

EDV-Schulen des Landkreises Deggendorf**1. Probe Schulaufgabe in ABAP**

Nachname, Vorname:

Dauer: 70 Minuten**Insgesamt: 50 Punkte**Windows Kennung:

ABAP Prüfungskennung:

Erstellen Sie für das Hauptprogramm einen neuen Report mit der Bezeichnung **ZXXX_XX_PROBE_SA1_2023** (z.B. „Z007_TP_PROBE_SA1_2023“). Kopieren Sie anschließend den Code von Vorlage.txt in ihren Report.

Hinweise:

- Lesen Sie sich die Aufgabenstellung erst **komplett** und **exakt** durch. Halten Sie sich an die abgebildeten Ausgaben (auf der letzten Seite)!
- Verwenden Sie passende Datentypen. Halten Sie sich an die bekannten Namenskonventionen aus dem Unterricht.
- Sofern keine Fehlerprüfung gefordert ist, muss diese auch nicht gemacht werden!
- Bei jeder Aufgabe sind die voraussichtlich erreichbaren Punkte angegeben.

Am Ende der Prüfung:

- **Aktivieren Sie bei Bearbeitungsende ihr Hauptprogramm und alle globalen Klassen!**
- Füllen Sie bei Bearbeitungsende die Kommentarfelder mit ihrem Namen aus! Kopieren Sie den Inhalt Ihres gesicherten Quelltextes (jeder einzelnen Aufgabe) in die Datei:
H:\Daten\nachname_aufgaben.txt

In den angelegten Klassen wechseln sie zum quelltextbasierten Editor und kopieren diesen Code in die Textdatei.

Viel Erfolg!

Perl



Weihnachten steht vor der Tür. Planen Sie mit folgenden Klassen Ihren Weihnachtseinkauf. Alle Klassen müssen global angelegt werden. Definieren Sie so wenig wie möglich public! Achten Sie auf redundanzfreie und performante Programmierung.

1) Klasse ZXXX_XX_CL_EINKAUF (ca. 20 Punkte)

Erstellen Sie eine abstrakte Klasse ZXXX_XX_CL_EINKAUF (wobei XX ihren Initialen entspricht, z.B. Z007_TP_CL_EINKAUF):

- Ein Einkauf hat eine Nummer (INTEGER), eine Bezeichnung (STRING) und einen Listenpreis (DEC10_2). Definieren Sie dafür innerhalb der Klasse einen Strukturtyp TS_EINKAUF und verwenden diesen Typen für das einzige Instanz-Attribut der Klasse.
- Die Nummer wird bei Anlage des Objekts automatisch gesetzt und muss beim Aufruf des Konstruktors nicht übergeben werden (d.h. es wird nur Bezeichnung und Preis übergeben). Der erste Einkauf erhält die Nummer 10. Bei jedem weiteren Objekt muss die Nummer um 10 erhöht werden. Legen Sie für die automatische Nummerierung ein statisches Attribut GN_AKT_NUMMER an, dass nur innerhalb der Klasse ZXXX_XX_CL_EINKAUF sichtbar ist.
- Die folgenden Methoden sind öffentlich sichtbar, falls nichts anderes angegeben ist:
 - Die funktionale Methode GET_BEZEICHNUNG liefert die Bezeichnung des Einkaufs als STRING zurück.
 - Die funktionale Methode GET_LISTENPREIS ist protected und liefert den Listenpreis für den Einkauf zurück.
 - Die abstrakte und funktionale Methode CALC_PREIS liefert den berechneten Endpreis des Einkaufs zurück. Dieser Endpreis muss später in den jeweiligen Unterklassen berechnet werden.
 - Die Methode DISPLAY_ATTRIBUTES gibt mit Hilfe von WRITE die Nummer, Bezeichnung und den Endpreis am Bildschirm aus. Für die Ausgabe von Endpreis und Bezeichnung werden die entsprechenden Methoden CALC_PREIS bzw. GET_BEZEICHNUNG verwendet.
 - Die funktionale Methode GET_AKT_NUMMER liefert die Nummer des letzten Einkaufs zurück.

