

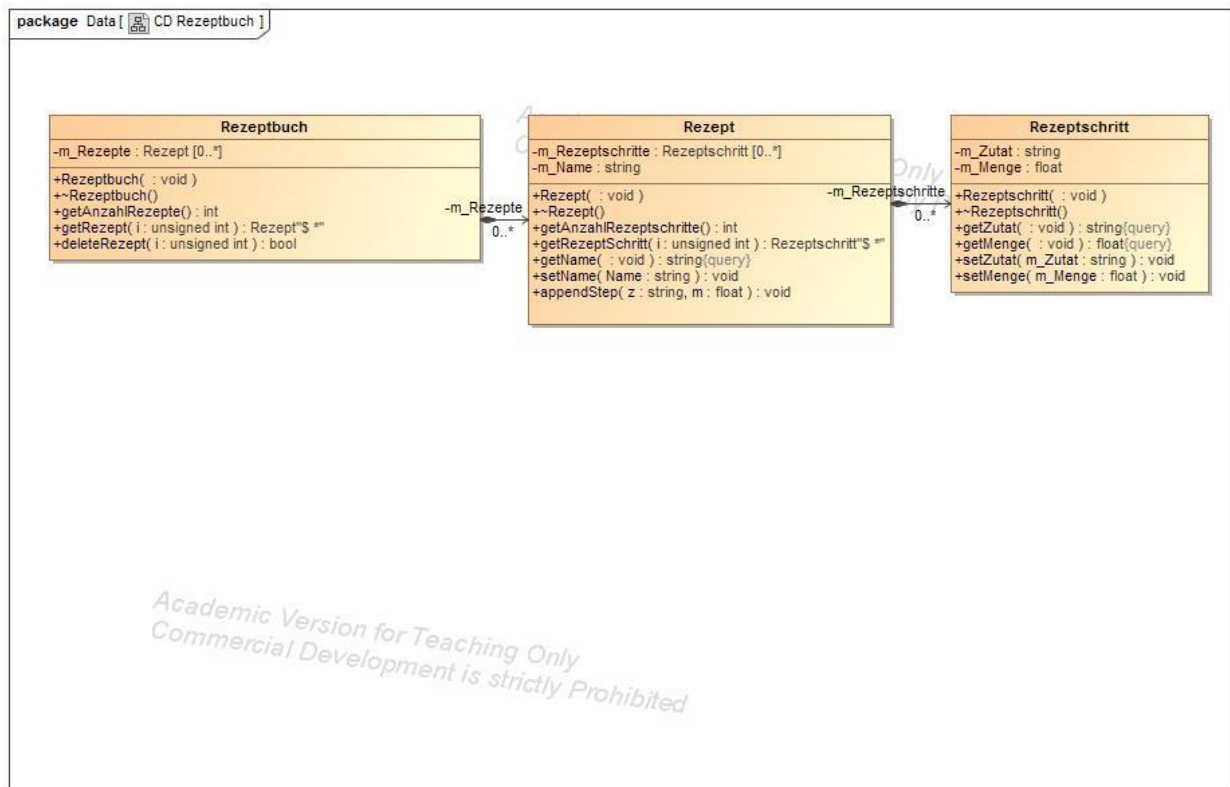
# Gesamtdokumentation

Author: Ralf Hahn  
Publisher: Hochschule Darmstadt  
Date: Sep 8, 2014 1:10:34 PM

Dies ist die automatische Gesamtdokumentation der UML Diagramme des Magicdraw Projektes.

## 1 Klassen

### CD Rezeptbuch



# Gesamtdokumentation

## 1.1 Rezeptschritt



| Klasse                 |   |
|------------------------|---|
| <b>Klassenname</b>     | Rezeptschritt   |
| <b>Dokumentation</b>   | Jeder Rezeptsschritt hat eine Zutat und eine Menge. Das wird frei interpretiert, d.h. eine Zutat kann auch "Mixen" sein. Die Menge bezieht sich dann auf die Zeit (20 = 20 Sekunden)  |
| <b>Abstrakt</b>        | false   |
| <b>Generalisierung</b> |   |
| <b>Attribute</b>       | private m_Zutat : string "" <ul style="list-style-type: none"> <li>Doku: Der Name der Zutat</li> </ul> private m_Menge : float "" <ul style="list-style-type: none"> <li>Doku: Die Menge der Zutat - das können Sekunden, Stück oder Gramm sein</li> </ul>  |
| <b>Operatoren</b>      | public Rezeptschritt ( : void ) : void <ul style="list-style-type: none"> <li>Doku: Konstruktor</li> </ul> public ~Rezeptschritt () : void <ul style="list-style-type: none"> <li>Doku: Destruktor</li> </ul> public getZutat ( : void ) : string <ul style="list-style-type: none"> <li>Doku: Liefert die Zutat des Rezeptschrittes</li> </ul> public getMenge ( : void ) : float <ul style="list-style-type: none"> <li>Doku: Liefert den Wert des Rezeptschrittes (die Menge)</li> </ul> public setZutat ( m_Zutat : string ) : void <ul style="list-style-type: none"> <li>Doku: Setzt die Zutat des Rezeptschrittes</li> </ul> public setMenge ( m_Menge : float ) : void <ul style="list-style-type: none"> <li>Doku: Setzt den Wert des Rezeptschrittes (die Menge)</li> </ul> |

