

Aufgabe: Eine Klasse **prod** soll Produkte abbilden und die **private**-Member **string pn** für den Produktnamen, **double preis** für den Preis, **unsigned long anz** für die Anzahl besitzen.

Ein Default-Konstruktor soll diese 3 Member mit Parametern initialisieren können, wobei die Defaultwerte "" (für string), 0.0 (für preis) und 0UL (für anz) anzugeben sind.

Der Konstruktor und der Destruktor sollen die Werte der Objekte zur Kontrolle noch einmal auf der Konsole ausgeben.

Die Methode **void out()** sollen die 3 Member ebenfalls auf Konsole ausgeben.

Die Methode **void ein()** soll die Member über Tastatur einlesen (**string** mit **getline(cin,pn)**).

Weiterhin sind für jedes Member eine öffentliche **get**-Methode zur Rückgabe des Members mit **return** und eine öffentliche **set**-Methode zum Schreiben des Members (mit Parameter und Default-Wert) zu implementieren. Wegen **string** ist **#include <string>** notwendig.

Die Klasse **stack** mit den **private**-Members **prod *stapel**, **unsigned long m** (max. Anzahl) und **unsigned long top** (Stapelindex) soll einen Stapel-Speicher (LIFO) mit maximaler Länge **m** abbilden.

Der Default-Konstruktor soll als Defaultparameter **m=3UL** enthalten, **top** mit **0UL** initialisieren und dynamisch stapel in der Länge **m** anlegen. Der Stapelindex **top** muß immer der Index des ersten freien Elements im Stapel sein.

Der Kopierkonstruktor legt eine Kopie der **stack**-Instanz an, die als Parameter übergeben wird. Für beide Instanzen muß **stapel** auf voneinander disjunkte Speicher zeigen !

Der Destruktor gibt **stapel** mit **delete** frei und setzt **top** , **m** und **stapel** auf 0..

Die Methode **bool IsFull()** soll **false** liefern, wenn **top** kleiner als **m** ist, sonst **true**.

Die Methode **bool IsEmpty()** soll **true** liefern, wenn **top == 0UL**, sonst **false**.

Die Methode **void Push(prod p)** legt **p** oben auf den Stapel, wenn der Stapel nicht voll ist.

Die Methode **void Pop()** entfernt das oberste Element, wenn der Stapel nicht leer ist.

Die Methode **prod Top()** gibt das oberste Element des Stapels zurück, **ohne** es zu löschen. Ist der Stapel in **Top()** leer, dann wird ein Pseudoelement vom Typ **prod** mit den Werten **pn=""**, **preis=-1.0**, **anz=0UL** geliefert. Die Methode **prod PopTop()** gibt das oberste Element des Stapels zurück und **löscht** es aus dem Stapel (hierfür **Top()** und **Pop()** nutzen).

Die Methode **unsigned long Length()** gibt den Wert von **top** zurück.

Eine Methode **double gpreis()** soll für alle Elemente des Stapels die Summe der Produkte aus Preis mal Anzahl zurückgeben, ohne den Stapel zu verändern.

Die Klassen sind mit folgender **main()**-Funktion zu testen.

```
int main(){
    stack ps; prod pp("Cu",12.0, 5); ps.Push(pp); pp.out();

    pp.set_pn("Pb"); pp.set_preis(16.00); pp.set_anz(10); ps.Push(pp);
    cout<<"\n  Name = "<<pp.get_pn()
        <<"\n  Preis = "<<pp.get_preis()
        <<"\nAnzahl = "<<pp.get_anz()<<endl;

    pp.ein(); ps.Push(pp);
    cout<<"Elementezahl= "<<ps.Length()<<endl;
    cout.setf(ios::boolalpha); cout<<"ps.IsFull() = "<<ps.IsFull()<<endl;
    cout<<"Gesamtpreis = "<<ps.gpreis()<<endl;
    pp=ps.Top(); ps.Pop(); cout<<"Elementezahl= "<<ps.Length()<<endl;
    pp.out();
    pp = ps.PopTop(); cout<<"Elementezahl= "<<ps.Length()<<endl; pp.out();

    pp = ps.PopTop(); cout<<"Elementezahl= "<<ps.Length()<<endl; pp.out();

    cout.setf(ios::boolalpha); cout<<"ps.IsEmpty() = "<<ps.IsEmpty()<<endl;

    ps.Pop(); pp = ps.Top(); pp.out(); cin.get(); return 0;
}
```

```
Konstruktor prod, pn =
preis =      0.00
anz =        0

Konstruktor prod, pn =
preis =      0.00
anz =        0

Konstruktor prod, pn =
preis =      0.00
anz =        0

Konstruktor stack, m = 3

Konstruktor prod, pn = Cu
preis =      12.00
anz =        5

Destruktor prod, pn = Cu
preis =      12.00
anz =        5

Ausgabe Produkt:
Produktname = Cu
Preis       =      12.00
Anzahl      =        5

Name = Pb
Preis = 16.00
Anzahl = 10

Destruktor prod, pn = Pb
preis =      16.00
anz =        10

Eingabe Produkt:
Produktname = Kupfer
Preis       = 42
Anzahl      = 100

Destruktor prod, pn = Kupfer
preis =      42.00
anz =      100
Elementezahl= 3
ps.IsFull() = true

Konstruktor prod, pn =
preis =      0.00
anz =        0

Konstruktor prod, pn =
preis =      0.00
anz =        0

Konstruktor prod, pn =
preis =      0.00
anz =        0

Kopierkonstruktor stack, m = 3

Destruktor prod, pn = Kupfer
preis =      42.00
anz =      100

Destruktor prod, pn = Kupfer
preis =      42.00
anz =      100

Destruktor prod, pn = Pb
preis =      16.00
anz =        10

Destruktor prod, pn = Pb
preis =      16.00
anz =        10
```

```
Destruktor prod, pn = Cu
preis =      12.00
anz =        5

Destruktor prod, pn = Cu
preis =      12.00
anz =        5

Destruktor prod, pn = Kupfer
preis =      42.00
anz =      100

Destruktor prod, pn = Pb
preis =      16.00
anz =        10

Destruktor prod, pn = Cu
preis =      12.00
anz =        5

Destruktor stack, m = 3
Gesamtpreis = 4420.00

Destruktor prod, pn = Kupfer
preis =      42.00
anz =      100
Elementezahl= 2

Ausgabe Produkt:
Produktname = Kupfer
Preis       =      42.00
Anzahl      =      100

Destruktor prod, pn = Pb
preis =      16.00
anz =        10

Destruktor prod, pn = Pb
preis =      16.00
anz =        10
Elementezahl= 1

Ausgabe Produkt:
Produktname = Pb
Preis       =      16.00
Anzahl      =        10

Destruktor prod, pn = Cu
preis =      12.00
anz =        5

Destruktor prod, pn = Cu
preis =      12.00
anz =        5
Elementezahl= 0

Ausgabe Produkt:
Produktname = Cu
Preis       =      12.00
Anzahl      =        5
ps.IsEmpty() = true
Stack is empty

Konstruktor prod, pn = null
preis =      -1.00
anz =        0

Destruktor prod, pn = null
preis =      -1.00
anz =        0

Ausgabe Produkt:
Produktname = null
Preis       =      -1.00
Anzahl      =        0
```

```
Destruktor prod, pn = null
preis =      -1.00
anz =        0

Destruktor prod, pn = Kupfer
preis =      42.00
anz =       100

Destruktor prod, pn = Pb
preis =      16.00
anz =        10

Destruktor prod, pn = Cu
preis =      12.00
anz =         5

Destruktor stack, m = 3
*/
```