



## Klausur

<b>Anrede</b>	<input type="checkbox"/> Frau	<input type="checkbox"/> Herr	<input type="checkbox"/> _____	<b>Datum:</b>	01.01.2001
<b>Name, Vorname</b>	Mustermann, Michael			<b>MatrikelNr:</b>	12345
<b>Ich möchte mein Ergebnis per E-Mail an nachfolgende Adresse erhalten:</b>					
<b>E-Mailadresse:</b>	m.mustermann@gmx.de				
(Schreiben Sie deutlich. Lassen Sie die Angabe leer, wenn Sie keine Mail von Prof. Off wollen.)					
<b>Dies ist mein letzter Versuch:</b>					
<input type="checkbox"/> Ja <input type="checkbox"/> Nein					

Wird von der Lehrkraft ausgefüllt					
<b>Aufgabe 1</b>		/ 10	7	Minuten	
<b>Aufgabe 2</b>		/ 7,5	8	Minuten	
<b>Aufgabe 3</b>		/ 13,5	10	Minuten	
<b>Aufgabe 4</b>		/ 16	20	Minuten	
<b>Aufgabe 5</b>		/ 40	30	Minuten	
<b>Aufgabe 6</b>		/ 13	15	Minuten	
<b>Gesamtpunkte</b>		/ 100	Zeit	90	Minuten
<b>Note</b>			1,0		

Notenübersicht	1,0	1,3	1,7	2,0	2,3	2,7	3,0	3,3	3,7	4,0
≥ Prozent	95%	90%	85%	80%	75%	70%	65%	60%	55%	50%
≥ Punkte	95	90	85	80	75	70	65	60	55	50

Bitte legen Sie Ihren Personalausweis und Studierendenausweis bereit. Es sind keine Hilfsmittel zulässig. Zusätzliches Papier wird vom Dozenten ausgegeben. Es darf nicht abgeschrieben und auch nicht abgeguckt werden. Auch der Versuch führt zum Ausschluss von der Klausur und der Bewertung der Leistung mit 5,0. Der Toilettenbesuch ist nach Anmeldung bei der Lehrkraft möglich. Wenn der erste Teilnehmer die Klausur abgegeben hat, sind keine Toilettenbesuche mehr möglich.

**Bitte wenden.**



**Aufgabe 1**

\_\_\_ / 10 Punkte

**Frage a)**

Wofür stehen die Buchstaben EVA im "EVA-Prinzip"? Welcher Buchstabe fehlt und warum?

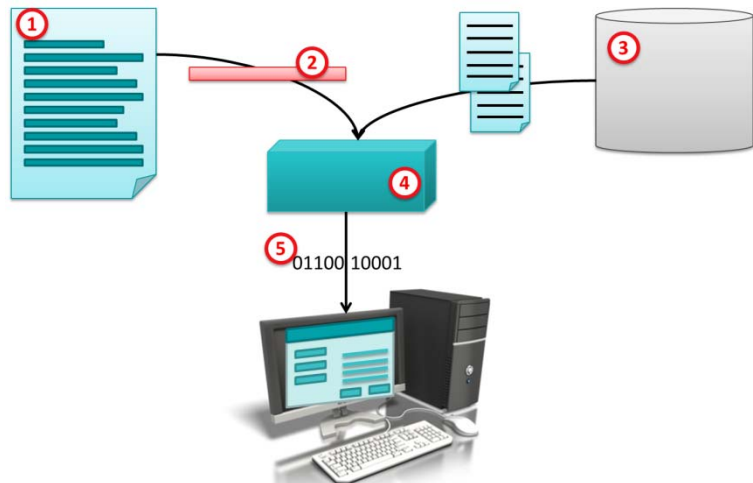
**Antwort a)**

E	Eingabe	___ / 0,5 P
V	Verarbeitung	___ / 0,5 P
A	Ausgabe	___ / 0,5 P
S	Fehlender Buchstabe (0,5P) und Begründung (1P): Speicherung, weil Daten nach Eingabe und Verarbeitung nicht nur ausgegeben, sondern dauerhaft gespeichert werden können; siehe LE01, Folie 42	___ / 1,5 P

**Frage b)**

Die Abbildung stellt die Arbeitsweise eines Interpreters dar.

