Aufgabe Nr.1 - Entwicklung eines Warenwirtschaftssystems: Modellierung und Implementierung von Artikelklassen

3

Erzeugt von Doxygen 1.9.8

Chapter 1

Hierarchie-Verzeichnis

1.1 Klassenhierarchie

Die Liste der Ableitungen ist -mit Einschränkungen- alphabetisch sortiert:

Chapter 2

Klassen-Verzeichnis

2.1 Auflistung der Klassen

Hier folgt die Aufzählung aller Klassen, Strukturen, Varianten und Schnittstellen mit einer Kurzbeschreibung:

Chapter 3

Datei-Verzeichnis

3.1 Auflistung der Dateien

Hier folgt die Aufzählung aller dokumentierten Dateien mit einer Kurzbeschreibung:

Chapter 4

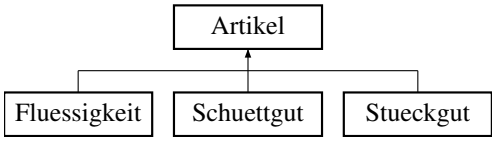
Klassen-Dokumentation

4.1 Artikel Klassenreferenz

Die Klasse "Artikel" repraesentiert einen Artikel mit verschiedenen Eigenschaften.

`#include <lager.hh>`

Klassendiagramm für Artikel:



Öffentliche Methoden

- Artikel (string name, string num, unsigned int bestand, masseinheit einheit, preis vp, preis np)

  Konstruktor fuer die Klasse "Artikel".
- ~**Artikel** ()

  Destruktor fuer die Klasse "Artikel".
- string getName () const

  Gibt den Namen des Artikels zurück.
- string getArtikelnummer () const

  Gibt die Artikelnummer des Artikels zurück.
- unsigned int getLagerabstand () const

  Gibt den Lagerbestand des Artikels zurück.
- string getMasseinheit () const

  Gibt die Masseinheit des Artikels zurück.
- preis getVerkaufpreis () const

  Gibt den Verkaufspreis des Artikels zurück.
- preis getNormpreis () const

  Gibt den Normalpreis des Artikels zurück.
- string getGruppe () const

  Gibt die Warengruppe des Artikels zurück.
- void setName (string name)

Setzt den Namen des Artikels.

- void setArtikelnummer (string num)

  Setzt die Artikelnummer des Artikels.
- void setLagerbestand (unsigned int bestand)

  Setzt den Lagerbestand des Artikels.
- void setMasseinheit (masseinheit einheit)

  Setzt die Masseinheit des Artikels.
- void setVerkaufpreis (preis vp)

  Setzt den Verkaufspreis des Artikels.
- void setNormpreis (preis np)

  Setzt den Normalpreis des Artikels.

Öffentliche, statische Methoden

- static void setGruppe (Warengruppen g)

  Setzt die Warengruppe fuer Artikel.

Statische öffentliche Attribute

- static Warengruppen **gruppe**

  Statische Warengruppen-Instanz, die fuer alle Artikel gemeinsam genutzt wird.

Geschützte Attribute

- string **artikelname**
- string **artikelnummer**
- unsigned int **lagerbestand**
- masseinheit **einheit**
- preis **verkaufpreis**
- preis **normpreis**

4.1.1 Ausführliche Beschreibung

Die Klasse "Artikel" repraesentiert einen Artikel mit verschiedenen Eigenschaften.

4.1.2 Beschreibung der Konstruktoren und Destruktoren

4.1.2.1 Artikel()

```
Artikel::Artikel (
            string name,
            string num,
            unsigned int bestand,
            masseinheit einheit,
            preis vp,
            preis np )
```

Konstruktor fuer die Klasse "Artikel".

Parameter

name	Der Name des Artikels.
num	Die Artikelnummer des Artikels.
bestand	Der Lagerbestand des Artikels.
einheit	Die Einheit des Artikels (stk, kg, l).
vp	Der Verkaufspreis des Artikels.
np	Der Normalpreis des Artikels.

4.1.3 Dokumentation der Elementfunktionen

4.1.3.1 getArtikelnummer()

```
string Artikel::getArtikelnummer ( ) const
```

Gibt die Artikelnummer des Artikels zurück.

