

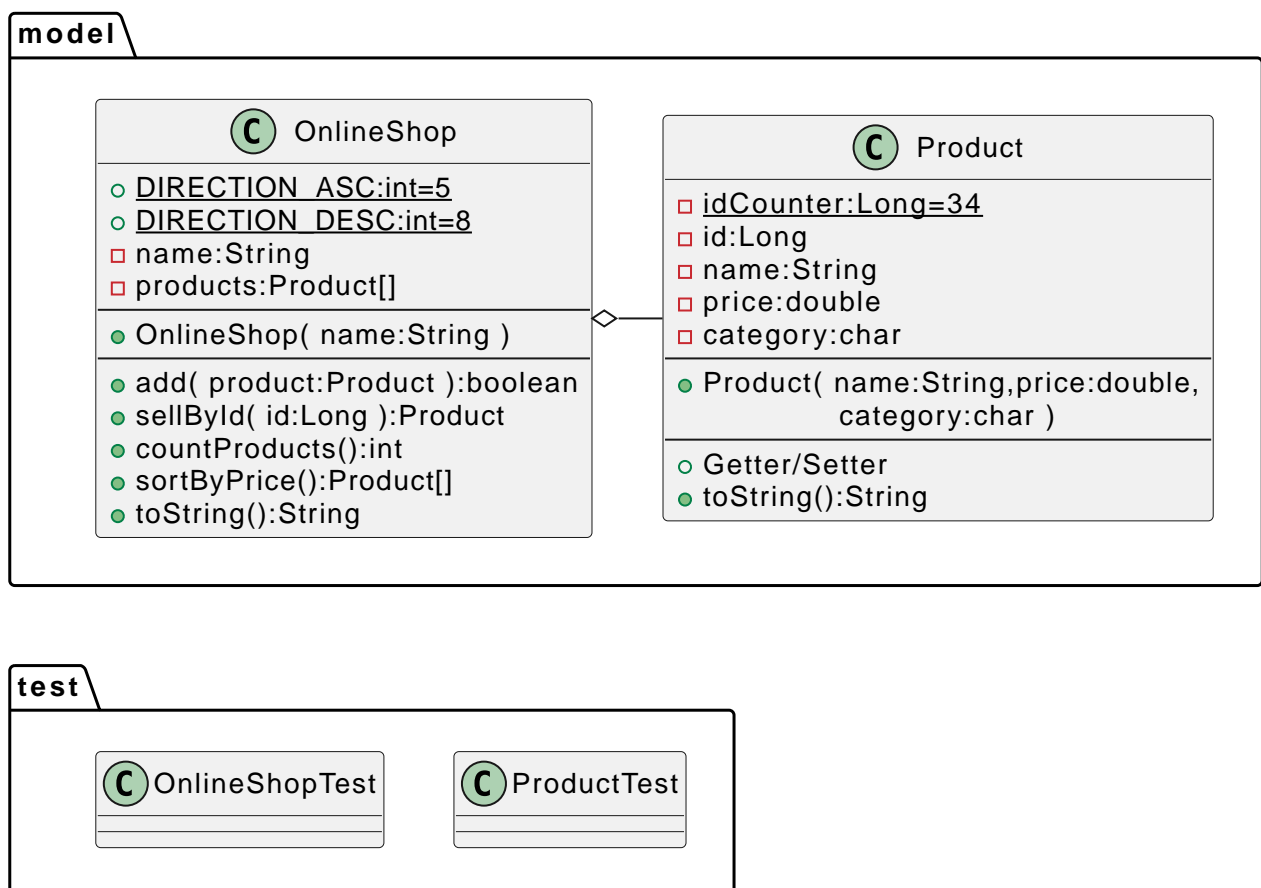
WARNING

- Bitte achten Sie darauf, nur kompilierfähige Programme abzugeben. Sollten Sie einen Compilerfehler nicht beheben können, kommentieren Sie das entsprechende Codestück aus. Nicht kompilierfähige Programme werden **negativ** bewertet!
- Halten Sie sich **exakt** an die Vorgaben, Formate sowie Klassen- und Methodennamen.