## 1.2 Rezeptbuch



| Klasse               |  |
|----------------------|--|
| <b>Klassenname</b>   | Rezeptbuch   |
| <b>Dokumentation</b> | Das Rezeptbuch kennt eine Liste von Rezepten. Es bietet die Möglichkeit Rezepte zu |

# Gesamtdokumentation

|                        |  |
|------------------------|--|
|                        | suchen und abzufragen. Das Initialisieren der Rezepte erfolgt über den Dateileser.   |
| <b>Abstrakt</b>        | false  |
| <b>Generalisierung</b> |  |
| <b>Attribute</b>       | private m_Rezepte : Rezept "std::list<\$ *>" <ul style="list-style-type: none"> <li>Doku: Ein Rezeptbuch besteht aus vielen Rezepten - diese werden als eine Liste abgelegt.</li> </ul>  |
| <b>Operatoren</b>      | public Rezeptbuch ( : void ) : void <ul style="list-style-type: none"> <li>Doku: Konstruktor; legt temporär einen Initialisierer an, damit dieser die Liste auffüllt.</li> </ul><br>public ~Rezeptbuch ( ) : void <ul style="list-style-type: none"> <li>Doku: Destruktor. Löscht insbesondere den Initialisierer</li> </ul> public getAnzahlRezepte ( ) : int <ul style="list-style-type: none"> <li>Doku: Liefert die Anzahl der verfügbaren Rezepte.</li> </ul> public getRezept ( i : unsigned int ) : Rezept <ul style="list-style-type: none"> <li>Doku: Liefert das i-te Rezept aus der Liste (Werte von 0..n-1). Ansonsten NULL.</li> </ul><br>public deleteRezept ( i : unsigned int ) : bool <ul style="list-style-type: none"> <li>Doku: Löscht das i-te Rezept. Liefert bei Erfolg true, sonst false.</li> </ul> |

## 1.3 Rezept



| Klasse                 |   |
|------------------------|---|
| <b>Klassenname</b>     | Rezept  |
| <b>Dokumentation</b>   | Ein Rezept hat einen Namen und verwaltet eine beliebig lange Liste von Rezeptschritten.   |
| <b>Abstrakt</b>        | false   |
| <b>Generalisierung</b> |   |
| <b>Attribute</b>       | private m_Rezeptschritte : Rezeptschritt "std::list<\$ *>" <ul style="list-style-type: none"> <li>Doku: Jedes Rezept besteht aus vielen Rezeptschritten. Die werden hier (geordnet) in einer Liste abgelegt.</li> </ul><br>private m_Name : string "" <ul style="list-style-type: none"> <li>Doku: Der Name des Rezeptes</li> </ul> |

# Gesamtdokumentation

|                   |   |
|-------------------|---|
| <b>Operatoren</b> | <pre> public Rezept ( : void ) : void     • Doku: Konstruktor - legt ein Rezept an. public ~Rezept () : void     • Doku: Destruktor. Löscht ein Rezept inklusive der enthaltenen Rezeptschritte. public getAnzahlRezeptschritte ( ) : int     • Doku: Liefert die Anzahl der Rezeptschritte public getRezeptSchritt ( i : unsigned int ) : Rezeptschritt     • Doku: Liefert einen Verweis auf den i-ten Rezeptschritt (0..n-1).       Sonst wird eine NULL geliefert.  public getName ( : void ) : string     • Doku: Liefert den Namen des Rezeptes public setName ( Name : string ) : void     • Doku: Setzt den Namen des Rezeptes public appendStep ( z : string m : float ) : void     • Doku: Erzeugt einen neuen Rezeptschritt mit der übergebenen Zutat und der       Menge und hängt den an die Liste der Schritte an.           </pre> |
|-------------------|---|

## 2 Weitere Diagramme