Rückgabe

Die Artikelnummer des Artikels.

4.1.3.2 getGruppe()

```
string Artikel::getGruppe ( ) const
```

Gibt die Warengruppe des Artikels zurück.

Rückgabe

Die Warengruppe des Artikels oder die Artikelnummer, falls keine Warengruppe gefunden wurde.

4.1.3.3 getLagerabstand()

```
unsigned int Artikel::getLagerabstand ( ) const
```

Gibt den Lagerbestand des Artikels zurück.

Rückgabe

Der Lagerbestand des Artikels.

4.1.3.4 getMasseinheit()

```
string Artikel::getMasseinheit ( ) const
```

Gibt die Masseinheit des Artikels zurück.

Rückgabe

Die Masseinheit des Artikels (stk, kg, l).

4.1.3.5 getName()

```
string Artikel::getName ( ) const
```

Gibt den Namen des Artikels zurück.

Rückgabe

Der Name des Artikels.

4.1.3.6 getNormpreis()

```
preis Artikel::getNormpreis ( ) const
```

Gibt den Normalpreis des Artikels zurück.

Rückgabe

Der Normalpreis des Artikels.

4.1.3.7 getVerkaufpreis()

```
preis Artikel::getVerkaufpreis ( ) const
```

Gibt den Verkaufspreis des Artikels zurück.

Rückgabe

Der Verkaufspreis des Artikels.

4.1.3.8 setArtikelnummer()

```
void Artikel::setArtikelnummer (
            string num )
```

Setzt die Artikelnummer des Artikels.

Parameter

num	Die neue Artikelnummer des Artikels.

4.1.3.9 setGruppe()

```
void Artikel::setGruppe (
            Warengruppen g )  [static]
```

Setzt die Warengruppe fuer Artikel.

Parameter

g	Die Warengruppe, die zugewiesen werden soll.

4.1.3.10 setLagerbestand()

```
void Artikel::setLagerbestand (
            unsigned int bestand )
```

Setzt den Lagerbestand des Artikels.

Parameter

bestand	Der neue Lagerbestand des Artikels.

4.1.3.11 setMasseinheit()

```
void Artikel::setMasseinheit (
            masseinheit einheit )
```

Setzt die Masseinheit des Artikels.

Parameter

einheit	Die neue Masseinheit des Artikels (stk, kg, l).

4.1.3.12 setName()

```
void Artikel::setName (
            string name )
```

Setzt den Namen des Artikels.

Parameter

name	Der neue Name des Artikels.

4.1.3.13 setNormpreis()

```
void Artikel::setNormpreis (
            preis np )
```

Setzt den Normalpreis des Artikels.

Parameter

np	Der neue Normalpreis des Artikels.

4.1.3.14 setVerkaufpreis()

```
void Artikel::setVerkaufpreis (
            preis vp )
```

Setzt den Verkaufspreis des Artikels.

Parameter

vp	Der neue Verkaufspreis des Artikels.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Dateien:

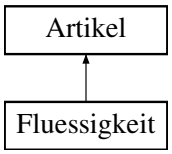
- /home/oliver/Programmieren_III/Aufgabe_1/src/lager.hh
- /home/oliver/Programmieren_III/Aufgabe_1/src/lager.cc

4.2 Fluessigkeit Klassenreferenz

Die Klasse "Fluessigkeit" erbt von der Klasse "Artikel" und spezialisiert sie fuer Fluessigkeits-Artikel.

```
#include <lager.hh>
```

Klassendiagramm für Fluessigkeit:



Öffentliche Methoden

- Fluessigkeit (string name, string num, double vol, preis np, unsigned int bestand=1)

  Konstruktor fuer die Klasse "Fluessigkeit".
- double getVolume () const

  Gibt das Volumen des Fluessigkeits-Artikels zurueck.
- void setVerkaufpreis (preis vp)
- void setVolume (double vol)

  Setzt das Volumen des Fluessigkeits-Artikels.

