

Klausur

Anrede	☐ Frau ☐ Herr ☐ _____	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:			
☐ Ja ☐ Nein			

Wird von der Lehrkraft ausgefüllt				
Aufgabe 1	0/ 4	5	Minuten	
Aufgabe 2	0/ 12	10	Minuten	
Aufgabe 3	0/ 5	5	Minuten	
Aufgabe 4	0/ 30	30	Minuten	
Aufgabe 5	0/ 20	30	Minuten	
Aufgabe 5	0/ 9	10	Minuten	
Gesamtpunkte	0/ 80	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	76	72	68	64	60	56	52	48	44	40

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

Bitte wenden.

Aufgabe 1**_____ / 4 Punkte****Frage c)**

Nennen Sie drei Beispiele für Systemsoftware! In welche weiteren Kategorien kann Software unterteilt werden? Nennen Sie mindestens zwei! (1P)

Antwort c)

Beispiele für Systemsoftware sind: iOS, Windows, Android, Mac OSX, Weiterhin gibt es Anwendungssoftware und systemnahe Software.

Frage e)

Sie haben zwei prinzipielle Möglichkeiten kennengelernt, wie eine Verzweigung im Programmablauf programmiert werden kann. Welche Möglichkeiten sind das? Wann wählen Sie die eine und wann die andere Möglichkeit? (1.5P)

Antwort e)

Die zwei prinzipiellen Möglichkeiten zur Umsetzung von Verzweigungen sind:

If-Then-(Else)-Else-End If

Select Case

Die If-Then-Struktur wird verwendet, wenn Bedingungen unterschiedlich oder komplex sind, z. B. Vergleiche wie „größer als“, „kleiner als“ oder Kombinationen mit logischen Operatoren (Und, Oder).

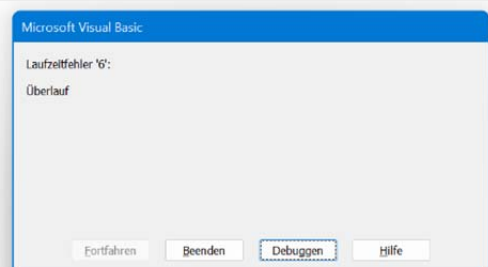
Select Case eignet sich besser bei festen, vorher bekannten Vergleichswerten, etwa konkreten Zahlen oder Zeichen, und wenn viele gleichartige Fälle unterschieden werden sollen. Sie sorgt dann für übersichtlicheren Code.

Frage g)

Bei der Ausführung der Prozedur tanzen() wird ein Fehler angezeigt. Was bedeutet er, bei welcher Anweisung tritt er auf und wie lässt er sich korrigieren? (1.5P)

Antwort g)

```
Private Sub tanzen()  
    Dim curTicketPreis As Currency  
    Dim curGesamt As Currency  
    Dim bytTanzpaare As Byte  
    Dim bytTeilnehmer As Byte  
  
    Let bytTanzpaare = 132  
    Let bytTeilnehmer = bytTanzpaare * 2  
    Let curGesamt = curTicketPreis * bytTeilnehmer  
  
    Debug.Print curGesamt  
End Sub
```



Der Überlauf bedeutet, dass ein Wert zu groß für seinen Datentyp ist. Er tritt auf bei der Zuweisung von bytTeilnehmer, weil $132 \times 2 = 264$ größer als 255 ist. Der Typ Byte kann nur Werte bis 255 speichern. Der Fehler lässt sich beheben, indem man statt Byte einen größeren Datentyp wie Integer verwendet. Dann passt auch der berechnete Wert in den Speicherbereich.

Aufgabe 2

_____ / 12 Punkte

Frage

Gegeben ist das folgende Struktogramm, mit dem ein Unternehmen ermitteln möchte, welche Zahlungsmittel (Bar, Karte, Gutschein) wie oft an der Kasse eingesetzt werden. Schreiben Sie den dargestellten Algorithmus als VBA-Prozedur. Halten Sie dabei auch die Namenskonventionen ein, die in der Lehrveranstaltung vermittelt wurden!

Antwort

Option Compare Database
Option Explicit

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

' Alle Deklarationen: 1.5P
Dim bytEingabe As Byte
Dim intBar As Integer
Dim intKarte As Integer
Dim intGutschein As Integer

' Eingabe (vorprüfend): 0.5P Initialisierung
Let bytEingabe = 99

' Vorprüfende Schleife mit Do Until
' 1P Do, 1P Until, 0.5P Ausdruck
Do Until bytEingabe = 0
 ' Eingabe mit InputBox und Typumwandlung
 ' (z.B. Val): 1P, Zuweisung 0.5P
 Let bytEingabe = Val(InputBox("Zahlungsmethode (1=Bar, 2=Karte, 3=Gutschein, 0=Ende):"))

