

Klausur

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

Wird von der Lehrkraft ausgefüllt				
Aufgabe 1		0/ 3	10	Minuten
Aufgabe 2		0/ 16	10	Minuten
Aufgabe 3		0/ 5	20	Minuten
Aufgabe 4		0/ 31	30	Minuten
Aufgabe 5		0/ 15	20	Minuten
Gesamtpunkte		0/ 70	Zeit	0 Minuten
Note		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	66,5	63	59,5	56	52,5	49	45,5	42	38,5	35

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 abgesehen 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 und Abgabe des Handys möglich. Nachdem der erste Teilnehmer seine Klausur abgegeben hat, sind keine Toilettenbesuche mehr möglich.

Bitte wenden.

Aufgabe 1

_____ / 3 Punkte

Frage a)

Was ist in der Phase "Auslieferung" des Softwarelebenszyklus zu tun? (1P)

Antwort a)

In der Phase "Auslieferung" des Softwarelebenszyklus werden folgende Aktivitäten durchgeführt:

Auslieferung/Installation: Die Software wird auf die Systeme des Kunden oder in die Produktionsumgebung übertragen.
Herstellung eines betriebsbereiten Zustands: Die Software wird so konfiguriert und getestet, dass sie vollständig betriebsbereit ist und ihre Funktionen wie vorgesehen erfüllt.
Schulung und Dokumentation: Benutzer und Administratoren werden geschult, und es wird entsprechende Dokumentation bereitgestellt, um den Betrieb und die Nutzung der Software zu unterstützen.

Verweise auf die Folien:

Siehe Folien zur "Auslieferung" in LE01, insbesondere Folie 71.

Frage b)

Welchem gemeinsamen Zweck dienen Checkboxen und Mehrfachauswahllisten? (0,5P)

Wann verwenden Sie Checkboxen und wann besser Mehrfachauswahllisten? (1,5P)

Antwort b)

Checkboxen und Mehrfachauswahllisten dienen dem Zweck, dem Benutzer die Möglichkeit zu geben, aus einer Liste von Optionen mehrere auszuwählen.

Checkboxen: Verwenden, wenn die Anzahl der Auswahlmöglichkeiten gering ist und alle Optionen auf einmal sichtbar sein sollen. Checkboxen erlauben eine einfache Mehrfachauswahl und sind intuitiv verständlich. Die Anzahl der Optionen sollte sich nicht mehr ändern.

Mehrfachauswahllisten: Verwenden, wenn die Anzahl der Auswahlmöglichkeiten groß ist oder der Platz begrenzt ist. Mehrfachauswahllisten bieten eine Scroll-Funktion, wodurch sie platzsparend sind und eine große Anzahl von Optionen übersichtlich darstellen können. Wenn zu erwarten ist, dass sich die Anzahl der Optionen ändern sind Mehrfachauswahllisten richtig.