Öffentliche Methoden geerbt von Artikel

- Artikel (string name, string num, unsigned int bestand, masseinheit einheit, preis vp, preis np)

  Konstruktor fuer die Klasse "Artikel".
- ~**Artikel** ()

  Destruktor fuer die Klasse "Artikel".
- string getName () const

  Gibt den Namen des Artikels zurück.
- string getArtikelnummer () const

  Gibt die Artikelnummer des Artikels zurück.
- unsigned int getLagerabstand () const

  Gibt den Lagerbestand des Artikels zurück.
- string getMasseinheit () const

  Gibt die Masseinheit des Artikels zurück.
- preis getVerkaufpreis () const

  Gibt den Verkaufspreis des Artikels zurück.
- preis getNormpreis () const

  Gibt den Normalpreis des Artikels zurück.
- string getGruppe () const

  Gibt die Warengruppe des Artikels zurück.
- void setName (string name)

  Setzt den Namen des Artikels.
- void setArtikelnummer (string num)

  Setzt die Artikelnummer des Artikels.
- void setLagerbestand (unsigned int bestand)

  Setzt den Lagerbestand des Artikels.
- void setMasseinheit (masseinheit einheit)

  Setzt die Masseinheit des Artikels.
- void setVerkaufpreis (preis vp)

  Setzt den Verkaufspreis des Artikels.
- void setNormpreis (preis np)

  Setzt den Normalpreis des Artikels.

Weitere Geerbte Elemente

Öffentliche, statische Methoden geerbt von Artikel

- static void setGruppe (Warengruppen g)

  Setzt die Warengruppe fuer Artikel.

Statische öffentliche Attribute geerbt von Artikel

- static Warengruppen **gruppe**

  Statische Warengruppen-Instanz, die fuer alle Artikel gemeinsam genutzt wird.

Geschützte Attribute geerbt von Artikel

- string **artikelname**
- string **artikelnummer**
- unsigned int **lagerbestand**
- masseinheit **einheit**
- preis **verkaufpreis**
- preis **normpreis**

4.2.1 Ausführliche Beschreibung

Die Klasse "Fluessigkeit" erbt von der Klasse "Artikel" und spezialisiert sie fuer Fluessigkeits-Artikel.

4.2.2 Beschreibung der Konstruktoren und Destruktoren

4.2.2.1 Fluessigkeit()

```
Fluessigkeit::Fluessigkeit (
            string name,
            string num,
            double vol,
            preis np,
            unsigned int bestand = 1 )
```

Konstruktor fuer die Klasse "Fluessigkeit".

Parameter

name	Der Name des Fluessigkeits-Artikels.
num	Die Artikelnummer des Fluessigkeits-Artikels.
vol	Das Volumen des Fluessigkeits-Artikels.
np	Der Normalpreis des Fluessigkeits-Artikels.
bestand	Der Lagerbestand des Fluessigkeits-Artikels (Standardwert: 1).

4.2.3 Dokumentation der Elementfunktionen

4.2.3.1 getVolume()

```
double Fluessigkeit::getVolume ( ) const
```

Gibt das Volumen des Fluessigkeits-Artikels zurueck.

Rückgabe

Das Volumen des Artikels.

4.2.3.2 setVerkaufpreis()

```
void Fluessigkeit::setVerkaufpreis (
            preis vp )
```

Parameter

vp	Der Verkaufspreis, der gesetzt werden soll.

4.2.3.3 setVolume()

```
void Fluessigkeit::setVolume (
             double vol )
```

Setzt das Volumen des Fluessigkeits-Artikels.

Parameter

vol	Das neue Volumen.

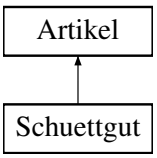
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Dateien:

- /home/oliver/Programmieren_III/Aufgabe_1/src/lager.hh
- /home/oliver/Programmieren_III/Aufgabe_1/src/lager.cc

4.3 Schuettgut Klassenreferenz

```
#include <lager.hh>
```

Klassendiagramm für Schuettgut:



Öffentliche Methoden

- Schuettgut (string name, string num, double groesse, preis np, unsigned int bestand=1)

  Konstruktor fuer die Klasse "Schuettgut".
- double getLosgroesse () const

  Gibt die Losgroeße des Schuettgut-Artikels zurueck.
- void setVerkaufpreis (preis vp)

  Setzt den Verkaufspreis des Schuettgut-Artikels basierend auf der Losgroeße.
- void setLosgroesse (double groesse)

  Setzt die Losgroeße des Schuettgut-Artikels.

