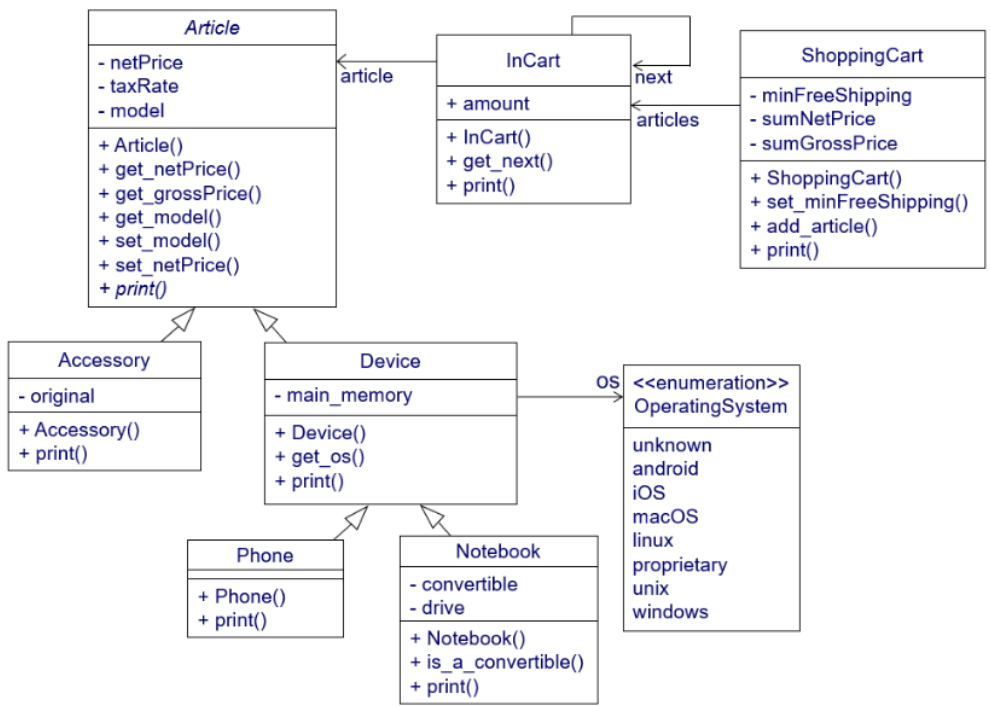
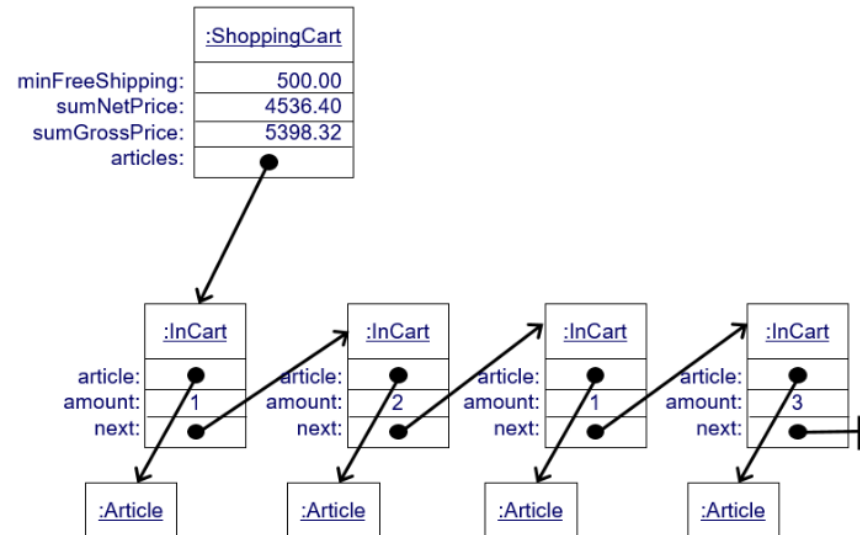


