

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	90 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 "Außerbetriebnahme" des Softwarelebenszyklus zu tun? (1P)

Antwort a)

In der Phase "Außerbetriebnahme" des Softwarelebenszyklus werden alle Aktivitäten durchgeführt, die mit dem Ende der Nutzung der Software verbunden sind. Diese umfassen:

- Beendigung der Nutzung: Die Software wird nicht mehr im täglichen Betrieb verwendet.
- Archivierung: Wichtige Daten und Softwarekomponenten werden archiviert, um sie bei Bedarf später wieder nutzen oder analysieren zu können.
- Migration: Datenbestände werden gegebenenfalls in ein neues System migriert, um sicherzustellen, dass keine wichtigen Informationen verloren gehen.
- Entfernung der Software: Die Software wird von allen Geräten und Systemen entfernt, auf denen sie installiert war.

Siehe Folien zur "Außerbetriebnahme" in LE01, insbesondere Folie 74

Frage b)

Welchem gemeinsamen Zweck dienen Radioknöpfe und Aufklapplisten? (0,5P) Wann verwenden Sie Radioknöpfe und wann besser Aufklapplisten? (1,5P)

Antwort b)

Radioknöpfe (Radio Buttons) und Aufklapplisten (Dropdown Lists) dienen dem gemeinsamen Zweck, dem Benutzer die Möglichkeit zu geben, aus einer Liste von Optionen genau eine auszuwählen.

Radioknöpfe: Verwenden Sie Radioknöpfe, wenn die Anzahl der Auswahlmöglichkeiten gering ist und alle Optionen direkt sichtbar sein sollen. Radioknöpfe sind ideal, wenn Sie dem Benutzer die Auswahl einer Option erleichtern möchten, indem alle möglichen Optionen auf einmal angezeigt werden. Die Anzahl der Optionen sollte sich nicht mehr ändern.

Aufklapplisten: Verwenden Sie Aufklapplisten, wenn die Anzahl der Auswahlmöglichkeiten groß ist oder der Platz begrenzt ist. Aufklapplisten sind platzsparend und helfen, das Layout der Benutzeroberfläche übersichtlich zu halten, indem sie nur die aktuell ausgewählte Option anzeigen und die übrigen Optionen in einer ausklappbaren Liste verbergen. Wenn zu erwarten ist, dass sich die Anzahl der Optionen ändern wird, sind Auswahllisten richtig.

Verweise:

LE 08, Folien zu Benutzeroberflächen und Ereignisverarbeitung, insbesondere die Folien zur Gestaltung von Benutzerelementen

Aufgabe 2

_____ / 16 Punkte

Frage

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

Antwort

Option Compare Database
Option Explicit

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

' Alle Deklarationen (mit richtigem Datentyp, Bezeichner,
' Namenskonvention) 1.5P
Dim strErgebnis As String
Dim bytPunkte As Byte
Dim intSumme As Integer

Let intSumme = 0 '0.5P

Do ' 1P

' Let inkl. Zuweisung 0.5P, Val 1P, InputBox 1P
Let bytPunkte = Val(InputBox("Punkte"))
Let intSumme = intSumme + bytPunkte ' 0.5P

Loop While bytPunkte > 0 '1P Loop, 1P While, 0.5P Ausdruck

Debug.Print intSumme ' 0.5P

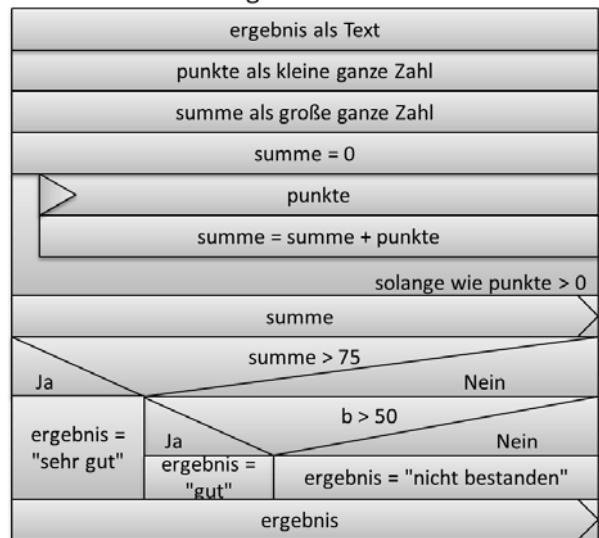
If intSumme > 75 Then 'If+Then 0.5P, Ausdruck 0.5P
Let strErgebnis = "sehr gut" '0.5P
Elseif intSumme > 50 Then 'Elseif+Then 1P, Ausdruck 0.5P
Let strErgebnis = "gut" '0.5P
Else '0.5
Let strErgebnis = "nicht bestanden" '0.5
End If '0.5