Öffentliche Methoden geerbt von Artikel

- Artikel (string name, string num, unsigned int bestand, masseinheit einheit, preis vp, preis np)

  Konstruktor fuer die Klasse "Artikel".
- ~**Artikel** ()

  Destruktor fuer die Klasse "Artikel".
- string getName () const

  Gibt den Namen des Artikels zurück.
- string getArtikelnummer () const

  Gibt die Artikelnummer des Artikels zurück.
- unsigned int getLagerabstand () const

  Gibt den Lagerbestand des Artikels zurück.
- string getMasseinheit () const

  Gibt die Masseinheit des Artikels zurück.
- preis getVerkaufpreis () const

  Gibt den Verkaufspreis des Artikels zurück.
- preis getNormpreis () const

  Gibt den Normalpreis des Artikels zurück.
- string getGruppe () const

  Gibt die Warengruppe des Artikels zurück.
- void setName (string name)

  Setzt den Namen des Artikels.
- void setArtikelnummer (string num)

  Setzt die Artikelnummer des Artikels.
- void setLagerbestand (unsigned int bestand)

  Setzt den Lagerbestand des Artikels.
- void setMasseinheit (masseinheit einheit)

  Setzt die Masseinheit des Artikels.
- void setVerkaufpreis (preis vp)

  Setzt den Verkaufspreis des Artikels.
- void setNormpreis (preis np)

  Setzt den Normalpreis des Artikels.

Weitere Geerbte Elemente

Öffentliche, statische Methoden geerbt von Artikel

- static void setGruppe (Warengruppen g)

  Setzt die Warengruppe fuer Artikel.

Statische öffentliche Attribute geerbt von Artikel

- static Warengruppen **gruppe**

  Statische Warengruppen-Instanz, die fuer alle Artikel gemeinsam genutzt wird.

Geschützte Attribute geerbt von Artikel

- string **artikelname**
- string **artikelnummer**
- unsigned int **lagerbestand**
- masseinheit **einheit**
- preis **verkaufpreis**
- preis **normpreis**

4.3.1 Ausführliche Beschreibung

Die Klasse "Schuettgut" erbt von der Klasse "Artikel" und spezialisiert sie fuer Schuettgut-Artikel.

4.3.2 Beschreibung der Konstruktoren und Destruktoren

4.3.2.1 Schuettgut()

```
Schuettgut::Schuettgut (
            string name,
            string num,
            double groesse,
            preis np,
            unsigned int bestand = 1 )
```

Konstruktor fuer die Klasse "Schuettgut".

Parameter

name	Der Name des Schuettgut-Artikels.
num	Die Artikelnummer des Schuettgut-Artikels.
groesse	Die Losgroeße des Schuettgut-Artikels.
np	Der Normalpreis des Schuettgut-Artikels.
bestand	Der Lagerbestand des Schuettgut-Artikels (Standardwert: 1).

4.3.3 Dokumentation der Elementfunktionen

4.3.3.1 getLosgroesse()

```
double Schuettgut::getLosgroesse ( ) const
```

Gibt die Losgroeße des Schuettgut-Artikels zurueck.

Rückgabe

Die Losgroeße des Artikels.

4.3.3.2 setLosgroesse()

```
void Schuettgut::setLosgroesse (
            double groesse )
```

Setzt die Losgroeße des Schuettgut-Artikels.

Parameter

groesse	Die neue Losgroeße.

4.3.3.3 setVerkaufpreis()

```
void Schuettgut::setVerkaufpreis (
            preis vp )
```

Setzt den Verkaufspreis des Schuettgut-Artikels basierend auf der Losgroeße.