## Überblick/Overview

In dieser Aufgabe soll die Einfachvererbung eingeübt werden mit statischem und dynamischem Binden. Dazu soll ein einfacher und erweiterbarer elektronischer Warenkorb programmiert werden. Als Beispiel-Artikel sollen Smartphones, Notebooks und Zubehör bestellt werden können (im UML-Klassendiagramm unten sind Datentypen, Parameter und Destruktoren aus Übersichtlichkeitsgründen weggelassen.)/ In this task single inheritance with static and dynamic binding shall be trained. For this a simple and extendable electronic shopping cart shall be programmed. As example articles smart phones, notebooks and accessories shall be ordered (due to clarity in the UML class diagram below data types, parameters and destructors are omitted.)





### Teilaufgabe 1/Subtask 1

Definieren Sie eine abstrakte Basisklasse für einen angebotenen Artikel mit Namen **Article**. Diese Klasse soll folgende Member haben:/

Define an abstract base class for an offered article with name **Article**. This class shall have following members:

- ein privates Zeichenketten-Attribut mit Namen **model** für eine Artikel-Modellbezeichnung./  
a private string attribute with name **model** storing the article model name.
- zwei private reellwertige Attribute mit Namen **netPrice** für den Nettopreis und **taxRate** für den Prozentsatz der Mehrwertsteuer für den Artikel./  
two private real valued attributes with names **netprice** for the net price and **taxRate** storing the tax value in percent of the article.
- ein öffentlicher Konstruktor mit drei Parametern zur Initialisierung der Attribute eines Objekts; der Mehrwert-/Umsatzsteuersatz soll als Defaultparameter 19 % betragen./  
a public constructor with three parameters to initialise the attributes of an object; the tax rate shall be a default parameter with 19 %.
- ein virtueller öffentlicher Destruktor, der eine Ausgabe "**~Article()**" bei dessen Aufruf und die Artikel-Modellbezeichnung ausgibt (siehe Beispiele unten)./  
a virtual public destructor producing an output "**~Article()**" at its call and the article model name (see examples below).

- eine öffentliche Methode mit Namen **set\_model**, die einen Zeichenketten-Parameter hat und die Modellbezeichnung zuweist./  
a public method with name **set\_model** assigning a string parameter to the model name attribute.
- eine öffentliche Methode mit Namen **set\_netPrice**, die einen reellwertigen Parameter hat und den Nettopreis zuweist./  
a public method with name **set\_netPrice** assigning the double parameter to the net price attribute.
- eine öffentliche Methode mit Namen **get\_model** ohne Parameter, die die Modellbezeichnung als Funktionswert zurück liefert./  
a public method with name **get\_model** without parameters returning the model name as function value.
- eine öffentliche Methode mit Namen **get\_netPrice** ohne Parameter, die den Nettopreis als Funktionswert zurück liefert./  
a public method with name **get\_netPrice** without parameters returning the net price as function value.
- eine öffentliche Methode mit Namen **get\_grossPrice** ohne Parameter, die den Bruttopreis (Bruttopreis = Nettopreis + Mehrwert-/Umsatzsteuer) als Funktionswert zurück liefert./  
a public method with name **get\_grossPrice** without parameters returning the gross price (gross price = net price + tax) as function value.
- eine virtuelle rein abstrakte öffentliche Methode mit Namen **print** ohne Parameter oder Rückgabewert./  
a pure virtual public method with name **print** without parameter or return value.