Bennen Sie die mit Zahlen markierten Elemente aus der Abbildung (2,5P)! Worin unterscheidet sich der Interpreter hauptsächlich vom Compiler (2,5P)?



**Antwort b)**

1	(siehe LE02, Folie Nr. 124 f.) Quellcode des Programms (0.5P)	<b>Hauptunterschied zum Compiler:</b> <i>Korrekturhinweis: max. 2,5P aus folgenden Antworten:</i>  liest jeweils eine Anweisung aus dem Quellcode (1P), übersetzt sie in Maschinencode (0.5P) und führt sie aus (0.5P); fährt dann mit der nächsten Anweisung fort (1P)  <i>Korrekturhinweis: Alternativ auch Punkte für die Abgrenzung zum Compiler, z.B.</i>  Compiler übersetzt das gesamte Programm, um es auszuführen (1P)
2	eine Zeile/eine Anweisung aus dem Quellcode (0.5P)	
3	Bibliotheken (0.5P)	
4	Interpreter (0.5P)	
5	Maschinencode (0.5P)	



**Frage c)**

Was ist in der Phase "Analyse und Entwurf" des Softwarelebenszyklus zu tun (2P)?

**Antwort c)**

LE01, Folie Nr. 64

*Korrekturhinweis: max. 2P aus folgenden Antworten:*

Analyse der Anforderungen (0.5P) und Entwurf einer oder mehrerer geeigneter Lösung, WIE (1P) das System arbeiten soll, ggf. Entscheidung über die umzusetzende Lösung (0.5P); Bei Unklarheiten/Lücken Rückkehr zur Anforderungsanalyse (1P).

**Aufgabe 2**

\_\_\_ / 7,5 Punkte

**Frage**

Bestimmen Sie den Wert der nachfolgenden Ausdrücke. Geben Sie an, welcher Datentyp geeignet wäre, den Wert ohne Verlust der Genauigkeit bzw. Informationsverlust zu speichern. Wenn mehrere Datentypen möglich sind, wählen Sie den Datentyp mit dem geringsten Speicherbedarf.

**Antwort**

Ausdruck	Ergebnis der Auswertung/Wert des Ausdrucks 0.5P	Geeigneter Datentyp (mit kleinstem Speicherbedarf) 1P	Punkte
$0.000000001 + 1$	1,000000001	Double	___ / 1,5 P
False XOR (False AND True)	False/Falsch	Boolean	___ / 1,5 P
$800 - 600 / 3$	600	Integer	___ / 1,5 P
$10 * 3.33 = 33.3$	True/Wahr	Boolean	___ / 1,5 P
#12/31/2015# + 1	01.01.2016	Date	___ / 1,5 P

**Bitte wenden.**



### Aufgabe 3

\_\_\_ / 13,5 Punkte

#### Frage

Definieren Sie einen zusammengesetzten Typ für einen Mietwagen. Der Mietwagen soll folgende Eigenschaften haben: Marke (z.B. Audi, BMW), Anzahl Sitzplätze, Kilometerstand, und Mietpreis pro Tag. Wählen Sie selbst passende Datentypen und geeignete Bezeichner für diese Eigenschaften. Schreiben Sie eine Prozedur, in der Sie eine Variable dieses Typs deklarieren. Initialisieren Sie bei dieser Variable die Eigenschaft Marke mit einem selbst gewählten Wert.

#### Antwort

```
Option Compare Database
Option Explicit

'2P Type, 0.5P Bezeichner, 1P im Modul
Type TMietwagen
'0.5P Alle Bezeichner, 1P alle Datentypen (passend)
strMarke As String
bytAnzahlPlaetze As Byte
lngKilometer As Long
curPreis As Currency
End Type '1P

'0.5P Sub, 0.5P Bezeichner
Sub mietwagen()

'0.5P Dim, 0.5P Bezeich '2P Datentyp
Dim wgnMeinMietwagen As TMietwagen

'0.5P Let, 0.5P Bezeich, 0.5P Punkt, 1P Eigenschaft, 0.5P Wert
Let wgnMeinMietwagen.strMarke = "Ford"

End Sub ' 0.5P
```