Parameter

vp	Der Verkaufspreis, der gesetzt werden soll.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Dateien:

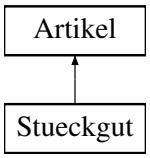
- /home/oliver/Programmieren_III/Aufgabe_1/src/lager.hh
- /home/oliver/Programmieren_III/Aufgabe_1/src/lager.cc

4.4 Stueckgut Klassenreferenz

Die Klasse "Stueckgut" erbt von der Klasse "Artikel" und spezialisiert sie fuer Stueckgut-Artikel.

```
#include <lager.hh>
```

Klassendiagramm für Stueckgut:



Öffentliche Methoden

- Stueckgut (string name, string num, preis vp, unsigned int bestand=1)

  Konstruktor fuer die Klasse "Stueckgut".

Öffentliche Methoden geerbt von Artikel

- Artikel (string name, string num, unsigned int bestand, masseinheit einheit, preis vp, preis np)

  Konstruktor fuer die Klasse "Artikel".
- ~**Artikel** ()

  Destruktor fuer die Klasse "Artikel".
- string getName () const

  Gibt den Namen des Artikels zurück.
- string getArtikelnummer () const

  Gibt die Artikelnummer des Artikels zurück.
- unsigned int getLagerabstand () const

  Gibt den Lagerbestand des Artikels zurück.
- string getMasseinheit () const

  Gibt die Masseinheit des Artikels zurück.
- preis getVerkaufpreis () const

  Gibt den Verkaufspreis des Artikels zurück.
- preis getNormpreis () const

  Gibt den Normalpreis des Artikels zurück.
- string getGruppe () const

  Gibt die Warengruppe des Artikels zurück.
- void setName (string name)

  Setzt den Namen des Artikels.
- void setArtikelnummer (string num)

  Setzt die Artikelnummer des Artikels.
- void setLagerbestand (unsigned int bestand)

  Setzt den Lagerbestand des Artikels.
- void setMasseinheit (masseinheit einheit)

  Setzt die Masseinheit des Artikels.
- void setVerkaufpreis (preis vp)

  Setzt den Verkaufspreis des Artikels.
- void setNormpreis (preis np)

  Setzt den Normalpreis des Artikels.

Weitere Geerbte Elemente

Öffentliche, statische Methoden geerbt von Artikel

- static void setGruppe (Warengruppen g)

  Setzt die Warengruppe fuer Artikel.

Statische öffentliche Attribute geerbt von Artikel

- static Warengruppen **gruppe**

  Statische Warengruppen-Instanz, die fuer alle Artikel gemeinsam genutzt wird.

Geschützte Attribute geerbt von Artikel

- string **artikelname**
- string **artikelnummer**
- unsigned int **lagerbestand**
- masseinheit **einheit**
- preis **verkaufpreis**
- preis **normpreis**

4.4.1 Ausführliche Beschreibung

Die Klasse "Stueckgut" erbt von der Klasse "Artikel" und spezialisiert sie fuer Stueckgut-Artikel.

4.4.2 Beschreibung der Konstruktoren und Destruktoren

4.4.2.1 Stueckgut()

```
Stueckgut::Stueckgut (
            string name,
            string num,
            preis vp,
            unsigned int bestand = 1 )
```

Konstruktor fuer die Klasse "Stueckgut".

Parameter

name	Der Name des Stueckgut-Artikels.
num	Die Artikelnummer des Stueckgut-Artikels.
vp	Der Verkaufspreis des Stueckgut-Artikels.
bestand	Der Lagerbestand des Stueckgut-Artikels (Standardwert: 1).

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Dateien:

- /home/oliver/Programmieren_III/Aufgabe_1/src/lager.hh
- /home/oliver/Programmieren_III/Aufgabe_1/src/lager.cc

4.5 Warengruppen Klassenreferenz

Öffentliche Methoden

- Warengruppen ()

  Konstruktor fuer die Klasse "Warengruppen".
- void **defaultList** ()

  Setzt eine Standard-Warengruppenliste (nicht implementiert).
- string getGruppe (string code)

  Gibt den Namen der Warengruppe fuer einen gegebenen Code zurueck.

- void addGruppe (string code, string name)

  Fuegt eine neue Warengruppe hinzu.
- void delGruppe (string code)

  Loescht eine Warengruppe anhand ihres Codes.
- void changeGruppe (string code, string name)

  Aendert den Namen einer vorhandenen Warengruppe.
- void **clear** ()

  Loescht alle Warengruppen und setzt sie zurck.