' 0.5P Select Case mit Zahlungsmethode, 0.5P Alle Case-Ausdrücke,
' 0.5P End Select
Select Case bytEingabe
 Case 1
 Let intBar = intBar + 1 ' Zuweisung 0.5P
 Case 2
 Let intKarte = intKarte + 1 ' Zuweisung 0.5P
 Case 3
 Let intGutschein = intGutschein + 1 ' Zuweisung 0.5P
 Case Else
 MsgBox "Bitte 1, 2, 3 oder 0 für Ende" ' Ausgabe (auch Debug.Print) 0.5P
End Select

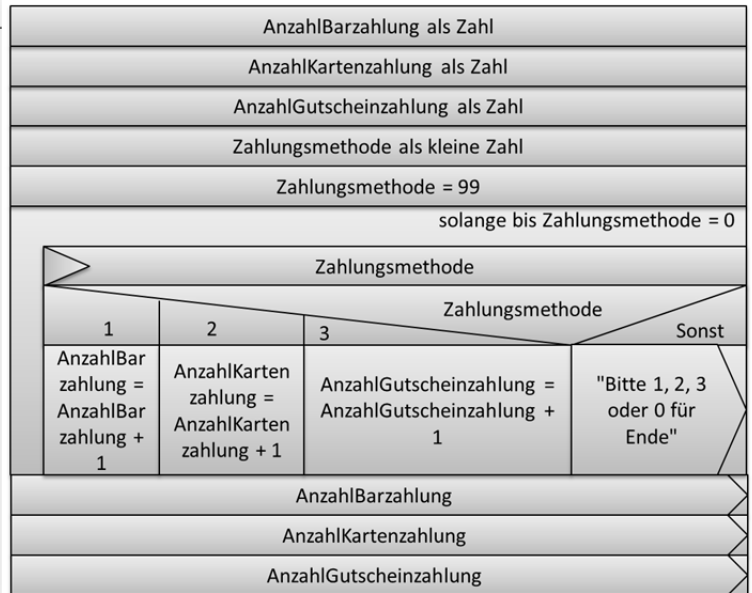
Loop ' 0.5P

' Ausgaben: alle 0.5P (auch einfache nur die Variable ist richtig)
Debug.Print "Barzahlungen: " & intBar
Debug.Print "Kartenzahlungen: " & intKarte
Debug.Print "Gutscheine: " & intGutschein

End Sub ' 0.5P
Korrekturhinweis: Hier darf es keine Abweichungen vom Algorithmus geben. Genau das ist Ziel der Aufgabe, weil er durch das Struktogramm fest vorgegeben. Deshalb gibt es Punktabzug, selbst wenn die einzelnen Anweisungen irgendwo/irgendwie korrekt sind.

Wurden die Namenskonventionen nicht eingehalten: -1P

zähleZahlungsmittel

**Bitte wenden.**

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)	
"Preis:" & 12.5 * 2 & "€"	"Preis:25€"	String	/ 1 P
10 < 20 Xor 5 > 3	False	Boolean	/ 1 P
1.11111111 * 2	2.22222222	Double	/ 1 P
400 + 2 * (10 - 5)	410	Integer	/ 1 P
10000 – 50000	-40000	Long	/ 1 P

Aufgabe 4

_____ / 30 Punkte

Frage

Ein Sportverein möchte die Ergebnisse eines Staffellaufs digital erfassen. In dem Modul **mdlStaffel** soll ein öffentlicher benutzerdefinierter Datentyp für Läufer angelegt werden, der den Namen des Läufers, die gelaufene Zeit in Sekunden (als Zahl mit Kommastelle) sowie ein Kennzeichen enthält, ob der Lauf gültig war oder nicht (z.B. weil der Staffelstab nicht richtig übergeben wurde). Außerdem sollen auf Ebene des Moduls zwei Konstanten definiert werden, die die Anzahl der startenden Mannschaften und die Anzahl der Läufer pro Mannschaft festlegen. Es starten fünf Mannschaften mit jeweils drei Läufern. Die Konstanten sollen außerhalb des Moduls nicht sichtbar sein.

Mit Hilfe dieser Konstanten ist auf Modulebene ein Feld zu deklarieren, in dem die Läufer gespeichert werden. Das Feld soll außerhalb des Moduls sichtbar sein.

Schreiben Sie eine Prozedur **initialisiereFeld**. Sie soll außerhalb des Moduls sichtbar sein. In dieser Prozedur sollen die folgenden zwei Läufer in das Feld eingetragen werden: 1. für die Mannschaft 3, Läufer 1 mit dem Namen „Müller“, der gelaufenen Zeit 12,3 Sekunden und als gültiger Lauf eingetragen. 2. für die Mannschaft 1, Läufer 3, mit dem Namen „Demir“, gelaufene Zeit 9,2 Sekunden und als ungültiger Lauf eingetragen werden.

