

2. Übungsblatt, Thema: ClassPractice

1 Define class Product

Eine Instance von dieser Klasse bildet eine Anzahl von Produkten ab. Jedes Produkt einer Instance hat den gleichen Namen und Einzelpreis.

Die Klasse soll folgende Attribute haben.

- SinglePrice (Preis eines einzelnen Produktes) vom Type Double
- Name (Name des Produktes) vom Type String
- ID (Eindeutige einzigartige Zahl zu einem Produkt) vom Type Int

Diese Attribute sollen ausserhalb der Klasse abgerufen werden können aber nicht verändert werden können. Diese Attribute können also nur während der Erstellung des Objekts gesetzt werden.

Attribute, welche auch ausserhalb der Klasse noch verändert werden sollen.

- Number (Anzahl des Produktes)

Es soll ein Konstruktor geschrieben werden. Welche 3 Argumente annimmt.

- 1. Parameter ist der Name des Produktes.
- 2. Parameter ist der Preis des Produktes.
- 3. Parameter ist die Anzahl des Produktes.

Test Program:

```
Product dummyProduct = new Product("Dummy", 0.0, 0);
Product dummyProduct1 = new Product("Dummy", 0.0, 0);
Product product = new Product("TV", 8000.0, 2);
Console.WriteLine($"product.ID = {product.ID}");
Console.WriteLine($"product.Name = {product.Name}");
Console.WriteLine($"product.SinglePrice = {product.SinglePrice}");
```

Ausgabe sollte sein:

```
product.ID = 2
product.Name = TV
product.SinglePrice = 8000
```

2 Implementierung von Property WholePrice

Getter, welche die gesamte Summe aller Einzelpreise eines Produktes zurück gibt.

```
Product product = new Product("Chips", 2.0, 5);
Console.WriteLine($"product.WholePrice = {product.WholePrice}");
```

Ausgabe sollte sein:

```
product.WholePrice = 10
```

3 Implementierung von Methods GetProductInfo

Gibt einen string zurück, welche ein Produkt als Text darstellt.

```
Product product = new Product("Gurke", 0.5, 10);
Console.WriteLine($"product.GetProductInfo() = {product.GetProductInfo()}");
```

Ausgabe sollte sein:

```
product.GetProductInfo() = Gurke, Single price: 0.5$, x10 = 5$
```

4 Implementiere eine Methode namens GetWholePrice

Dieser Methode nimmt ein Array von Produkts entgegen und gibt dann die Summe aller Preise von Produkten zurück.

```
Product[] list = new Product[] { new Product("Auto", 20000.0, 3), new Product("Chips", 2.0, 20) };
Console.WriteLine($"Product.GetWholePrice(list) = {Product.GetWholePrice(list)}");
```

Ausgabe sollte sein:

```
Product.GetWholePrice(list) = 60040
```

5 Implementiere eine Methode namens ProductTable

Diese Methode soll einen string zurück geben. Diese Methode soll ein Arrays aus Produkts annehmen. Der String stellt eine Tabelle aus den Produkten dar. Dabei ist jede Zeile die Ausgabe von der Methode **GetProductInfo** eines Produktes.

Die letzt Zeile soll aber noch den gesamten Preis alle Produkte zurück geben mit der zurückgegeben Wert durch die Methode **GetWholePrice**. Die Ausgabe der letzten Zeile für den Gesamtpreis mit den Wert 60040 soll zum Beispiel folgendermaßen aussehen.

```
Total Price: 60040
```

```
var list = new Product[] { new Product("Auto", 20000.0, 3), new Product("Chips", 2.0, 20) };
Console.WriteLine(Product.ProductTable(list));
```

Ausgabe sollte sein:

Auto, Single price: 20000\$, x3 = 60000\$

Chips, Single price: 2\$, x20 = 40\$

Total Price: 60040