4.5.1 Beschreibung der Konstruktoren und Destruktoren

4.5.1.1 Warengruppen()

```
Warengruppen::Warengruppen ( )
```

Konstruktor fuer die Klasse "Warengruppen".

Dieser Konstruktor initialisiert eine leere Warengruppenliste.

4.5.2 Dokumentation der Elementfunktionen

4.5.2.1 addGruppe()

```
void Warengruppen::addGruppe (
            string code,
            string name )
```

Fuegt eine neue Warengruppe hinzu.

Parameter

code	Der Warengruppencode.
name	Der Name der Warengruppe.

4.5.2.2 changeGruppe()

```
void Warengruppen::changeGruppe (
            string code,
            string name )
```

Aendert den Namen einer vorhandenen Warengruppe.

Parameter

code	Der Warengruppencode.
name	Der neue Name der Warengruppe.

4.5.2.3 delGruppe()

```
void Warengruppen::delGruppe (
            string code )
```

Loescht eine Warengruppe anhand ihres Codes.

Parameter

code	Der Warengruppencode.

4.5.2.4 getGruppe()

```
string Warengruppen::getGruppe (
            string code )
```

Gibt den Namen der Warengruppe fuer einen gegebenen Code zurueck.

Parameter

code	Der Warengruppencode.

Rückgabe

Der Name der Warengruppe oder der Code, falls keine Warengruppe gefunden wurde.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Dateien:

- /home/oliver/Programmieren_III/Aufgabe_1/src/lager.hh
- /home/oliver/Programmieren_III/Aufgabe_1/src/lager.cc

Chapter 5

Datei-Dokumentation

5.1 /home/oliver/Programmieren_III/Aufgabe_1/src/lager.cc-Dateireferenz

Implementierung der Lagerverwaltungsfunktionen.

```
#include "lager.hh"
#include <cmath>
#include <iostream>
#include <map>
#include <string>
```

5.1.1 Ausführliche Beschreibung

Implementierung der Lagerverwaltungsfunktionen.

Autoren

Yaman Alsaady, Oliver Schmidt

Version

0.1

Datum

2023-10-04

Dies ist die Implementierung der Funktionen fuer die Lagerverwaltung, einschließlich der Warengruppenverwaltung und der Artikelklassen.

Copyright

5.2 /home/oliver/Programmieren_III/Aufgabe_1/src/lager.hh-↩ Dateireferenz

Dieses Header-Datei enthaelt die Definitionen von Klassen und Funktionen zur Verwaltung von Artikeln und Warengruppen in einem C++-Programm.

```
#include <iostream>
#include <map>
#include <string>
```

Klassen

- class Warengruppen
- class Artikel

  Die Klasse "Artikel" repraesentiert einen Artikel mit verschiedenen Eigenschaften.
- class Stueckgut

  Die Klasse "Stueckgut" erbt von der Klasse "Artikel" und spezialisiert sie fuer Stueckgut-Artikel.
- class Schuettgut
- class Fluessigkeit

  Die Klasse "Fluessigkeit" erbt von der Klasse "Artikel" und spezialisiert sie fuer Fluessigkeits-Artikel.

Typdefinitionen

- typedef double **preis**

Aufzählungen

- enum **masseinheit** { **stk** , **kg** , **l** }

5.2.1 Ausführliche Beschreibung

Dieses Header-Datei enthaelt die Definitionen von Klassen und Funktionen zur Verwaltung von Artikeln und Warengruppen in einem C++-Programm.