Verweise:
LE 08, Folien zu Benutzeroberflächen und Ereignisverarbeitung

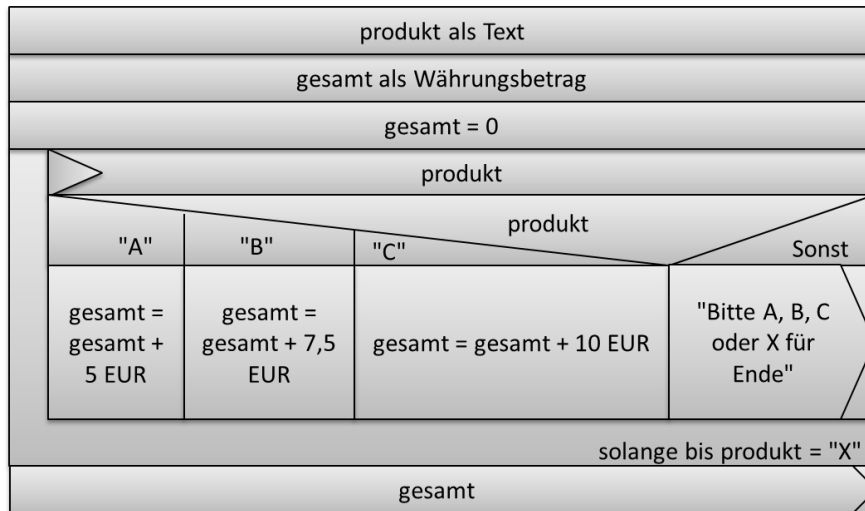
Aufgabe 2B

_____ / 16 Punkte

Frage

Gegeben ist das folgende Struktogramm. Schreiben Sie den dargestellten Algorithmus als VBA-Prozedur!

berechneGesamtpreis



Antwort

```
Option Compare Database
Option Explicit
```

```
' 0.5P Sub, 0.5P Bezeichner
Sub berechneGesamtpreis()
```

```
' Alle Deklarationen (mit richtigem Datentyp, Bezeichner, Namenskonvention) 1.5P
```

```
Dim strProdukt As String
Dim curGesamtpreis As Currency
```

```
Let curGesamtpreis = 0 '0.5P
```

```
' Nachprüfende Schleife
Do '1P Do
```

```
' Let inkl. Zuweisung 0.5P, KEIN Val 1P, InputBox 1P
Let strProdukt = InputBox("Produktbezeichnung (A, B, C oder X zum Beenden)")
```

```
Select Case strProdukt '0.5P Select Case, 0.5P Produkt
```

```
' Alle Case: 3P inkl. Case, Ausdruck und Zuweisung
Case "A"
```

```
Let curGesamtpreis = curGesamtpreis + 5
```

```
Case "B"
```

```
Let curGesamtpreis = curGesamtpreis + 7
```

```
Case "C"
```

```
Let curGesamtpreis = curGesamtpreis + 10
```

```
Case Else '1P Case Else
```

```
MsgBox "Bitte Produkt A, B, C oder X für Ende eingeben." '0.5P
```

```
End Select '0.5P End Select
```

```
Loop Until strProdukt = "X" '1P Loop, 1P Until, 0.5P Ausdruck
```

```
' Ausgabe des Gesamtpreises
```

```
Debug.Print "Gesamtpreis: " & curGesamtpreis '0.5P
```

```
End Sub '0.5P
```

Korrekturhinweis: Hier darf es keine Abweichungen vom Algorithmus geben (z.B. Schleifen innerhalb der Verzweigung). Genau das ist Ziel der Aufgabe, weil er durch das Struktogramm fest vorgegeben. Deshalb gibt es Punktabzug, selbst wenn die einzelnen Anweisungen korrekt sind.

Aufgabe 3

_____ / 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	Wert des Ausdrucks	Datentyp (kl. Speicherbedarf)	
#6/1/2024# - 1	#5/31/2024# oder "31.5.2024"	Date oder String	/ 1 P
600 + 12 / 6 - 300	302	Integer	/ 1 P
1.11111111 * 2	2.22222222	Double	/ 1 P
18 / 3 & "€"	"6 €"	String	/ 1 P
Not (True) Xor False	False	Boolean	/ 1 P

Aufgabe 4

_____ / 31 Punkte

Frage

Für eine Klausur wird ein Programm zur Verwaltung der Teilnehmer im Hörsaal benötigt. Der Hörsaal hat 10 Reihen und in jeder Reihe 12 Plätze.

Definieren Sie im gegebenen Modul "mdlKlausur" einen zusammengesetzten Datentyp für einen Studierenden mit der MatrikelNr, der Anzahl der Klausurpunkte (z.B. 64,5), dem Vor- und Nachnamen und einem Kennzeichen, ob sein Studierendenausweis bereits kontrolliert wurde oder noch nicht. Der Datentyp soll außerhalb des Moduls sichtbar sein.

Deklarieren Sie auf Ebene des Moduls ein zweidimensionales Feld für die Klausur, in dem sich Studierende speichern lassen. Der Gültigkeitsbereich des Feldes soll das ganze Modul sein. Das Feld soll außerhalb des Moduls nicht sichtbar sein.