Schreiben Sie weiterhin eine Prozedur **anzeigenErgebnisse**, die außerhalb des Moduls nicht sichtbar sein soll. Darin sollen zwei geschachtelte Zählerschleifen verwendet werden, um alle Mannschaften und deren Läufer zu durchlaufen. Den Ende-Wert für die Zählerschleifen legen Sie unter Verwendung der zuvor deklarierten Konstanten minus 1 fest. Für jeden Läufer sollen im Direktfenster die Nummer der Mannschaft, die Nummer des Läufers, der Name, die Zeit und die Gültigkeit auszugeben.

Halten Sie die aus der Lehrveranstaltung bekannten Namenskonventionen ein!

Antwort

Das Modul mdlStaffel beginnt hier:

Option Compare Database
Option Explicit

```
' 0.5P Public, 0.5P Type verwendet, 0.5P sinnvoller Bezeichner
' 1.5P Eigenschaften inkl. Bezeichner und passendem Datentyp
' 1P gesamte Typdefinition auf Modulebene
Public Type TLaeufer
    strName As String
    sngZeit As Single
    bolGueltig As Boolean
End Type '0.5P

' Konstanten
' 0.5P Private, 1P Bezeichner (unter Einhaltung der Namenskonvention), 0.5P Datentyp
' 1P Const, 1P Initialisierung mit Wert
' 1P auf Modulebene
' Hinweis: Byte ist hier gewählt, Integer wäre auch korrekt (Kommentarwert)
Private Const ANZAHL_MANNschaften As Byte = 5
Private Const LAEUFER_JE_MANNschaft As Byte = 3

' 0.5P Public, 0.5P Bezeichner, 1P richtiger Datentyp
' 1P Indexgröße korrekt (ANZAHL_MANNschaften-1 und LAEUFER_JE_MANNschaft-1)
' 1P auf Modulebene
Public lafErgebnisse(ANZAHL_MANNschaften - 1, LAEUFER_JE_MANNschaft - 1) As TLaeufer
' oder nicht so gut, aber trotzdem als richtig akzeptiert:
' Private lafErgebnisse(4, 2) As TLaeufer

' 0.5P Public, 0.5P Sub korrekt, 0.5P sinnvoller Bezeichner
Public Sub initialisiereFeld()

    ' Hinweis: Freie Entscheidung, ob mit oder ohne With (gleiche Punktzahl)

    ' Mannschaft 3, Läufer 1 d.h. Index-Position im Feld ist (2, 0)
    ' 0.5P Werte korrekt (alle sechs)
    ' 0.5P Eigenschaften korrekt (alle sechs)
    ' 0.5P Nutzung des Feldes mit Klammern (alle sechs)
    ' 1P Indexposition korrekt (2,0 bei allen drei)
    Let lafErgebnisse(2, 0).strName = "Müller"
    Let lafErgebnisse(2, 0).sngZeit = 12.3
    Let lafErgebnisse(2, 0).bolGueltig = True

    ' Mannschaft 1, Läufer 3 d.h. Index-Position (0, 2)
    ' 1P Indexposition korrekt (0,2 bei allen drei)
    With lafErgebnisse(0, 2)
        Let .strName = "Demir"
        Let .sngZeit = 9.2
        Let .bolGueltig = False
    End With

End Sub '0.5P
```

Bitte wenden.

Aufgabe 5**_____ / 20 Punkte****Frage**

Gegeben ist das folgende Formular frmStaffelErgebnisse, in dem die Ergebnisse einzelner Staffelläufer angezeigt werden können. Kann in den Aufklappliste die Nummer einer Mannschaft und eines Läufers gewählt werden. Die verfügbare Auswahlmöglichkeiten finden Sie in den Screenshots dargestellt.

The diagram shows the form **frmStaffelErgebnisse** with the following controls labeled:

- txtTeams**: Label for the 'Mannschaften' dropdown.
- lblTeams**: Label for the 'Mannschaften' dropdown.
- cmbLaeufer**: Label for the 'Läufer' dropdown.
- lblLaeufer**: Label for the 'Läufer' dropdown.
- btnShow**: Button labeled 'Anzeigen'.
- txtName**: Text box for the runner's name.
- lblName**: Label for the name text box.
- txtZeit**: Text box for the time.
- lblZeit**: Label for the time text box.

Two screenshots show the available options in the dropdown lists:

Mannschaften:

Name	Mannschaften
0	Mannschaft A
1	Mannschaft B
2	Mannschaft C
3	Mannschaft D
4	Mannschaft E

Läufer:

Läufer	Zeit
0	Läufer A
1	Läufer B
2	Läufer C

Schreiben Sie eine Ereignisprozedur, die ausgeführt wird, wenn der Benutzer auf die Schaltfläche „Anzeigen“ klickt.