## Teilaufgabe 2/Subtask 2

Definieren Sie eine Klasse für Zubehörteile mit Namen **Accessory** abgeleitet von der abstrakten Klasse **Article**. Diese Klasse soll folgende Member haben:/

Define a class with name **Accessory** derived from abstract class **Article**. This class shall have following members:

- ein privates Boolesches Attribut mit Namen **original** für ein Original-Zubehörteil./  
a private Boolean attribute with name **original** for an original accessory.
- ein öffentlicher Konstruktor mit drei Parametern für die Artikelbezeichnung, den Nettopreis und ob dies ein Original-Zubehör ist mit **true** als Defaultparameter. Der Mehrwertsteuersatz soll automatisch über den Defaultparameter vom Klassenkonstruktor **Article** initialisiert werden./  
a public constructor with three parameters for the article name, the net price and whether it is an original accessory with **true** as default parameter. The tax rate shall be automatically initialised by the default parameter of class constructor **Article**.
- ein virtueller öffentlicher Destruktor, der eine Ausgabe "**~Accessory()**" bei dessen Aufruf ausgibt (siehe Beispiele unten)./  
a virtual public destructor producing an output "**~Accessory()**" at its call (see examples at the bottom).
- eine virtuelle öffentliche Methode mit Namen **print** ohne Parameter, die die Artikelbezeichnung ausgibt sowie dahinter "**(original accessory)**" bei einem Originalartikel (siehe Beispiele unten)./  
a virtual public method with name **print** without parameters writing the article name and "**(original accessory)**", if it is an original one (see examples at the bottom).

## Teilaufgabe 3/Subtask 3

- Definieren Sie eine C++11-Aufzählungsklasse mit Namen **OperatingSystem** und den Aufzählungswerten **unknown**, **android**, **iOS**, **macOS**, **linux**, **proprietary**, **unix** und **windows**./  
Define a C++11 enumeration class with name **OperatingSystem** and enumeration values **unknown**, **android**, **iOS**, **macOS**, **linux**, **proprietary**, **unix** and **windows**.
- Definieren Sie einen überladenen Ausgabeoperator << zur textuellen Ausgabe eines Betriebssystem-Namens ("**unknown OS**", "**Android OS**", "**iOS**", "**MacOS**", "**Linux OS**", "**proprietary OS**", "**Unix OS**", "**MS Windows OS**") auf einen Ausgabedatenstrom (siehe Beispiel unten)./  
Define an overloaded output operator << for a textual output of an operating system name ("**unknown OS**", "**Android OS**", "**iOS**", "**MacOS**", "**Linux OS**", "**proprietary OS**", "**Unix OS**", "**MS Windows OS**") onto an output stream (see example at the bottom).

#### Teilaufgabe 4/Subtask 4

Definieren Sie eine Klasse mit Namen **Device** abgeleitet von der abstrakten Klasse **Article**. Diese Klasse soll folgende Member haben:/  
Define a class with name **Device** derived from abstract class **Article**. This class shall have following members:

Define a class with name **Device** derived from abstract class **Article**. This class shall have following members:

- ein privates ganzzahliges Attribut mit Namen **main\_memory** für die Größe des eingebauten Hauptspeichers./  
a private integer attribute with name **main\_memory** storing the size of the main memory.
- ein privates Attribut mit Namen **os** vom obigen Aufzählungstyp für das Betriebssystem des Geräts./  
a private attribute with name **os** for the operating system of the device.
- ein öffentlicher Konstruktor mit vier Parametern für die Artikelbezeichnung, den Nettopreis, die Hauptspeichergröße und das Betriebssystem mit Defaultparameter unbekannt (**unknown**); der Mehrwertsteuersatz soll automatisch über den Defaultparameter vom Klassenkonstruktor **Article** initialisiert werden./  
a public constructor with four parameters for the article name, the net price, the main memory size and the operating system with default parameter **unknown**; the tax rate shall be automatically initialised by the default parameter of class constructor **Article**.
- ein virtueller öffentlicher Destruktor, der eine Ausgabe "**~Device()**" bei dessen Aufruf ausgibt (siehe Beispiele unten)./  
a virtual public destructor producing an output "**~Device()**" at its call (see examples at the bottom).
- eine öffentliche Methode mit Namen **get\_os** ohne Parameter, die als Funktionswert das Betriebssystem zurück liefert./  
a public method with name **get\_os** without parameter returning the operating system.
- eine virtuelle öffentliche Methode mit Namen **print** ohne Parameter oder Rückgabewert, die die Artikelbezeichnung, dahinter "**RAM**", die Hauptspeichergröße und "**GB**" sowie das Betriebssystem ausgibt (siehe Beispiele unten)./  
a virtual public method with name **print** without parameter or return value outputting the article name, followed by "**RAM**", the main memory size, "**GB**" as well as the operating system (see examples at the bottom).