Schreiben Sie eine Prozedur "anzeigen", die außerhalb des Moduls nicht sichtbar sein soll. Tragen Sie innerhalb der Prozedur zwei Studierende in das Feld ein: Laura Stern mit Matr.-Nr. 987654 und 35,5 Punkten sitzt in der 1. Reihe auf dem 7. Platz. Ihr Ausweis wurde bereits kontrolliert. Ali Yilmaz mit Matr.-Nr. 876543 und 75 Punkten sitzt in der 8. Reihe auf dem 10. Platz. Sein Ausweis wurde noch nicht kontrolliert.

Geben Sie mit Hilfe von zwei Zählerschleifen alle im Feld gespeicherten Studierenden mit der Matrikelnummer im Direktbereich aus. Zählen Sie dabei, bei wie vielen die Ausweise bereits kontrolliert sind. Geben Sie nach der Schleife diese Anzahl in einem Meldungsfenster aus.

Antwort

```
Option Compare Database
Option Explicit

'0.5P Public, 0.5P Type, 0.5P Bezeichner, 1P auf Modulebene
Public Type TStudierender

' Alle: 1P Bezeichner, 1P Datentyp
strMatrNr As String ' z.B. 104 oder 212a (weil es kein Zimmer 13 auf der 2. Etage gibt)
strName As String
sngPunkte As Currency
bolKontrolle As Boolean

End Type '0.5

' 0.5P Private, 0.5P Bezeichner, 0.5P Klammern, 0.5P Index mit Komma,
' 1P Größe, 2P Datentyp, 1P auf Modulebene
Private stuKlausur(9, 11) As TStudierender

'0.5P Private, 0.5P Sub, 0.5P Bezeichner
Private Sub ausgeben()

' Alle Deklarationen (mit richtigem Datentyp, Bezeichner) 1.5P
Dim bytReihe As Byte
Dim bytPlatz As Byte
Dim bytKontrollen As Byte

' Beispieldaten eingetragen (mit oder ohne With) für alle zusammen:
' Let 1P, 0.5 Bezeichner, 0.5 Klammern, 0.5P Komma, 1P Index, 0.5P Werte

' 1. Etage, 7. Platz
Let stuKlausur(0, 6).strMatrNr = "987654"
Let stuKlausur(0, 6).strName = "Laura Stern"
Let stuKlausur(0, 6).sngPunkte = 35.5 ' Korrekturhinweis: Auf Dezimalpunkt achten
Let stuKlausur(0, 6).bolKontrolle = True

' 8. Reihe, 10. Platz
With stuKlausur(7, 9)
Let .strMatrNr = "876543"
Let .strName = "Ali Yilmaz"
Let .sngPunkte = 75
Let .bolKontrolle = False
End With

' Zusätzlich für Initialisierung: Let 0.5P, 0.5 Bezeichner, 0.5P Wert
Let bytKontrollen = 0

'Für alle For 0.5P, To 0.5P
For bytReihe = 0 To 9 ' Zusätzlich Schleifenvariable 0.5P, Ende 1P
For bytPlatz = 0 To 11 ' Zusätzlich Schleifenvariable 0.5P, Ende 1P

'1P Zugriff auf Feld, 1P Indizes inkl. Klammern
'1P Zugriff auf Eigenschaften im Feld, 0.5P Debug.Print
Debug.Print stuKlausur(bytReihe, bytPlatz).strMatrNr

'0.5P If, 0.5P Then, 0.5P Ausdruck
If stuKlausur(bytReihe, bytPlatz).bolKontrolle = True Then
Let bytKontrollen = bytKontrollen + 1
End If

Next ' 0.5P Plätze
Next ' 0.5P Reihen

' 0.5P MsgBox ggf. inkl. Text, 0.5P Wert
MsgBox "Anzahl kontrollierter Ausweise: " & bytKontrollen

End Sub ' 0.5P
```

Aufgabe 5

_____ / **15 Punkte**

Frage

Gegeben ist die folgende Funktion, die als Ergebnis liefert, ob eine Banküberweisung zulässig ist oder nicht. Dazu bekommt Sie den Kontostand und den Überweisungsbetrag.