Autoren

Yaman Alsaady, Oliver Schmidt

Version

0.1

Datum

2023-10-04

Copyright

5.3 lager.hh

gehe zur Dokumentation dieser Datei

```
00001
00015 #ifndef LAGER_HH
00016 #define LAGER_HH
00017
00018 #include <iostream>
00019 #include <map>
00020 #include <string>
00021
00022 using namespace std;
00023 enum masseinheit { stk, kg, l };
00024 typedef double preis;
00025
00026 class Warengruppen {
00027 private:
00028   map<string, string> mapGruppe;
00029   map<string, string>::iterator iter;
00030
00031 public:
00032
00038   Warengruppen();
00039
00043   void defaultList();
00044
00051   string getGruppe(string code);
00052
00059   void addGruppe(string code, string name);
00060
00066   void delGruppe(string code);
00067
00074   void changeGruppe(string code, string name);
00075
00079   void clear();
00080 };
00081
00085 class Artikel {
00086 protected:
00087   string artikelname;
00088   string artikelnummer;
00089   unsigned int lagerbestand;
00090   masseinheit einheit;
00091   preis verkaufpreis;
00092   preis normpreis;
00093
00094 public:
00095
00106   Artikel(string name, string num, unsigned int bestand, masseinheit einheit,
00107           preis vp, preis np);
00108
00109
00110   // Getter-Funktionen
00111
00115   ~Artikel();
00116
00120   static Warengruppen gruppe;
00121
00127   static void setGruppe(Warengruppen g);
00128
00134 string getName() const;
00135
00141 string getArtikelnummer() const;
00142
00148 unsigned int getLagerabstand() const;
00149
00155 string getMasseinheit() const;
00156
00162 preis getVerkaufpreis() const;
00163
00169 preis getNormpreis() const;
00170
00176 string getGruppe() const;
00177
00178
00179 // Setter-Funktionen
00180
00186 void setName(string name);
00187
00193 void setArtikelnummer(string num);
00194
00200 void setLagerbestand(unsigned int bestand);
00201
00207 void setMasseinheit(masseinheit einheit);
00208
```

```
00214 void setVerkaufpreis(preis vp);
00215
00221 void setNormpreis(preis np);
00222
00223 };
00224
00228 class Stueckgut : public Artikel {
00229 private:
00230 public:
00239   Stueckgut(string name, string num, preis vp, unsigned int bestand = 1);
00240 };
00241
00245 class Schuettgut : public Artikel {
00246 private:
00247   double losgroesse;
00248
00249 public:
00259   Schuettgut(string name, string num, double groesse, preis np,
00260              unsigned int bestand = 1);
00266   double getLosgroesse() const;
00267
00273   void setVerkaufpreis(preis vp);
00274
00280   void setLosgroesse(double groesse);
00281 };
00282
00286 class Fluessigkeit : public Artikel {
00287 private:
00288   double volume;
00289 public:
00299   Fluessigkeit(string name, string num, double vol, preis np,
00300              unsigned int bestand = 1);
00301
00307   double getVolume() const;
00308
00313   void setVerkaufpreis(preis vp);
00314
00320   void setVolume(double vol);
00321 };
00322
00323 #endif // !LAGER_HH
```

5.4 /home/oliver/Programmieren_III/Aufgabe_1/src/main.cc-Dateireferenz

Hauptprogramm fuer Lagerverwaltung.

```
#include "lager.hh"
#include <iostream>
#include <string>
```

Funktionen

- void printInfo (Artikel produkt)

  Zeigt Informationen zu einem Artikel an.
- void printInfo (Schuettgut produkt)

  Zeigt Informationen zu einem Schuettgut an.
- void **printInfo** (Fluessigkeit produkt)
- int **main** ()

5.4.1 Ausführliche Beschreibung

Hauptprogramm fuer Lagerverwaltung.

Autoren

Yaman Alsaady, Oliver Schmidt

Version

0.1

Datum

2023-10-04

Dieses Programm dient zur Verwaltung von Lagerbestaenden verschiedener Produkte. Es erstellt Produkte unterschiedlicher Typen und zeigt Informationen zu ihnen an.

Copyright

5.4.2 Dokumentation der Funktionen

5.4.2.1 printInfo() [1/2]

```
void printInfo (
            Artikel produkt )
```

Zeigt Informationen zu einem Artikel an.

Parameter

produkt	Der Artikel, dessen Informationen angezeigt werden sollen.

5.4.2.2 printInfo() [2/2]

```
void printInfo (
            Schuettgut produkt )
```

Zeigt Informationen zu einem Schuettgut an.

Parameter

produkt	Das Schuettgut, dessen Informationen angezeigt werden sollen.