#### Teilaufgabe 5/Subtask 5

Definieren Sie eine Klasse mit Namen **Notebook** abgeleitet von der Klasse **Device**. Diese Klasse soll folgende Member haben:/  
Define a class with name **Notebook** derived from the class **Device**. This class shall have following members:

Define a class with name **Notebook** derived from class **Device**. This class shall have following members:

- ein privates Zeichenketten-Attribut mit Namen **drive** für das Laufwerk./  
a private string attribute with name **drive**.
- ein privates Boolesches Attribut mit Namen **convertible** für ein Convertible./  
a private Boolean attribute with name **convertible** for a convertible one.
- ein öffentlicher Konstruktor mit sechs Parametern für die Artikelbezeichnung, den Nettopreis, die Hauptspeichergröße, das Laufwerk, das Betriebssystem mit Defaultparameter **OperatingSystem::linux** sowie ob das Notebook ein Convertible ist mit Defaultparameter **false**; der Mehrwertsteuersatz soll automatisch über den Konstruktor der Oberklasse initialisiert werden./  
a public constructor with six parameters for the article name, the net price, the main memory size, the drive, the operating system with default parameter **OperatingSystem::linux** as well as whether it is a convertible with default parameter **false**; the tax rate shall be initialised automatically by the constructor of the upper class.
- ein virtueller öffentlicher Destruktor, der eine Ausgabe "**~Notebook()**" bei dessen Aufruf ausgibt (siehe Beispiele unten)./  
a virtual public destructor producing an output "**~Notebook()**" at its call (see examples at the bottom).
- eine öffentliche Methode mit Namen **is\_a\_convertible** ohne Parameter, die als Booleschen Funktionswert zurück liefert, ob das Notebook ein Convertible ist./  
a public method with name **is\_a\_convertible** without parameters returning a Boolean function value whether the notebook is a convertible.
- eine virtuelle öffentliche Methode mit Namen **print** ohne Parameter, die die **print**-Methode der direkten Oberklasse aufruft, im Falle eines Convertible die Zeichenkette "**(convertible)**" und danach das Laufwerk ausgibt./  
a virtual public method with name **print** without parameters, calling method **print** of the direct upper class, in case of a convertible outputs string "**(convertible)**" and afterwards the drive.

### Teilaufgabe 6/Subtask 6

Definieren Sie eine Klasse mit Namen **Phone** abgeleitet von der Klasse **Device**. Diese Klasse soll folgende Member haben: /

Define a class with name **Phone** derived from class **Device**. This class shall have following members:

- ein öffentlicher Konstruktor mit vier Parametern für die Artikelbezeichnung, den Nettopreis, die Hauptspeichergröße mit 4GB als Defaultparameter und das Betriebssystem Android OS als Defaultparameter; der Mehrwertsteuersatz soll automatisch über den Konstruktor der Oberklasse initialisiert werden./  
a public constructor with four parameters for the article name, the net price, the main memory size with 4GB as Defaultparameter and the operating system Android OS as default parameter; the tax rate shall be initialised automatically by the constructor of the upper class.
- ein virtueller öffentlicher Destruktor, der eine Ausgabe "**~Phone()**" bei dessen Aufruf ausgibt (siehe Beispiele unten)./  
a virtual public destructor producing an output "**~Phone()**" at its call (see examples at the bottom).
- eine virtuelle öffentliche Methode mit Namen **print** ohne Parameter, die die **print**-Methode der direkten Oberklasse aufruft und dahinter "**phone**" ausgibt./  
a virtual public method with name **print** without parameters calling method **print** of the direct upper class and afterwards outputting "**phone**".