Finden Sie die Testdaten für einen White Box-Test bei dem mindestens **drei** verschiedene Ausführungspfade durchlaufen werden. Schreiben Sie pro Ausführungspfad eine Zeile in die folgende Tabelle. Tragen Sie die bei diesem Ausführungspfad durchlaufenen Zeilennummern in die erste Spalte ein. Tragen Sie in die zweite Spalte ein, welche Testdaten als Parameterwerte für die Ausführung dieses Pfades übergeben werden müssen. Begründen Sie, warum Sie diese Werte gewählt haben. Tragen Sie in die dritte Spalte ein, was dann als Ergebnis der Funktion zurückgegeben wird.

```

1  Function pruefeUeberweisung (pcurKontostand As Currency, pcurBetrag As
   Currency) As Boolean
2
3  Dim bolErgebnis As Boolean
4
5  If pcurBetrag <= 0 Then
6      Let bolErgebnis = False
7
8  ElseIf pcurKontostand >= pcurBetrag Then
9      Let bolErgebnis = True
10
11 Else
12     Let bolErgebnis = False
13 End If
14
15 Let pruefeUeberweisung = bolErgebnis
16 End Function
    
```

Antwort

Ausführungspfad (2P) (3x 2 P)	Parameterwerte als Testdaten (1.5P)/ Begründung (1P) (3x 2.5P)	Ergebnis (0.5P) (3x 0.5P)	Punkte
<p>Korrekturhinweis: Schritte des Ausführungspfad es, die in Klammer stehen bedeuten keinen Punktabzug, wenn sie fehlen. Alle Schritte richtig und richtige Reihenfolge = volle Punktzahl, überwiegend richtige Schritte und Reihenfolge, 0.5P, sonst 0P.</p>	<p>Korrekturhinweis: Werte hier nur exemplarisch; alle anderen Testdaten, die den gleichen Ausführungspfad ergeben bzw. der selben Äquivalenzklasse entsprechen, sind ebenfalls richtig</p>		

(1),2,3,4,(9),10,(11)	pcurKontostand = pcurBetrag = Begründung: Warum diese Werte? pcurKontostand =100 pcurBetrag = -1 Erster Pfad im If wird durchlaufen.	Falsch
(1),2,3,5,6,(9),10,(11)	pcurKontostand = pcurBetrag = Begründung: Warum diese Werte? pcurKontostand =100 pcurBetrag = 1 Zweiter Pfad im If wird durchlaufen.	True
(1),2,3,5,7,8,(9),10,(11)	pcurKontostand = pcurBetrag = Begründung: Warum diese Werte? pcurKontostand =1 pcurBetrag = 100 Dritter Pfad im If wird durchlaufen, weil Kontostand kleiner als Betrag ist	Falsch

Weitere Fälle könnten sein: Kontostand = Betrag, Betrag = 0

Platz für Notizen

- **Declaration Variable:** Dim strName as string
- **Declaration Konstante:** Const pi as double = 3.14159265359
- **Private/Public/Static Name as string**

Array/Array
 • Typumwandlung
 • Explizite:
 Dim objKommazahl as Double
 Dim intEinszahl as Integer
 Dim objKommazahl as Double
 Dim intEinszahl as Integer
 Dim objKommazahl as Double
 Dim intEinszahl as Integer
 Dim objKommazahl as Double
 Dim intEinszahl as Integer

Verzweigung
 • If...Then...Else...ElseIf
 • Select Case
 • Case...

