

Aufgabe (Strukturen, enum, Datenfeld, benutzergesteuerte Schleife):

In einem Online-Shop sollen die Kunden erfasst werden mit:

- Vorname (beliebig lange Zeichenkette)
- Nachname (beliebig lange Zeichenkette)
- Kundennummer (Ganzzahl)
- Kaufsumme (Geldbetrag als Summe aller bisher erfolgten Bestellungen)
- Kundenstatus (Auswahl aus der nachstehend definierten Liste)

Für den Kundenstatus gibt es 4 Zustände und Bezeichnungen:

- Neu: „Neuer Kunde“
- Premium: „Premium-Kunde“
- Gold: „Gold-Kunde“
- Platin: „Platin-Kunde“

Die „Spielregeln“ sind recht einfach:

Wurde ein Kunde erfasst, hat aber noch nie etwas bestellt, dann ist er ein „Neuer Kunde“, die Kaufsumme ist (logischerweise) noch 0 Euro.

Bei jeder Bestellung wird die Kaufsumme um den Bestellwert erhöht.

Bei jeder Bestellung ab 5,00 Euro wird der Kundenstatus um 1 erhöht.

Aufgabe 1:

Deklarieren Sie die benötigten/geeigneten eigenen Datentypen. (Struktur, enum?)

Legen Sie ein Datenfeld für 10.000 Kunden an und initialisieren Sie die ersten 3 Kunden mit beliebigen Werten. Zeigen Sie die 3 Kunden mit allen Daten an (siehe Muster).

Aufgabe 2:

Der Benutzer kann nun anhand der Kundennummer einen Kunden auswählen, der etwas bestellen will. Anschließend wird der Wert der Bestellung eingegeben und die Kaufsumme und der Kundenstatus für den ausgewählten Kunden aktualisiert.

Kundenliste:

KdNr	Name	Status	Summe
123	Max Mustermann	Neuer Kunde	0,00
234	Peter Pan	Gold-Kunde	43,25
345	Darth Vader	Platin-Kunde	123,45

Geben Sie die Kundennummer fuer eine neue Bestellung ein: **234**
Geben Sie die Kaufsumme fuer diese Bestellung ein: **10**

Kundenliste:

KdNr	Name	Status	Summe
123	Max Mustermann	Neuer Kunde	0,00
234	Peter Pan	Platin-Kunde	53,25
345	Darth Vader	Platin-Kunde	123,45

Geben Sie die Kundennummer fuer eine neue Bestellung ein: **0**

Die Eingabe der Kundennummer 0 beendet das Programm.

Aufgabe 3 (Einführung von Klassen und Objekten):

Für den Kunden werden Sie (hoffentlich...) eine Struktur angelegt haben.

Ersetzen Sie nun den Datentyp **struct** durch den „Datentyp“ **class**. (Wir sprechen hier dann allerdings nicht mehr von einem Datentyp, sondern von einer „Klassenvereinbarung“.)

Läuft das Programm noch?

Denken Sie an die „**private**“- und „**public**“-Zugriffsbereiche einer Klasse!

Alles „public“ machen, ist jedoch nicht Sinn der Sache...

Lassen Sie die **Eigenschaften** eines Kunden „private“ und formulieren Sie **Funktionen** (wir nennen Sie bei Klassen „**Methoden**“), mit denen Sie auf die Eigenschaften des Kunden zugreifen können: zum Beispiel zum Anzeigen (als „get“-Methode) oder auch zum Verändern der Werte (bei einem neuen Kauf, als „set“-Methode).

Schreiben Sie Ihr Hauptprogramm so um, dass Sie diese Methoden bequem verwenden können.

Wenn es nicht auf Anhieb gleich passt: macht nichts – wir gehen das nachher im Theorie-Unterricht ganz detailliert und Schritt für Schritt noch einmal durch!