Rufen Sie im Modul **mdlStaffel** aus Aufgabe 4 die Prozedur **initialisiereFeld** auf.

Deklarieren Sie Variablen mit geeigneten Datentypen, um die Werte der ausgewählten Mannschaftsnummer und der Läufernummer entgegenzunehmen. Ermitteln Sie dann die in den Aufklapplisten ausgewählten Werte und speichern Sie diese in den deklarierten Variablen. Nutzen Sie die beiden Werte anschließend, um im Feld (aus Aufgabenteil 4) für den entsprechenden Läufer der Mannschaft den Namen, die Zeit und das Kennzeichen ob der Lauf gültig war oder nicht zu ermitteln. Speichern Sie diese Angaben in Variablen, die sich zuvor mit passenden Datentypen deklariert haben.

Zeigen Sie Name und Zeit in den entsprechenden Textfeldern an. Falls der Lauf nicht gültig ist, geben Sie ein Meldungsfenster mit dem Text "Ungültiger Lauf" aus.

Antwort

Option Compare Database
Option Explicit

' 0.5P Private, 0.5P Sub, 1P Bezeichner, 0.5P Ereignis

Private Sub btnShow_Click()

' Für alle Dim 0.5P, Bezeichner 0.5P, As 0.5P, Datentyp 1P

Dim bytMannschaft As Byte

Dim bytLaeufer As Byte

Dim strName As String

Dim sngZeit As Single

Dim bolGueltig As Boolean

' Für alle Let 1P insgesamt

' Bezeichner 0.5P, 0.5P Zuweisung, 1P Zugriff auf Oberfläche

' Hinweis: Keine Typumwandlung (z.B. Val()) erforderlich

Let bytMannschaft = Me.cmbTeam.Value

Let bytLaeufer = Me.cmbLaeufer.Value

' 0.5 Call, 1P Modul und Punktnotation, 0.5P Prozedur

Call mdlStaffellauf.initialisiereFeld

' 1P Zugriff auf Feld, 0.5P Bezeichner, 0.5P Klammern,

' 1P Indexposition korrekt mit Variablen, Zugriff über Modul und Feld 2P

Let strName = mdlStaffellauf.lafErgebnisse(bytMannschaft, bytLaeufer).strName

Let sngZeit = mdlStaffellauf.lafErgebnisse(bytMannschaft, bytLaeufer).sngZeit

Let bolGueltig = mdlStaffellauf.lafErgebnisse(bytMannschaft, bytLaeufer).bolGueltig

' 0.5P Bezeichner, 0.5P Zuweisung, 1P Zugriff auf Oberfläche

Let Me.txtName.Value = strName

Let Me.txtZeit.Value = sngZeit

' 1P If Then korrekt, 0.5P Vergleich oder Not-Operator

If bolGueltig = False Then

' Alternativ: If Not(Gueltig) Then

 MsgBox "Der Lauf war ungültig!" '0.5P

End If '0.5P

End Sub '0.5P

Aufgabe 6

_____ / 9 Punkte

Frage

Gegeben ist die folgende Funktion, die immer eine Zahl vom Typ Integer als Parameter übergeben bekommt und das Produkt aller Vorgänger der Zahl 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 berechneProdukt(bytZahl As Byte) As Long
2     Dim i As Integer
3     Dim lngProdukt As Long
4     Let lngProdukt = 1
5     Let i = 1
6     Do
7         Let lngProdukt = lngProdukt * i
8         Let i = i + 1
9     Loop Until i > bytZahl
10    Let berechneProdukt = lngProdukt
11 End Function

```

Antwort

Ausführungspfad (1P) (3x 1 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.	Parameterwerte als Testdaten (0.5P)/ Begründung (1P) (3x 1.5P) Korrekturhinweis: Werte hier nur exemplarisch; alle anderen Testdaten, die den gleichen Ausführungspfad ergeben bzw. der selben Äquivalenzklasse entsprechen, sind ebenfalls richtig	Ergebnis (0.5P) (3x 0.5P)	Punkte
1, 2, 3, 5, 6, 10, 11	Wert: intZahl = 0 Begründung: Warum dieser Wert? Schleife wird nicht betreten, da Startwert > Endwert nicht erfüllt	1	

1, 2, 3, 5, 6, 7, 8, 6, 10, 11	Wert: intZahl = 1 Begründung: Warum dieser Wert? Schleife wird genau einmal durchlaufen	1
1, 2, 3, 5, 6, 7, 8, 6, 7, 8, 6, 10, 11	Wert: intZahl = 2 Begründung: Warum dieser Wert? Schleife wird mehrfach durchlaufen (hier: 2×)	2 (= 1 x 1 x 2)