Prozedur
 • Call funktion (strName)
 • Sub funktion (param as string)
 • End Sub
 • 4 Modul
 • Call funktion (param as string)
 • End Sub

Funktion
 • Function funktion (param as string) as string
 • Dim strName as string
 • Return strName
 • End Function

zum gesetzte Datentypen:
 • Dim strName as string
 • Dim intName as Integer
 • Dim dblName as Double
 • Dim objName as Object

einfache Felder (-Array):
 • Dim strName(2) as string
 • Dim intName(2) as Integer
 • Dim dblName(2) as Double
 • Dim objName(2) as Object

dynamische Felder
 • Dim strName as string
 • ReDim strName (upperbound)
 • Dim intName as Integer
 • ReDim intName (upperbound)
 • Dim dblName as Double
 • ReDim dblName (upperbound)
 • Dim objName as Object
 • ReDim objName (upperbound)

mehrdimensionales Feld
 • Dim strName(2, 2) as string
 • Dim intName(2, 2) as Integer
 • Dim dblName(2, 2) as Double
 • Dim objName(2, 2) as Object

Mag
 • Dim strName as string
 • Set objName as Object
 • Dim intName as Integer
 • Dim dblName as Double
 • Dim objName as Object

Schwerpunkte/Ergebnisse:

1. Aufgabe / Problem
2. Anforderungsanalyse
3. Analyse & Entwurf
4. Implementierung
5. Test
6. Validierung
7. Wartung
8. 2. Auftragsbearbeitung

Verzweigung
 • If...Then...Else...ElseIf
 • Select Case
 • Case...

Prozedur
 • Call funktion (strName)
 • Sub funktion (param as string)
 • End Sub

Funktion
 • Function funktion (param as string) as string
 • Dim strName as string
 • Return strName
 • End Function

zum gesetzte Datentypen:
 • Dim strName as string
 • Dim intName as Integer
 • Dim dblName as Double
 • Dim objName as Object

einfache Felder (-Array):
 • Dim strName(2) as string
 • Dim intName(2) as Integer
 • Dim dblName(2) as Double
 • Dim objName(2) as Object

dynamische Felder
 • Dim strName as string
 • ReDim strName (upperbound)
 • Dim intName as Integer
 • ReDim intName (upperbound)
 • Dim dblName as Double
 • ReDim dblName (upperbound)
 • Dim objName as Object
 • ReDim objName (upperbound)

mehrdimensionales Feld
 • Dim strName(2, 2) as string
 • Dim intName(2, 2) as Integer
 • Dim dblName(2, 2) as Double
 • Dim objName(2, 2) as Object

Mag
 • Dim strName as string
 • Set objName as Object
 • Dim intName as Integer
 • Dim dblName as Double
 • Dim objName as Object

Message Box; MsgBox
 • MsgBox ("Text")
 • MsgBox ("Text", vbInformation)
 • MsgBox ("Text", vbQuestion)
 • MsgBox ("Text", vbWarning)
 • MsgBox ("Text", vbError)
 • MsgBox ("Text", vbCritical)

Datentypen Zahlen:
 • Byte
 • Integer
 • Long
 • Single
 • Double
 • Currency
 • Decimal

Verzweigung
 • If...Then...Else...ElseIf
 • Select Case
 • Case...

Prozedur
 • Call funktion (strName)
 • Sub funktion (param as string)
 • End Sub

Funktion
 • Function funktion (param as string) as string
 • Dim strName as string
 • Return strName
 • End Function

zum gesetzte Datentypen:
 • Dim strName as string
 • Dim intName as Integer
 • Dim dblName as Double
 • Dim objName as Object

einfache Felder (-Array):
 • Dim strName(2) as string
 • Dim intName(2) as Integer
 • Dim dblName(2) as Double
 • Dim objName(2) as Object

dynamische Felder
 • Dim strName as string
 • ReDim strName (upperbound)
 • Dim intName as Integer
 • ReDim intName (upperbound)
 • Dim dblName as Double
 • ReDim dblName (upperbound)
 • Dim objName as Object
 • ReDim objName (upperbound)

mehrdimensionales Feld
 • Dim strName(2, 2) as string
 • Dim intName(2, 2) as Integer
 • Dim dblName(2, 2) as Double
 • Dim objName(2, 2) as Object

Mag
 • Dim strName as string
 • Set objName as Object
 • Dim intName as Integer
 • Dim dblName as Double
 • Dim objName as Object