Debug.Print strErgebnis '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.

berechneKlausurergebnis



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)	
$600 + 12 / 6 - 300$	302	Integer	/ 1 P
$2.22222222 / 2$	1.11111111	Double	/ 1 P
$19 \setminus 6$	3	Byte	/ 1 P
Not (False) Xor True	False	Boolean	/ 1 P
#1/1/2023# - 1	#12/31/2023# oder "31.12.2023"	Date oder String	/ 1 P

Aufgabe 4

_____ / 31 Punkte

Frage

Für ein Hotel wird ein Programm zur Verwaltung der Zimmer benötigt. Das Hotel hat 5 Etagen und auf jeder Etage 15 Zimmer.

Definieren Sie im gegebenen Modul "mdlHotel" einen zusammengesetzten Datentyp für ein Hotelzimmer mit der Zimmernummer, der Anzahl Betten im Zimmer, dem Zimmerpreis pro Nacht und einem Kennzeichen, ob das Zimmer aktuell belegt ist oder nicht. Der Datentyp soll außerhalb des Moduls sichtbar sein. Beachten Sie, dass es im Hotel keine Zimmer mit der Nr. 13 gibt, weil das 13. Zimmer auf jeder Etage immer die Zimmernummer 12a hat (z.B. 212a oder 412a)

Deklaren Sie auf Ebene des Moduls ein zweidimensionales Feld für das Hotel, in dem sich Hotelzimmer 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 Hotelzimmer in das Feld ein: Das Doppelzimmer Nr. 107 (1. Etage, Nr. 7) mit 2 Betten zum Preis von 120,75 EUR ist aktuell belegt. Das Einzelzimmer 312a (3. Etage, Nr. 13) mit 1 Bett zum Preis von 90 EUR ist aktuell frei.

Geben Sie mit Hilfe von zwei Zählerschleifen alle im Feld gespeicherten Zimmer mit der Zimmernummer im Direktbereich aus. Zählen Sie dabei, wie viele Zimmer belegt 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 THotelzimmer
```

```
' Alle: 1P Bezeichner, 1P Datentyp
strZimmerNr As String ' z.B. 104 oder 212a (weil es kein Zimmer 13 auf der 2. Etage gibt)
bytAnzahlBetten As Byte
curPreisProNacht As Currency
bolBelegt 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 zimHotel(4, 14) As THotelzimmer
```

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

```
' Alle Deklarationen (mit richtigem Datentyp, Bezeichner) 1.5P
Dim bytEtage As Byte
Dim bytZimmer As Byte
Dim bytBelegungen 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. Zimmer
Let zimHotel(0, 6).strZimmerNr = "107"
Let zimHotel(0, 6).bytAnzahlBetten = 2
Let zimHotel(0, 6).curPreisProNacht = 120.75 ' Korrekturhinweis: Auf Dezimalpunkt achten
Let zimHotel(0, 6).bolBelegt = True
```

```
' 3. Etage, 13. Zimmer
With zimHotel(2, 12)
Let .strZimmerNr = "312a"
Let .bytAnzahlBetten = 1
Let .curPreisProNacht = 90
Let .bolBelegt = False
End With
```

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

```
'Für alle For 0.5P, To 0.5P
For bytEtage = 0 To 4 ' Zusätzlich Schleifenvariable 0.5P, Ende 1P
For bytZimmer = 0 To 14 ' 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 zimHotel(bytEtage, bytZimmer).strZimmerNr
```

```
'0.5P If, 0.5P Then, 0.5P Ausdruck
If zimHotel(bytEtage, bytZimmer).bolBelegt = True Then
Let bytBelegungen = bytBelegungen + 1
End If
```

```
Next ' 0.5P Zimmer
Next ' 0.5P Etagen
```

```
' 0.5P MsgBox ggf. inkl. Text, 0.5P Wert
MsgBox "Anzahl belegter Zimmer: " & bytBelegungen
```

```
End Sub ' 0.5P
```

Aufgabe 5

_____ / 15 Punkte

Frage

Gegeben ist die folgende Funktion, die für eine gegebene Tarifzone und eine Anzahl Fahrkarten den Gesamtpreis ermittelt und als Ergebnis zurückliefert.

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 berechGesamtpreis(pstrTarifzone As String, pbytAnzahlFahrkarten As
    Byte) As Currency
2
3  Dim curEinzelpreis As Currency
4
5  Select Case pstrTarifzone
6      Case "A"
7          Let curEinzelpreis = 2.5
8
9          Case "B"
10             Let curEinzelpreis = 3
11
12            Case Else
13                Let curEinzelpreis = 3.5
14
15        End Select
16
17    Let berechGesamtpreis = curEinzelpreis * pbytAnzahlFahrkarten
18
19 End Function
    
```

