LU13.L02 - Hofladen erweitert

Umsetzung

1. Anpassung Produkt

```
    @property
    def article_value(self):
        return self._price * self._stock
```

2. Funktion definieren

```
def handle_inventory(list):
    """
    prints the inventory and the total value
    :param list: the article list
    :return: None
    """
    sum = 0
    for article in list:
        sum = sum + article.article_value
        print(article.name + ' : ' +
str(article.article_value))
    print('\nGesamt : ' + str(sum))
```

3. Programmfluss anpassen

```
if article_name == 'Inventory':
  handle_inventory(article_list)
  article_name = input('Artikelname > ')
  continue
else:
```

Gesamtlösung

article.py

```
from dataclasses import dataclass


@dataclass
class Article:
    """
    an article in the farmshop
    """
    name: str
    price: float
    stock: int

    @property
    def name(self):
        return self._name

    @name.setter
    def name(self, value):
        self._name = value

    @property
    def price(self):
        return self._price

    @price.setter
    def price(self, value):
        self._price = value

    @property
    def stock(self):
        return self._stock

    @stock.setter
    def stock(self, value):
        self._stock = value

    @property
    def article_value(self):
        return self._price * self._stock

if __name__ == '__main__':
    pass
```

article.py

```
from article import Article


def main():
    article_list = []
    article_name = input('Artikelname > ')
    while article_name != 'Exit':
        if article_name == 'Inventory':
            handle_inventory(article_list)
            article_name = input('Artikelname > ')
            continue
        else:
            article = find_article(article_list,
article_name)

            if article is None:
                article = Article(article_name, 0.0, 0)
                article_list.append(article)
                article.price = input_float('Preis
> ')
            else:
                print('Bestand     : ' +
str(article.stock))

            amount = input_float('Menge       > ')
            article.stock = (article.stock + amount)
            article_name = input('Artikelname > ')

    return article_list


def handle_inventory(list):
    """
    prints the inventory and the total value
    :param list: the article list
    :return: None
    """
    sum = 0
    for article in list:
        sum = sum + article.article_value
        print(article.name + ' : ' +
str(article.article_value))
    print('Gesamt : ' + str(sum))


def input_int(prompt):
    """
    reads an integer input from the user
    :param text: the prompt to be shown
```

```
        :return: the integer number
        """
        number = None
        while number is None:
            try:
                number = int(input(prompt))
            except ValueError:
                print("Please, enter a whole number!")
                continue
        return number


    def input_float(prompt):
        """
        reads a decimal number input from the user
        :param text: the prompt to be shown
        :return: the decimal number
        """
        number = None
        while number is None:
            try:
                number = float(input(prompt))
            except ValueError:
                print("Please, enter a real number!")
                continue
        return number


    def find_article(list, name):
        """
        finds an article in the article list
        :param list: the article list
        :param name: the article name to be found
        :return: article or None=not found
        """
        for article in list:
            if article.name == name:
                return article
        return None


    if __name__ == '__main__':
        main()
```