Beschreibung

Die beiden Unternehmer *Ab Zocker* und *Gaune Rei* benötigen für ihr neues Startup-Unternehmen einen Prototyp für einen Onlinehandel, bei dem unterschiedliche Produkte verwaltet und zum Kauf angeboten werden können. Entwickeln Sie die Software auf Basis des UML-Diagramms und den gegebenen Anforderungen.

UML-Diagramm



Prolog

- Angabe auf dem **Z: Laufwerk** entpacken.
- Entpacktes Projekt `s4-plf-001_online_shop` auf den Desktop kopieren.

- **ALLE** Dateien und Verzeichnisse vom **Z: Laufwerk** löschen.
- Projektordner `s4-plf-001_online_shop` mit IntelliJ öffnen.

1. Aufgabe - Product

Implementieren Sie die Model-Klasse **Product** gemäß UML-Diagramm und Klassenbeschreibung. Sie können die vorhandenen Testklassen für JUnit-Tests verwenden.

Table 1. Klassenbeschreibung

Attribut/Methode	Beschreibung										
idCounter	<ul style="list-style-type: none">• Counter, der für jede erzeugte Product-Instanz eine eindeutige Id vergibt.• Der Counter soll beim Wert 34 starten.										
id	<ul style="list-style-type: none">• Jede Product-Instanz muss vom idCounter eine eindeutige Id erhalten, die nach ihrer Zuweisung nicht mehr verändert werden darf.										
name	<ul style="list-style-type: none">• Repräsentiert den Namen eines Produktes.										
price	<ul style="list-style-type: none">• Preis des Produktes in Euro.										
category	<ul style="list-style-type: none">• Die Kategorie eines Produktes darf folgende Werte haben:<table><tr><th>char</th><th>Kategorie-Bezeichnung</th></tr><tr><td>m</td><td>Media</td></tr><tr><td>f</td><td>Food</td></tr><tr><td>c</td><td>Cosmetics</td></tr><tr><td>o</td><td>Office Equipment</td></tr></table>• Wird ein falscher Wert übergeben, so soll eine IllegalArgumentException geworfen werden.	char	Kategorie-Bezeichnung	m	Media	f	Food	c	Cosmetics	o	Office Equipment
char	Kategorie-Bezeichnung										
m	Media										
f	Food										
c	Cosmetics										
o	Office Equipment										

Attribut/Methode	Beschreibung
toString String	<ul style="list-style-type: none"> Die Methode gibt einen String über sämtliche Informationen eines Produktes zurück: <div data-bbox="568 277 1458 376" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> ID, NAME, PRICE €, CATEGORY </div> <ul style="list-style-type: none"> Die Kategorie soll dabei ausgeschrieben werden. <p>Beispiel:</p> <div data-bbox="518 555 1458 967" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> 34, Smo-King Kaltrauchgenerator Big-Grill-SM0, 139.00 €, Food 35, WURSTBARON® Premium Tomahawk Steak vom Jungbullen, 49.99 €, Food 36, NYX Professional Makeup Epic Ink Eye Liner, 10.02 €, Cosmetics 37, Smoofl Hundeeis, Premium Hundesnacks, 13.46 €, Food 38, Bleistiftanspitzer Miauende Katze, 15.94 €, Office Equipment </div>

2. Aufgabe - OnlineShop

Implementieren Sie die Klasse `OnlineShop` gemäß UML-Diagramm und Klassenbeschreibung. Sie können die vorhandenen Testklassen für JUnit-Tests verwenden.