Antwort

Ausführungspfad (2P)	Parameterwerte als Testdaten (1.5P)/ Begründung (1P)	Ergebnis (0.5P)	Punkte
<p>(3x 2 P)</p> <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>(3x 2.5P)</p> <p>Korrekturhinweis: Werte hier nur exemplarisch; alle anderen Testdaten, die den gleichen Ausführungspfad ergeben bzw. der selben Äquivalenzklasse entsprechen, sind ebenfalls richtig</p>	<p>(3x 0.5P)</p>	

<p>(1),2,3,4,5,(10),11,(12)</p>	<p>pstrTarifzone = pbytAnzahlFahrkarten =</p> <p>Begründung: Warum diese Werte?</p> <p>pstrTarifzone = "A" pbytAnzahlFahrkarten = 2 Zweiter Case wird durchlaufen, Anzahl Fahrkarten egal</p>	<p>5</p>
<p>(1),2,3,(4),6,7,(10),11,(12)</p>	<p>pstrTarifzone = pbytAnzahlFahrkarten =</p> <p>Begründung: Warum diese Werte?</p> <p>pstrTarifzone = "B" pbytAnzahlFahrkarten = 3 Zweiter Case wird durchlaufen, Anzahl Fahrkarten egal</p>	<p>9</p>
<p>(1),2,3,4,6,8,9,(10),11,(12)</p>	<p>pstrTarifzone = pbytAnzahlFahrkarten =</p> <p>Begründung: Warum diese Werte?</p> <p>pstrTarifzone = "B" pbytAnzahlFahrkarten = 1 Zweiter Case wird durchlaufen, Anzahl Fahrkarten egal</p>	<p>3.5</p>

Platz für Notizen

Decorators: Dim str Name as String
 1. Dekoration Konstante: Const pi as Double = 3.14159265359
 2. Private/Public StrName as String
 3. Typumwandlung
 4. Verzweigung
 5. Prozedur
 6. Funktion
 7. Substruktur (Parameter as String)
 8. Substruktur (ich kenne - " & Parameter)
 9. Substruktur (Modul, Fkt & Prozedur)
 10. Substruktur (Modul, Fkt & Prozedur)
 11. Substruktur (Modul, Fkt & Prozedur)
 12. Substruktur (Modul, Fkt & Prozedur)

Prozeduren:
 1. Anzahlgabe / Problem
 2. Anzahlgabe / Problem
 3. Anzahlgabe / Problem
 4. Anzahlgabe / Problem
 5. Test
 6. Anzahlgabe / Problem
 7. Anzahlgabe / Problem
 8. Anzahlgabe / Problem

Datenarten:
 1. Byte
 2. Integer
 3. Long
 4. Single
 5. Double
 6. Currency
 7. Boolean
 8. Variant
 9. Object
 10. Array
 11. String
 12. Enum
 13. Delegate
 14. Interface
 15. Class
 16. Structure
 17. Enum
 18. Delegate
 19. Interface
 20. Class
 21. Structure

Prozeduren:
 1. Anzahlgabe / Problem
 2. Anzahlgabe / Problem
 3. Anzahlgabe / Problem
 4. Anzahlgabe / Problem
 5. Test
 6. Anzahlgabe / Problem
 7. Anzahlgabe / Problem
 8. Anzahlgabe / Problem

Datenarten:
 1. Byte
 2. Integer
 3. Long
 4. Single
 5. Double
 6. Currency
 7. Boolean
 8. Variant
 9. Object
 10. Array
 11. String
 12. Enum
 13. Delegate
 14. Interface
 15. Class
 16. Structure
 17. Enum
 18. Delegate
 19. Interface
 20. Class
 21. Structure

Prozeduren:
 1. Anzahlgabe / Problem
 2. Anzahlgabe / Problem
 3. Anzahlgabe / Problem
 4. Anzahlgabe / Problem
 5. Test
 6. Anzahlgabe / Problem
 7. Anzahlgabe / Problem
 8. Anzahlgabe / Problem

Datenarten:
 1. Byte
 2. Integer
 3. Long
 4. Single
 5. Double
 6. Currency
 7. Boolean
 8. Variant
 9. Object
 10. Array
 11. String
 12. Enum
 13. Delegate
 14. Interface
 15. Class
 16. Structure
 17. Enum
 18. Delegate
 19. Interface
 20. Class
 21. Structure