

class Shop
+ produkte: [Produkt] + kunden: [Kunde] + status: ShopStatus
+ produkteAnzeigen() + menuAnzeigen() + shoppingStart()

struct Warenkorb
+ produkte: [Produkt]
+ mutating hinzufuegen() + anzeigen() + gesamtpreis()

class Kunde
+ kundenNr: String + kontostand: Double + bonuspunkte: Int
+ anzeigen() + kontostandAktualisieren() + bonuspunkteAktualisieren()

class Produkt
+ artikelNr: String + name: String + preis: Double + lagerbestand: Int
+ anzeigen() + rabattgeben(prozent: Int)

class IMac
+ screenSize: String
+ upgradeScreenSize()

class MacBookAir
+ prozessor: String
+ upgradeProzessor()

class iPhone
+ speicher: Int
+ upgradeSpeicher()

struct Geschenk
+ name: String + wert: Double + beschreibung: String
+ anzeigen()

struct Zubehoer
+ name: String + preis: Double + rabattPreis: Double
+ anzeigen() + berechneRabett()

protocoll Kaufbar	protocoll Rabattierbar
+ preis: Double {get} + lagerbestand: Int {get}	+ rabatt: Double {get set} + originalpreis: Double {get}
+ kaufen(int) -> Bool	+ berechnePreis() -> Double

<u>Enum ShopStatus</u>
case shopping case bestellbestätigung case bestellabschluss

<u>Hauptmenü</u>	<u>Abbruchbedingungen</u>
1. Kunden auswählen 2. Kundeninfos anzeigen 3. Produkte aussuchen 4. Warenkorb anzeigen 5. Bestellung abschließen 6. Bestellung abbrechen	1. Erfolgreicher Einkauf bzw. Checkout für Programmende 2. Abbruch der Bestellung für Programmende

<u>Extension 1</u>	<u>Extension 2</u>
Textformat für Titel mit .capitalized Großschreibung erster Buchstabe mit .uppercased()	Gesamtpreis aller Elemente des Warenkorbbes berechnen