## Teilaufgabe 7/Subtask 7

Definieren Sie eine Klasse für einen Artikel in einem Warenkorb mit Namen **InCart**. Diese Klasse soll folgende Member haben: /

Define a class for an article in the shopping cart with name **InCart**. This class shall have following members:

- ein privates Zeiger-Attribut mit Namen **article** auf ein Objekt der Klasse **Article**. /  
a private pointer attribute with name **article** to an object of class **Article**.
- ein privates ganzzahliges Attribut mit Namen **amount** für die Anzahl dieses Artikels im Warenkorb. /  
a private integer attribute with name **amount** how many of this article are placed in the shopping cart.
- ein privates Attribut mit Namen **next** vom Typ Zeiger auf **InCart** zum Aufbau einer Liste. /  
a private attribute with name **next** of type pointer to **InCart** for building up a list.
- ein öffentlicher Konstruktor mit drei Parametern zur Initialisierung der drei Attribute eines Objekts; die Anzahl soll als Defaultparameter 0 sein, der Zeiger auf das nächste Element im Warenkorb ebenfalls als Defaultparameter ein Nullzeiger. /  
a public constructor with three parameters to initialise the three attributes of an object; the amount shall be 0 as default parameter, the pointer to the next article in a shopping cart a null pointer also as default parameter.
- ein öffentlicher Destruktor, der "**~Incart()**" und die Anzahl ausgibt sowie den Destruktor für den Artikel aufruft (siehe Beispiele unten). /  
a virtual public destructor outputting "**~Incart()**" and the amount as well as calling the destructor for the article (see examples at the bottom).
- eine öffentliche Methode mit Namen **get\_next** ohne Parameter, die einen Zeiger auf das nächste Objekt im Warenkorb als Funktionswert zurück liefert. /  
a public method with name **get\_next** without parameters returning a pointer to the next article in shopping cart as function value.
- eine virtuelle öffentliche Methode mit Namen **print** ohne Parameter, die die Anzahl ausgibt, dann dem Artikel im Warenkorb die Nachricht **print** sendet, danach in einer neuen Zeile genau wie im Beispiel unten formatiert den Einzel-Bruttopreis und den Gesamt-Bruttopreis für die gewählte Anzahl dieses Artikels auf den Standardausgabestrom schreibt (siehe Beispiele unten). /  
a virtual public method with name **print** without parameters, outputting the amount, then sending message **print** to the article in the shopping cart, then in a new line outputting formatted as in the examples at the bottom the single gross price and the total gross price for the chosen amount of articles onto an output stream (see example at the bottom).

## Teilaufgabe 8/Subtask 8

Definieren Sie eine Klasse für einen elektronischen Warenkorb mit Namen **ShoppingCart**. Diese Klasse soll folgende Member haben: /

Define a class for a shopping cart with name **ShoppingCart**. This class shall have following members:

- ein privates Zeiger-Attribut mit Namen **articles** auf den Kopf einer Liste von Artikeln im Warenkorb der Klasse **InCart**. /  
a private pointer attribute with name **articles** to the head of a list of articles in the shopping cart of class **InCart**.