Table 2. Klassenbeschreibung







Attribut/Methode	Beschreibung
<code>name</code>	Der Name des Onlineshops.
<code>products</code>	<ul style="list-style-type: none">• Alle Produkte sollen im Array <code>products</code> vom Typ <code>Product[]</code> gespeichert werden.• Insgesamt sollen 5 Produkte gespeichert werden.• Achten Sie genau darauf, dass keine Methode eine Referenz auf das Array zurückliefert.
<code>add(product)</code> <code>boolean</code>	<ul style="list-style-type: none">• Die Methode fügt ein neues Produkt in den Onlineshop hinzu.• Das Produkt darf nicht doppelt aufgenommen werden.• Wenn ein Produkt aufgenommen wurde, so soll die Methode <code>true</code> zurückliefern.• Im Fehlerfall bzw. wenn ein Produkt nicht aufgenommen wurde, so soll die Methode <code>false</code> zurückliefern.
<code>sellById(id)</code> <code>Product</code>	<ul style="list-style-type: none">• Die Methode dient zum Verkaufen eines Produktes.• Wenn das Produkt mit der übergebenen <code>id</code> im Shop vorhanden ist, so soll es aus dem Array entfernt werden. Geben Sie das entfernte Objekt zurück.• Ist das Produkt mit der gesuchten <code>id</code> nicht vorhanden, so soll der Wert <code>null</code> zurückgegeben werden.
<code>countProducts()</code> <code>int</code>	<ul style="list-style-type: none">• Zählt alle im Shop vorhandenen Produkte und liefert den Wert zurück.
<code>sortByPrice()</code> <code>Product[]</code>	<ul style="list-style-type: none">• Sortiert alle Produkte, aufsteigend, nach deren Preis.• Die Methode soll ein neues Array mit den sortierten Werten zurückliefern.• Alle <code>null</code> Werte sollen an das Ende des Arrays sortiert werden.• Das bestehende Array <code>products</code> soll unsortiert bleiben. <p>Hinweis</p> <p>Kopieren Sie die Daten vor dem Sortieren zuerst in ein neues Array mit der gleichen Größe wie <code>products</code>.</p>

Attribut/Methode	Beschreibung
<code>toString()</code> <code>String</code>	<ul style="list-style-type: none"> Liefert detaillierte Informationen über den Shop zurück: <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> Shopbezeichnung: Joe's Milchbar Vorhandene Produkte: 34, Smo-King Kaltrauchgenerator Big-Grill-SMO 1,25 Liter, 139.00 €, Food ... </div> <ul style="list-style-type: none"> Verwenden Sie eine Instanz der Klasse <code>StringBuilder</code> um die Informationen der Produkte zu sammeln.
BONUSAUFGABE <code>sortByPrice(direction)</code>	<ul style="list-style-type: none"> Erweitern Sie die Methode <code>sortByPrice()</code> um den Parameter <code>direction</code>. Wenn ein falscher Wert übergeben wird, muss eine <code>IllegalArgumentException</code> geworfen werden. Der Parameter <code>direction</code> gibt die Sortierreihenfolge an: <p>Aufsteigend</p> <ul style="list-style-type: none"> Wenn der Wert gleich wie dem Wert der Konstante <code>DIRECTION_ASC</code> ist. Wenn ein ungültiger Wert übergeben wurde. <p>Absteigend</p> <ul style="list-style-type: none"> Wenn der Wert gleich dem Wert der Konstante <code>DIRECTION_DESC</code> ist. <ul style="list-style-type: none"> Alle <code>null</code> Werte sollen bei beiden Varianten an das Ende des Arrays sortiert werden.

3. Aufgabe - JUnit

Gegeben sei die Klasse `Pet` im package `test`. Implementieren Sie die JUnit Testklasse `PetTest` im richtigen Ordner.

- Analysieren Sie vor der Implementierung die Klasse `Pet`.
- Implementieren Sie **einen** JUnit-Testfall für die Methode `setName` in der Testklasse `PetTest`, der **nicht** die Funktion testet.

 Pet
 name : String
 Pet(name:String)
 getName():String
 setName(name:String):void
 toString():String

4. Aufgabe - Abgabeverfahren

- Projektordner `s4-plf-001_online_shop` mit dem Windows Explorer öffnen.
 - Datei `abgabe.cmd` ausführen.
 - **Z: Laufwerk** öffnen und Abgabe kontrollieren.
 - Vom Prüfungsrechner **abmelden**.
-

Viel Erfolg!!!