## Aufgabe 4

\_\_\_ / 16 Punkte

### Frage

Schreiben Sie eine private Funktion `potenz`, die die Potenz einer Zahl bildet (z.B.  $4^3 = 4*4*4 = 64$  oder  $12^2 = 12*12 = 144$ ) und das Ergebnis als ganze Zahl zurückliefert. Die Funktion soll zwei ganze Zahlen als Parameter übergeben bekommen. Eine Zahl ist die Basis und die andere der Exponent (z.B. bei  $4^3$  ist 4 die Basis und 3 der Exponent). Gehen Sie wie folgt vor: Deklarieren Sie eine Variable für die Potenz. Initialisieren Sie diese Variable mit einem geeigneten Wert. Benutzen Sie eine Zählerschleife, die von 1 bis zum Wert des Exponenten läuft. Multiplizieren Sie innerhalb der Schleife die Basis mit der Variable für die Potenz. Weisen Sie das Ergebnis der Multiplikation der Variable für die Potenz zu. Geben Sie nach der Schleife den Wert der Variablen als Ergebnis der Funktion zurück.

### Antwort

```
Option Compare Database
Option Explicit
```

```
'Private 0.5P, Function 0.5P, Bezeichner 0.5P, 1P Parameter, 1P Datentyp, 1P Rückgabewert
Private Function potenz(pintBasis As Integer, pintExponent As Integer) As Long
```

```
' 0.5P Dim, 0.5P Bezeichner gültig und sinnvoll
```

```
' 0.5P As, 1P passender Datentyp
```

```
Dim i As Integer ' Muss zum Datentyp des Exponenten passen
```

```
Dim lngPotenz As Long ' Wichtig, Datentyp muss zum Typ des Rückgabewertes passen
```

```
'Für alle
```

```
' 0.5P Let + Bezeichner, 0.5P Zuweisung, 0.5P Wert
```

```
Let lngPotenz = 1
```

```
'0.5P For+To, 0.5 Schleifenvar, 0.5 Initialisierung, 0.5 Exponent+ 1P Parameterverwendung
```

```
For i = 1 To pintExponent
```

```
'0.5P Let+Bezeichner, 1P Parameterverwendung, ' 0.5P Ausdruck
```

```
Let lngPotenz = lngPotenz * pintBasis
```

```
Next '0.5P
```

```
'1.5P Rückgabe des Ergebnisses
```

```
Let potenz = lngPotenz
```

```
End Function '0.5P
```



## Aufgabe 5

\_\_\_ / 40 Punkte

### Frage a)

Schreiben Sie im Modul mdlShop eine öffentliche Funktion `pruefeVerfuegbarkeit`, die als Parameter eine Kleidungsgröße als ganze Zahl übergeben bekommt (z.B. 38 oder 42). Die Funktion soll als Ergebnis zurückliefern, ob ein Kleidungsstück in dieser Größe im Online-Shop lieferbar ist oder nicht. Wenn es Größe 38 oder Größe 40 ist, dann ist das Kleidungsstück lieferbar. In allen anderen Fällen ist es nicht lieferbar. Verwenden Sie **nicht** die If-Then-Else-Verzweigung. Das Modul mdlShop beginnt hier:

### Antwort a)

```
Option Compare Database
Option Explicit
```

```
' Lösung A
```

```
' Public 0.5P, Function 0.5P, Bezeichner 0.5P
```

```
' 1P Parameter, 1P Datentyp, 1P Rückgabe
```

```
Public Function pruefeVerfuegbarkeit(pbytGroesse As Byte) As Boolean
```

```
    ' 0.5P Dim, 0.5P Bezeichner gültig und sinnvoll
```

```
    ' 0.5P As, 1P passender Datentyp
```

```
    Dim bolVerfuegbar As Boolean
```

```
    '2P Select Case+Case+Case Else+End Select
```

```
    '2P Parameterverwendung
```

```
    Select Case pbytGroesse
```

```
        Case 38
```

```
            Let bolVerfuegbar = True '0.5P Gr. 38 lieferbar
```

```
        Case 40
```

```
            Let bolVerfuegbar = True '0.5P Gr. 40 lieferbar
```

```
        Case Else
```

```
            Let bolVerfuegbar = False '0.5P Gr. Rest nicht lieferbar
```

```
    End Select
```

```
    '2P Rückgabe des Ergebnisses
```

```
    Let pruefeVerfuegbarkeit = bolVerfuegbar
```

```
End Function '0.5P
```

```
' Alternative Lösung B, aber nicht erlaubt wg. If-Then-Else
```

```
Public Function pruefeVerfuegbarkeit2(pbytGroesse As Byte) As Boolean
```

```
    Dim bolVerfuegbar As Boolean
```

```
    If pbytGroesse = 38 Or pbytGroesse = 40 Then
```

```
        Let bolVerfuegbar = True
```

```
    Else
```

```
        Let bolVerfuegbar = False
```

```
    End If
```

```
    Let pruefeVerfuegbarkeit2 = bolVerfuegbar
```

```
End Function
```

```
' Fortgeschrittene Lösung C
```

```
Public Function pruefeVerfuegbarkeit3(pbytGroesse As Byte) As Boolean
```

```
    Let pruefeVerfuegbarkeit3 = (pbytGroesse = 38 Or pbytGroesse = 40)
```

```
End Function
```