- drei private reellwertige Attribute mit Namen **minFreeShipping**, ab welchem Einkaufswert ein kostenloser Versand angeboten wird, **sumNetPrice** für die Gesamt-Nettopreis-Summe aller Artikel im Warenkorb und **sumGrossPrice** für die Gesamt-Bruttopreis-Summe./  
three private real valued attributes with names **minFreeShipping** for which purchase value free shipping is offered, **sumNetPrice** for the total net price sum of all articles in the shopping cart and **sumGrossPrice** for the total gross price sum.
- ein öffentlicher Standardkonstruktor, der einen leeren Warenkorb initialisiert, beide Summenwerte auf 0 EUR sowie den Einkaufswert für einen freien Versand auf 1000 EUR./  
a public standard constructor initialising an empty shopping cart with both sum values 0 EUR and the purchase value for free shipping with 1000 EUR.
- ein öffentlicher Destruktor, der "**~ShoppingCart()**" ausgibt und danach jeden Artikel im Warenkorb beginnend mit einer Ausgabe "**delete article:** " jeweils komplett löscht, so dass kein Speicherleck existiert./  
a public destructor outputting "**~ShoppingCart()**" and afterwards deleting each article in the shopping cart starting with an output "**delete article:** " such that no memory leak will exist.
- eine öffentliche Methode mit Namen **set\_minFreeShipping** mit einem Parameter, der dessen Wert dem gleichnamigen Attribut zuweist./  
a public method with name **set\_minFreeShipping** with one parameter assigning its value to the same named attribute.
- eine öffentliche Methode mit Namen **add\_article**, die als Parameter eine Anzahl und einen Zeiger auf einen Artikel hat, ein neues Objekt **InCart** auf dem Heap dem Warenkorb hinzufügt und auch die Werte der beiden Summenvariable der Klasse entsprechend aktualisiert./  
a public method with name **add\_article** with an amount and a pointer to an article as parameters, adding a new object **InCart** on heap to the shopping cart and actualising the two sum variables of the class appropriately.
- eine virtuelle öffentliche Methode mit Namen **print** ohne Parameter, die den Warenkorb wie in den Beispielen unten formatiert auf den Standardausgabestrom ausgibt. Dabei sollen insbesondere am Ende des Warenkorbs die Summe netto, die gesamte Mehrwert-/Umsatzsteuer (Differenz Brutto- minus Netto-Gesamtsumme) sowie die Summe brutto ausgegeben werden. Beachten Sie bei der Ausgabe die (eventuellen) Versandkosten von 5.90 EUR (siehe Beispiel unten)!/  
a virtual public method with name **print** without parameters writing a shopping onto the standard output stream cart formatted as given in the examples at the bottom. Especially the net price sum, the tax (difference gross minus net total sum) and the gross price sum shall be outputted at the end of the shopping cart. Take care to also regard the (may be) shipping cost output of 5.90 EUR (see example at the bottom)!

### Teilaufgabe 9/Subtask 9

Schreiben Sie eine Funktion **main**, in der Sie zum Testen einige Artikel als neue Objekte auf dem Heap erzeugen und einem elektronischen Warenkorb hinzufügen:

To test write a **main** function placing some articles as new objects on the heap into an electronic shopping cart:

- Definieren Sie ein Warenkorb-Objekt vom Typ **ShoppingCart** über den Standardkonstruktor./  
define a shopping cart object of type **ShoppingCart** using the standard constructor.
- Senden Sie diesem Objekt eine Nachricht, dass die Waren ab 500 EUR versandkostenfrei geliefert werden./  
Send a message to this object that the articles get free shipping starting from 500 EUR.
- Fügen Sie diesem Warenkorb als neuem Zubehör-Artikel auf dem Heap mit Anzahl drei Powerbanks "**YOLOX 10k Wireless Powerbank Qi**" zum Einzel-Nettopreis 31.08 EUR als kein Originalzubehör hinzu./

