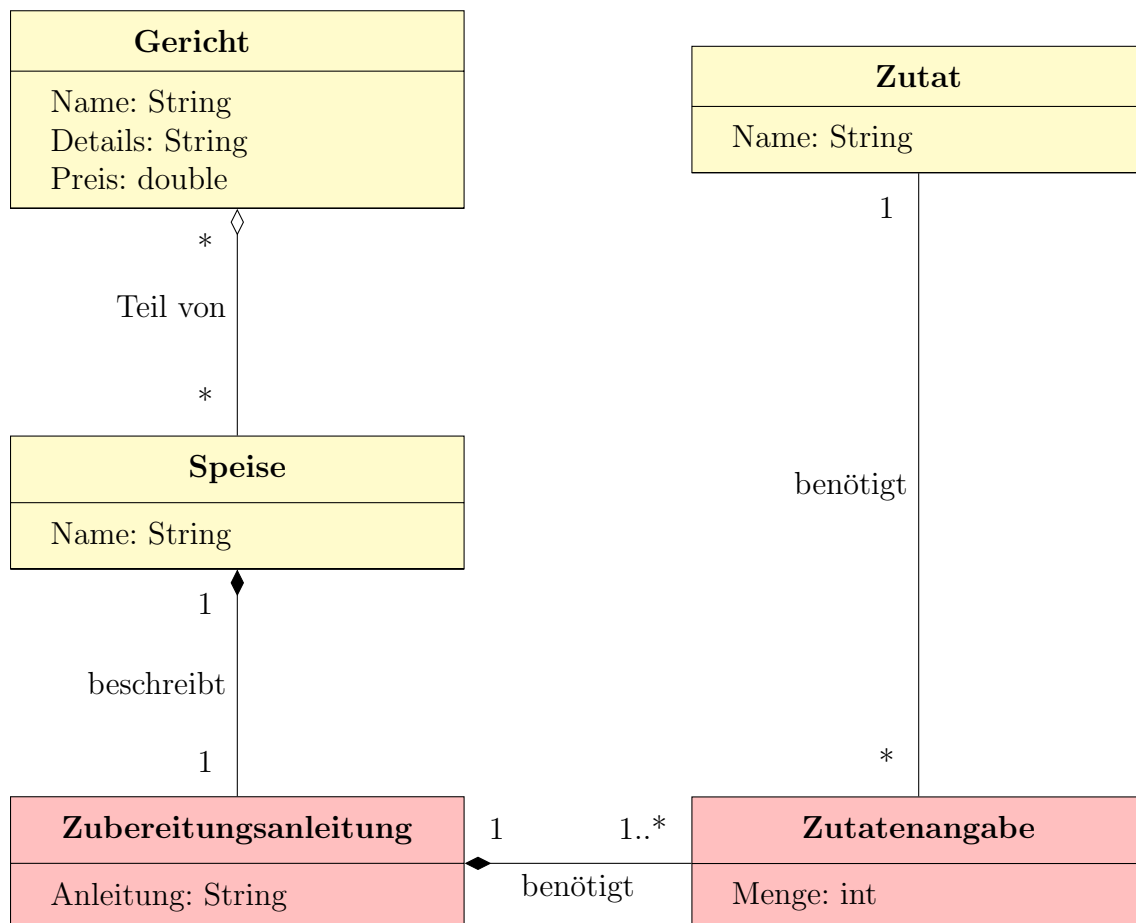
Inhaltsverzeichnis

1	Meilenstein 1 – Datenzugriffsschicht	3
1.1	Teilaufgabe 1: Ausschnitt aus Logischem DM mit Entities und Value Objects	3
1.1.1	Klassendiagramm	3
1.1.2	Fachliches Glossar	4
1.1.3	Erweiterungen der Aufgabenstellung	4
1.2	Teilaufgabe 2: Entities und Value Objects mit JPA-Annotierung	5
1.2.1	Annotationen der Entities und Value Objects	5
1.2.2	H2-Console	6

1 Meilenstein 1 – Datenzugriffsschicht

1.1 Teilaufgabe 1: Ausschnitt aus Logischem DM mit Entities und Value Objects

1.1.1 Klassendiagramm



■ : Value Object
■ : Entity

1.1.2 Fachliches Glossar

Geschäftsobjekt	Attribut	Erklärung
Gericht		Vom Restaurant angebotenes Mahl.
	Name	Gerichtsbezeichnung.
	Details	Wird dem Gast angezeigt. Enthält nähere Angaben zu den Zutaten.
	Preis	Geldbetrag der für das Gericht zu bezahlen ist.
Speise		Teil eines Gerichts. Beispielsweise wäre eine Salatbeilage als Speise zu verstehen.
	Name	Bezeichnung der Speise.
Zubereitungsanleitung		Leitfaden zur Zubereitung einer Speise.
	Anleitung	Erklärender Text, der beschreibt, wie eine Speise zuzubereiten ist.
Zutat		Benötigt für die Zubereitung einer Speise.
	Name	Bezeichnung der Zutat.
Zutatenangabe		Zuordnung zwischen Zutat und Zubereitungsanleitung. Gibt die Menge einer Zutat an, die für die Zubereitung notwendig ist.
	Menge	Die benötigte Menge.

1.1.3 Erweiterungen der Aufgabenstellung

Da es in unserem Logischen Datenmodell keine 1:1-Beziehung gab, haben wir eine zusätzliche redundante Entität eingebaut.

Hierbei handelt es sich um die Entität Speise. Diese Entität hätte genauso gut einfach Teil der Zubereitungsanleitung sein können und ist nur in unser Modell aufgenommen worden, damit wir die für die Aufgabenstellung benötigte 1:1-Beziehung in unserem Diagramm haben.

1.2 Teilaufgabe 2: Entities und Value Objects mit JPA-Annotierung

1.2.1 Annotationen der Entities und Value Objects

Gericht

```
@Entity
public class Gericht {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private int id;
    private String name;
    private String details;
    private double preis;

    // Ein Gericht besteht aus mehreren Speisen und eine Speise kann
    // mehreren Gerichten zugeordnet sein.
    @ManyToMany(cascade= CascadeType.ALL, fetch = FetchType.EAGER)
    @JoinTable(name = "gericht_speise",
        joinColumns = @JoinColumn(name = "gericht_id"),
        inverseJoinColumns = @JoinColumn(name = "speise_id")
    )
    private Set<Speise> speisen = new HashSet<Speise>();
}
```

Speise

```
@Entity
public class Speise {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private int id;
    private String name;

    // Zubereitungsanleitung als Value Object
    @Embedded
    private Zubereitungsanleitung anleitung;
}
```

Zubereitungsanleitung

```
@Embeddable
public class Zubereitungsanleitung {
    private String anleitung;
}
```

```
// Die Anleitung enthaelt mehrere Zutatenangaben als Value-Objects
@ElementCollection (targetClass = Zutatenangabe.class, fetch =
    FetchType.EAGER)
@CollectionTable(name = "ZUTATENANGABE")
private Set<Zutatenangabe> angaben = new HashSet<Zutatenangabe>();
```

Zutat

```
@Entity
public class Zutat {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private int id;
    private String name;
```

Zutatenangabe

```
@Embeddable
public class Zutatenangabe {
    private int menge;

    @ManyToOne(cascade = CascadeType.ALL)
    private Zutat zutat;
```

1.2.2 H2-Console