

# LU13.A02 - Hofladen erweitern

## Ausgangslage

Der Bauer der den Hofladen betreut ist sehr glücklich über die neue Bestandsverwaltung für den Hofladen aus der [LU12.A02](#). Für seinen Steuerverwalter muss der Bauer jederzeit den Gesamtwert pro Produkt des Bestandes im Hofladen und zusätzlich noch den Gesamtwert aller Produkte im Hofladen ausgeben können.

Sie können diese Aufgabe lösen, indem Sie entweder:



- Ihren Hofladen aus LU12 erweitern und ihre Lösung hier auf Moodle abgeben.
- Die Github Classroom Aufgabe annehmen und mit der Lösung von LU12 weitermachen und ihre Lösung comitten und pushen.

## Programmaufgabe

Ihr Programm kann bereits die Artikel im Hofladen verwalten.

### Er funktioniert bisher so:

- Der Benutzer gibt zuerst den Namens eines Artikels ein.
- Falls der Name Exit ist, wird das Programm beendet.
- Falls der Artikel noch nicht existiert, muss der Benutzer den Preis eingeben.
- Andernfalls wird der aktuelle Bestand ausgegeben.
- Zuletzt gibt der Benutzer die Veränderung des Bestands ein.

### Neu kommt dazu:

- Der Benutzer gibt zuerst den Namens eines Artikels ein.
- Falls der Name Exit ist, wird das Programm beendet.
- Falls der Name Inventory ist, druckt das Programm für jeden Artikel den Produktwert (Preis \* Bestand) und den Gesamtwert über alle Produkte aus
- Falls der Artikel noch nicht existiert, muss der Benutzer den Preis eingeben.
- Andernfalls wird der aktuelle Bestand ausgegeben.
- Zuletzt gibt der Benutzer die Veränderung des Bestands ein.

## Beispiel

```
Artikelname > Milch  
Preis      > 5.75
```

```
Menge          > 5

Artikelname > Wurst
Preis       > 21.10
Menge      > 12

Artikelname > Milch
Bestand    : 5
Menge     > -2

Artikelname > Inventory

Milch      : 17.25
Wurst     : 253.20

Total     : 270.45

Artikelname > Exit
```

## Programmablauf neu

1. Erstelle eine leere Liste für die Artikel
2. Eingabe des Artikelnamens
3. Solange Artikelname nicht Exit ist
  1. Falls die Eingabe Inventory
    1. Für jedemn Artikel in der Liste
      1. Gib den Namen und den Produktwert (Menge \* Preis) aus
      2. Addiere den Wert zum Gesamtwert
    2. Gib den Gesamtwert aus
    3. Eingabe des Artikelnamens
    4. Begin die Schleife von vorne
  2. Sonst
    1. Suche den Artikel in der Liste und merke dir diesen Artikel
    2. Falls kein Artikel gefunden wurde
      1. Erstelle ein neues Artikel-Objekt mit dem Artikelnamen, Preis=0.00 und Bestand=0
      2. Speichere das Objekt in der Liste
      3. Eingabe des Preises
    3. Sonst
      1. Ausgabe des aktuellen Bestands
    4. Eingabe der Menge
    5. Addiere die Menge zum Bestand
    6. Eingabe des Artikelnamens
4. Gib die Artikelliste als Returnwert zurück

# Umsetzung

## 1. Anpassung Produkt

Zu jedem Produkt werden diese Attribute gespeichert:

- Name des Artikels (Text)
- Preis des Artikels (Dezimalzahl)
- Bestand (Ganzzahl)

Zusätzlich zu diesen drei Angaben benötigen wir neu auch noch den Produktwert. Wie wir in der [Theorie zu LU13](#) gelernt haben, können wir einen Getter, resp. ein property für einen Wert auch ohne eigenes Attribut machen. Ergänzen Sie also ihr Produkt um ein property für `product_value`.

## 2. Funktion definieren

Bisher haben Sie Ihren Code in mehrere Funktionen aufgeteilt. Auch hier würde es sich anbieten, eine Funktion zu schreiben, die dafür verantwortlich ist, das Inventar auszudrucken und den Gesamtwert zu berechnen und auszugeben.

Definieren sie also eine Funktion, die über eine ihr übergebene Liste iteriert und von jedem Produkt den name und `product_value` ausgibt. Ebenso soll die Funktion eine Summe über alle `product_value` erstellen und diese am Schluss ausgeben.

1. Erstelle eine leere Liste für die Artikel
  2. Eingabe des Artikelnamens
  3. Solange Artikelname nicht `Exit` ist
    1. Falls die Eingabe `Inventory`
      1. Für jedemn Artikel in der Liste
        1. Gib den Namen und den Produktwert ( $\text{Menge} * \text{Preis}$ ) aus
        2. Addiere den Wert zum Gesamtwert
      2. Gib den Gesamtwert aus
    3. Eingabe des Artikelnamens
    4. Begin die Schleife von vorne
  2. Sonst
    1. Suche den Artikel in der Liste
    2. Falls kein Artikel gefunden wurde
      1. Erstelle ein neues Artikel-Objekt mit dem Artikelnamen, `Preis=0.00` und `Bestand=0`
      2. Speichere das Objekt in der Liste
      3. Eingabe des Preises
    3. Sonst
      1. Ausgabe des aktuellen Bestands
    4. Eingabe der Menge
    5. Addiere die Menge zum Bestand
    6. Eingabe des Artikelnamens
4. Gib die Artikelliste als Returnwert zurück

### 3. Programmfluss anpassen

Damit ihr Programm jetzt auf Inventory als Artikelname reagieren kann, müssen Sie den Kontrollfluss anpassen.

1. Erstelle eine leere Liste für die Artikel
2. Eingabe des Artikelnamens
3. Solange Artikelname nicht Exit ist
  1. Falls die Eingabe Inventory
    1. Für jedemn Artikel in der Liste
      1. Gib den Namen und den Produktwert (Menge \* Preis) aus
      2. Addiere den Wert zum Gesamtwert
    2. Gib den Gesamtwert aus
    3. Eingabe des Artikelnamens
    4. Begin die Schleife von vorne
  2. Sonst
    1. Suche den Artikel in der Liste
    2. Falls kein Artikel gefunden wurde
      1. Erstelle ein neues Artikel-Objekt mit dem Artikelnamen, Preis=0.00 und Bestand=0
      2. Speichere das Objekt in der Liste
      3. Eingabe des Preises
    3. Sonst
      1. Ausgabe des aktuellen Bestands
    4. Eingabe der Menge
    5. Addiere die Menge zum Bestand
    6. Eingabe des Artikelnamens
4. Gib die Artikelliste als Returnwert zurück

From:  
<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:  
<https://wiki.bzz.ch/modul/archiv/m319python/learningunits/lu13/aufgaben/hofladen>

Last update: **2024/03/28 14:07**