2) Klasse ZXXX_XX_CL_GESCHENK (ca. 10 Punkte)

Erstellen Sie eine Klasse ZXXX_XX_CL_GESCHENK:

- Diese Klasse erbt von ZXXX_XX_CL_EINKAUF.
- Eine Instanz der Klasse verfügt über ein weiteres Attribut, nämlich den Namen des Beschenkten (STRING). Dieser wird beim Konstruktor zusätzlich übergeben.
- Beim Endpreis für ein Geschenk wird immer ein Rabatt gewährt, der vom Listenpreis abhängig ist. Dieser wird dann von der Methode CALC_PREIS zurückgegeben.

Listenpreis größer als	Rabatt
50,00 €	5 Prozent
250,00 €	9 Prozent
500,00 €	13 Prozent

- Die Methode GET_BEZEICHNUNG muss redefiniert werden. Die zurückgegebene Bezeichnung für ein Geschenk setzt sich immer zusammen aus:

Bezeichnung Einkauf „für“ Beschenkter

3) Klasse ZXXX_XX_CL_NAHRUNG (ca. 10 Punkte)

Erstellen Sie eine Klasse ZXXX_XX_CL_NAHRUNG:

- Diese Klasse erbt von ZXXX_XX_CL_EINKAUF.
- Eine Instanz verfügt über ein weiteres Attribut, nämlich die eingekaufte Menge (INTEGER). Diese wird beim Konstruktor zusätzlich übergeben. Ist die übergebene Menge 0 oder kleiner, so wird diese im Konstruktor auf 1 gesetzt.
- Die Methode GET_BEZEICHNUNG muss redefiniert werden. Die zurückgegebene Bezeichnung für eine Nahrung setzt sich immer zusammen aus:
Bezeichnung Einkauf „a“ Listenpreis „(“ Menge „Stück)“
- Bei der Methode CALC_PREIS wird der Listenpreis mit der Menge multipliziert und als Ergebnis für den Endpreis zurückgegeben.

4) Hauptprogramm ZXXX_XX_PROBE_SA1_2023 (ca. 10 Punkte)

Legen Sie zwei Objekte der Klasse ZXXX_XX_CL_GESCHENK („Buch“, 59.90, „Maxi“), („Schlemmerkiste“, 279.99, „Romina“) und zwei Objekte der Klasse ZXXX_XX_CL_NAHRUNG („Kasten Augustiner“, „19.99“, 3), („1 Kilo Rinderfilet“, 26.00, –5) an.

Legen Sie eine interne Tabelle passenden Typs an, die Einkäufe speichern kann und legen Sie die erzeugten Objekte in dieser ab.

Mit Hilfe einer Schleife sollen die Einkäufe ausgegeben werden. Zusätzlich wird innerhalb der Schleife die Summe aller Einkäufe (aller Endpreise) berechnet und am Ende ausgegeben.

Beim Ausführen des Hauptprogramms sollte sich folgendes Bild ergeben.

Ausgabe:

Meine Einkaufsliste für Weihnachten

Nummer des letzten Einkaufs: 40

10	Buch für Maxi	56,91
20	Schlemmerkiste für Romina	254,79
30	Kasten Augustiner a 19.99 (3 Stück)	59,97
40	1 Kilo Rinderfilet a 26.00 (1 Stück)	26,00

Summe gesamt: 397,67

Vorlage

Hauptprogramm (siehe auch Vorlage.txt):

```
DATA: go_geschenk      TYPE REF TO zxxx_xx_cl_geschenk,  
      go_nahrung       TYPE REF TO zxxx_xx_cl_nahrung,  
      gn_summe         TYPE dec10_2.
```

```
WRITE: / 'Meine Einkaufsliste für Weihnachten'.
```

```
ULINE.
```

```
*WRITE: / 'Nummer des letzten Einkaufs:', get_akt_nummer( ).
```

```
ULINE.
```

```
ULINE.
```

```
WRITE: /30 'Summe gesamt:', 45 gn_summe.
```