- Fügen Sie diesem Warenkorb als neuem Zubehör-Artikel auf dem Heap mit Anzahl drei Powerbanks "**YOOLOX 10k Wireless Powerbank Qi**" zum Einzel-Nettopreis 31.08 EUR als kein Originalzubehör hinzu./  
Add to this shopping cart a new accessory article on heap with an amount of three powerbanks "**YOOLOX 10k Wireless Powerbank Qi**" with single net price 31.08 EUR being no original accessory.
- Drucken Sie den kompletten Warenkorb aus (Versandkosten fallen an)./  
Print the complete shopping cart (there are shipping costs).
- Fügen Sie dem Warenkorb ein neues Smartphone "**Samsung Galaxy S10+ SM-G975F/DS Dual SIM**" mit Nettopreis 661.67 EUR, 8GB Hauptspeicher und Android OS als Betriebssystem hinzu./  
Add to this shopping cart a new smart phone "**Samsung Galaxy S10+ SM-G975F/DS Dual SIM**" with net price 661.67 EUR, 8GB main memory size and Android OS operating system.
- Fügen Sie dem Warenkorb zwei neue Smartphones "**Apple iPhone 11 Pro 256GB**" mit Nettopreis 1097.47 EUR, 4GB Hauptspeicher und iOS als Betriebssystem hinzu./  
Add to this shopping cart two new smart phones "**Apple iPhone 11 Pro 256GB**" with net price 1097.47 EUR, 4GB main memory size and iOS operating system.
- Fügen Sie dem Warenkorb einen neues Notebook "**ASUS ROG Strix Scar III G731**" mit Nettopreis 1586.55 EUR, 16GB Hauptspeicher, als Laufwerk "**512GB SSD + 1TB SSHD**" und MS Windows als Betriebssystem hinzu./  
Add to this shopping cart a new notebook "**ASUS ROG Strix Scar III G731**" with net price 1586.55 EUR, 16GB main memory size, "**512GB SSD + 1TB SSHD**" as drive and MS Windows operating system.
- Drucken Sie den kompletten Warenkorb erneut aus (dieser ist nun frei von Versandkosten)./  
Print again the complete shopping cart (now there are no shipping costs).

Mit dem Ende der Funktion **main** wird das Warenkorb-Objekt automatisch wieder gelöscht; Ihr Programm soll dazu die Ausgaben der Destruktoraufrufe wie im Beispiel unten zeigen./  
With the end offunction **main** automatically the complete shopping cart object gets deleted; your program shall show the outputs of the destructor calls like in the example given below.

### Beispiel Programmlauf/Example Program Run

```

+++++
+                P R O T O - T E C - S H O P                +
+                Y O U R   S H O P P I N G C A R T           +
+++++
3 YOOLOX 10k Wireless Powerbank Qi
                                     36.99 EUR           110.96 EUR
                                     -----
                                     sum net price:      93.24 EUR
                                     tax:                 17.72 EUR
                                     sum gross price:     110.96 EUR
                                     -----
                                     shipping:            5.90 EUR
                                     -----
                                     total:              116.86 EUR
                                     =====

```



```

=====
+++++
+          P R O T O - T E C - S H O P          +
+          Y O U R   S H O P P I N G C A R T     +
+++++
1 ASUS ROG Strix Scar III G731, RAM 16GB, MS Windows OS, 512GB SSD + 1TB SSHD
                                1887.99 EUR          1887.99 EUR

2 Apple iPhone 11 Pro 256GB, RAM 4GB, iOS phone
                                1305.99 EUR          2611.98 EUR

1 Samsung Galaxy S10+ SM-G975F/DS Dual SIM, RAM 8GB, Android OS phone
                                787.39 EUR           787.39 EUR

3 YOOLOX 10k Wireless Powerbank Qi
                                36.99 EUR            110.96 EUR

                                -----
                                sum net price:    4536.40 EUR
                                tax:              861.92 EUR
                                sum gross price:   5398.32 EUR
                                -----
                                shipping:         0.00 EUR
                                -----
                                total:            5398.32 EUR
                                =====

```

```

~ShoppingCart()
delete article: ~InCart() 1 ~Notebook() ~Device() ~Article() ASUS ROG Strix Scar III G731
delete article: ~InCart() 2 ~Phone() ~Device() ~Article() Apple iPhone 11 Pro 256GB
delete article: ~InCart() 1 ~Phone() ~Device() ~Article() Samsung Galaxy S10+ SM-G975F/DS Dual SIM
delete article: ~InCart() 3 ~Accessory() ~Article() YOOLOX 10k Wireless Powerbank Qi

```