### Frage b)

Gegeben ist das folgende Formular. Die Namen der Oberflächenelemente sind angegeben.



Schreiben Sie eine Ereignisprozedur, die aufgerufen wird, sobald der Benutzer die Auswahl in der Aufklappliste "Größe" ändert. Ermitteln Sie dann den in der Aufklappliste ausgewählten Wert. (Gehen Sie davon aus, dass die Aufklappliste den Wert aus der ersten Spalte liefert.) Rufen Sie die Funktion im Modul mdlShop aus Aufgabe a) auf und übergeben Sie diesen Wert als Parameter. Wenn die Funktion `false` zurückliefert, deaktivieren Sie die Schaltfläche "In den Warenkorb legen" und machen den Text "Leider Ausverkauft!" sichtbar. Andernfalls aktivieren Sie die Schaltfläche "In den Warenkorb legen" und machen den Text "Leider Ausverkauft!" unsichtbar.

### Antwort b)

```
Option Compare Database
Option Explicit

'Private 0.5P, Sub 0.5P, Bezeichner 2P, Ereignis 2P
Private Sub cmbSize_Change()
'Für alle
' 0.5P Dim, 0.5P Bezeichner gültig und sinnvoll
' 0.5P As, 1P passender Datentyp
Dim bytGroesse As Byte
Dim bolLieferbar As Boolean

'Für alle
' 0.5P Let, 0.5P Bezeichner, 0.5P Zuweisung
' Me 1P, 1P Value
Let bytGroesse = Me.cmbSize.Value '1P Bezeichner

' Zusätzlich Zuweisung Rückgabewert 2P, Modul 1P,
' Funktion 1P, Parameter 1P
Let bolLieferbar = mdlShop.pruefeVerfuegbarkeit(bytGroesse)

'2P If + Then + Else + End If, 0.5P Bedingung
' 2P alle Enabled/Visible, 1P alle Werte
' 2P Bezeichner alle Bezeichner
If bolLieferbar = False Then
Let Me.lblAusverkauft.Visible = True
Let Me.btnHinzufuegen.Enabled = False
Else
Let Me.lblAusverkauft.Visible = False
Let Me.btnHinzufuegen.Enabled = True
End If

End Sub '0.5P
```

**Bitte wenden.**



## Aufgabe 6

\_\_\_ / 13 Punkte

### Frage

Gegeben ist das folgende Struktogramm.  
Programmieren Sie es als Prozedur in VBA!

### Antwort

```
Option Compare Database  
Option Explicit
```

```
' Sub+Bezeichner 0.5P  
Sub berechne()
```

```
'Für alle  
' 0.5P Dim, 0.5P Bezeichner gültig und sinnvoll  
' 0.5P As, 1P passender Datentyp  
Dim i As Byte  
Dim n As Integer  
Dim f As Long
```

```
'Für alle  
' 0.5P Let, 0.5P Bezeichner, 0.5P Zuweisung+Wert  
Let f = 1
```

```
' zusätzlich 1P InputBox, 0.5P Typumdw/Val  
Let n = Val(InputBox("Eine ganze Zahl bitte:"))
```

```
'1P If+Then, 0.5P Bedingung  
If n > 0 Then  
Do '1P  
  Let f = f * i '0.5P  
  Let i = i + 1 '0.5P  
' 1P Loop, 0.5P Until, 0.5P Bedingung  
  Loop Until i > n  
End If
```

```
Debug.Print n ' 0.5P  
Debug.Print f ' 0.5P
```

```
End Sub '0.5P
```

berechne

