

Szenario

Sie sollen für den örtlichen Tante-Emma-Laden eine neue Anwendungssoftware mit den folgenden Funktionen entwickeln:

- Informationen zu allen Waren des Tante-Emma-Ladens anzeigen: *Bezeichnung, Warengruppe, Preis, Einheit, Verfügbare Menge*
- Informationen zu einem Warenkorb anzeigen: *Warenkorbeinträge, Gesamtsumme des Warenkorbs*
- Informationen zu allen Warenkorbeinträgen eines Warenkorbs anzeigen: *Index, Ware, Preis, Einheit, Menge, Zwischensumme*
- Waren zum Warenkorb hinzufügen
- Zwischensummen und Gesamtsumme eines Warenkorbs berechnen und anzeigen
- Kauf abschließen

Sie haben bereits das Design der Anwendungssoftware erstellt (Klassendiagramm und Oberflächen-Entwürfe) und sollen sich nun an die Umsetzung des Designs machen.

Wörterbuch

CornerShop: Tante-Emma-Laden

ShoppingCart: Warenkorb

ShoppingCartItem: Warenkorbeintrag

Product: Produkt

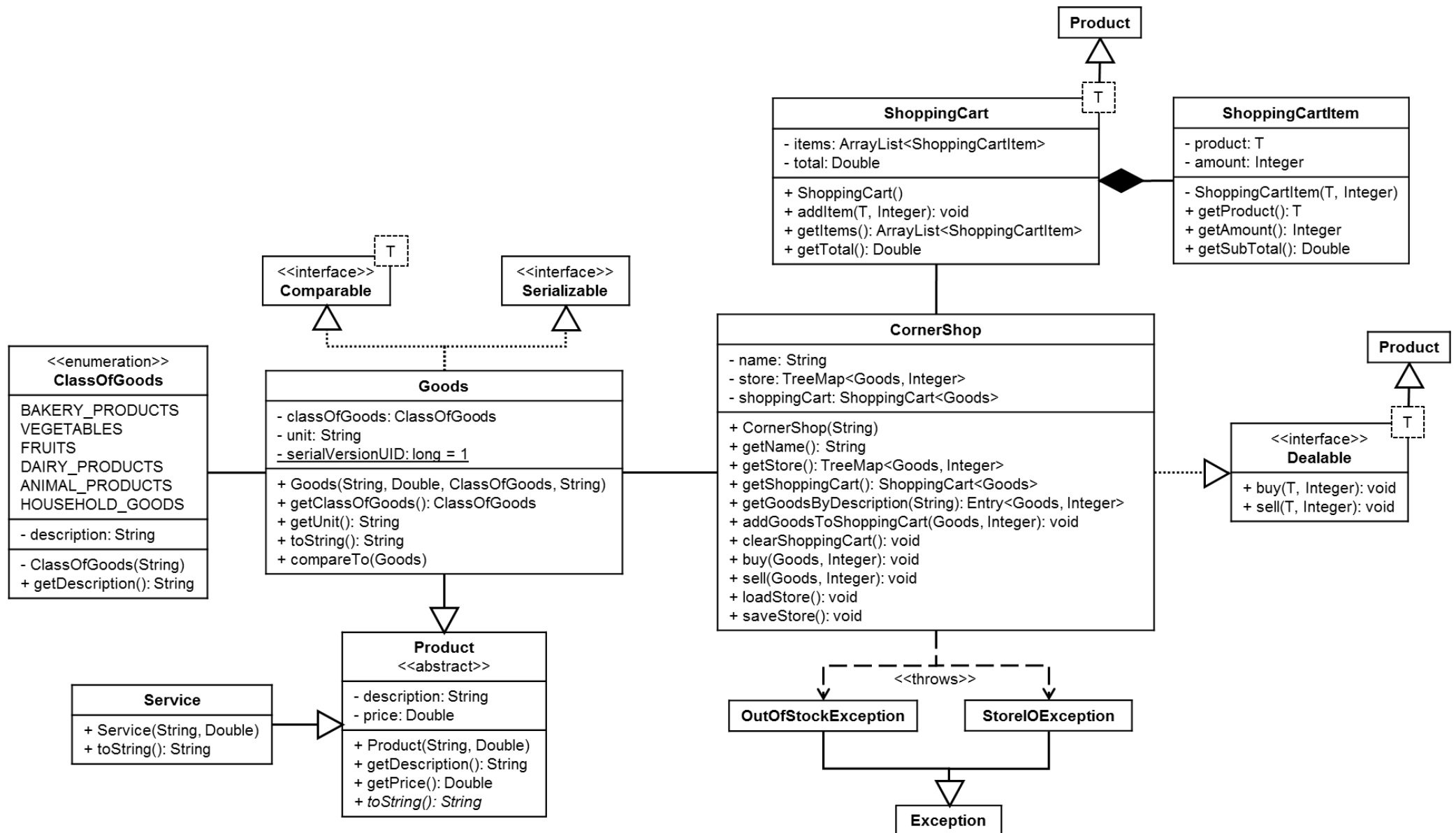
Goods: Ware(n)

Service: Dienstleistung

ClassOfGoods: Warengruppe

Deable: handelbar

Klassendiagramm zum Szenario



Auszug aus den Oberflächen-Entwürfen zum Szenario

CornerShopFrame

Tante-Emma-Laden

Waren				
Bezeichnung	Warengruppe	Preis	Einheit	verfügbare Me...
Brot	Backwaren	2,49 €	Laib	40
Butter	Milchprodukte	1,49 €	Päckchen	60
Eier	Milchprodukte	0,19 €	12er-Packung	240
Kartoffeln	Gemüse	3,49 €	1kg-Packung	5
Milch	Milchprodukte	0,79 €	Liter	50
Toilettenpapier	Haushaltsarti...	2,39 €	Packung	20

Warenkorb					
Index	Ware	Preis	Einheit	Menge	Zwischens...
1	Brot	2,49 €	Laib	1	2,49 €
2	Butter	1,49 €	Päckchen	2	2,98 €
3	Eier	0,19 €	12er-Packu...	1	0,19 €
4	Kartoffeln	3,49 €	1kg-Packung	1	3,49 €

Gesamtsumme: 22,77 €

Ware zum Warenkorb hinzufügen

Kauf abschließen

Quellcode zum Quicksort

```
public static int[] quickSort(int[] data) {
    return sort(data, 0, data.length - 1);
}

public static int[] sort(int[] data, int left, int right) {
    int center = left + (right - left) / 2;
    int pivot = data[center];
    int i = left, j = right;
    while (i <= j) {
        while (data[i] < pivot) {
            i++;
        }
        while (data[j] > pivot) {
            j--;
        }
        if (i <= j) {
            int temp = data[i];
            data[i] = data[j];
            data[j] = temp;
            i++;
            j--;
        }
    }
    if (left < j) {
        sort(data, left, j);
    }
    if (right > i) {
        sort(data, i, right);
    }
    return data;
}
```