

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/ 4 | 5 | Minuten |
| Aufgabe 2 | 0/ 13 | 15 | Minuten |
| Aufgabe 3 | 0/ 5 | 5 | Minuten |
| Aufgabe 4 | 0/ 20 | 30 | Minuten |
| Aufgabe 5 | 0/ 20 | 25 | Minuten |
| Aufgabe 6 | 0/ 8 | 10 | 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

_____ / 4 Punkte

Frage a)

Welchen Zwecken dient der Debugger? Nennen Sie mindestens zwei! (1P)

Antwort a)

Ein Debugger dient dem schrittweisen Ausführen und Überprüfen von Programmen, um Fehler gezielt zu finden. Er zeigt dabei den aktuellen Wert von Variablen, erlaubt das Setzen von Haltepunkten (Breakpoints) und das Nachvollziehen des Programmablaufs. So kann nachvollzogen werden, an welcher Stelle ein unerwartetes Verhalten auftritt.

Frage b)

Was ist der Hauptunterschied zwischen einer vor- und einer nachprüfenden Schleife? (1.5P)

Antwort b)

Der Hauptunterschied liegt im Zeitpunkt der Bedingungsprüfung: Bei einer vorprüfenden Schleife (Do While ... Loop) wird die Bedingung vor dem ersten Schleifendurchlauf geprüft – sie kann also auch gar nicht ausgeführt werden. Bei einer nachprüfenden Schleife (Do ... Loop While) erfolgt die Prüfung erst nach dem ersten Durchlauf – die Schleife wird daher mindestens einmal ausgeführt.

Frage c)

Der Benutzer erwartet die Ausgabe von 4,5 im Direktbereich. Warum wird von der Prozedur das falsche Ergebnis ausgegeben? Wie lässt es sich korrigieren? (1.5P)

Antwort c)

Die Prozedur gibt fälschlich den Wert 4 statt 4,5 aus, weil der Ausdruck $9 / 2$ zwar das Ergebnis 4,5 liefert, dieses in eine Long-Variable gespeichert wird. Beim Zuweisen wird die Nachkommastelle abgeschnitten. Um das zu korrigieren, muss lngZahl durch einen Datentyp mit Nachkommastellen wie Single ersetzt werden.

Option Compare Database

```
Private Sub division()  
    Dim lngZahl As Long  
    Let lngZahl = 9 / 2  
    Debug.Print lngZahl  
End Sub
```

Direktbereich
4

Aufgabe 2

_____ / 13 Punkte

Frage

Gegeben ist das folgende Struktogramm, das das Maximum aus drei Zahlen bildet. 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 bestimmeMaximum()

' Alle Deklarationen: 1.5P
Dim intZahl1 As Integer
Dim intZahl2 As Integer
Dim intZahl3 As Integer
Dim intMaximum As Integer

' Beginn der Schleife
Do ' 0.5P

' Eingabe mit InputBox und Typumwandlung (z.B. Val):
' 1P, Zuweisung 0.5P
Let intZahl1 = Val(InputBox("Bitte erste Zahl eingeben:"))
Let intZahl2 = Val(InputBox("Bitte zweite Zahl eingeben:"))
Let intZahl3 = Val(InputBox("Bitte dritte Zahl eingeben:"))

' Vergleich erste vs. zweite Zahl
If intZahl1 > intZahl2 Then ' 0.5P If+Then, 0.5P Ausdruck
 Let intMaximum = intZahl1 ' 0.5P
Else
 Let intMaximum = intZahl2 ' 0.5P
End If ' 0.5P

' Vergleich dritte Zahl mit aktuellem Maximum
If intZahl3 > intMaximum Then ' 0.5P If+Then, 0.5P Ausdruck
 Let intMaximum = intZahl3 ' 0.5P
End If ' 0.5P

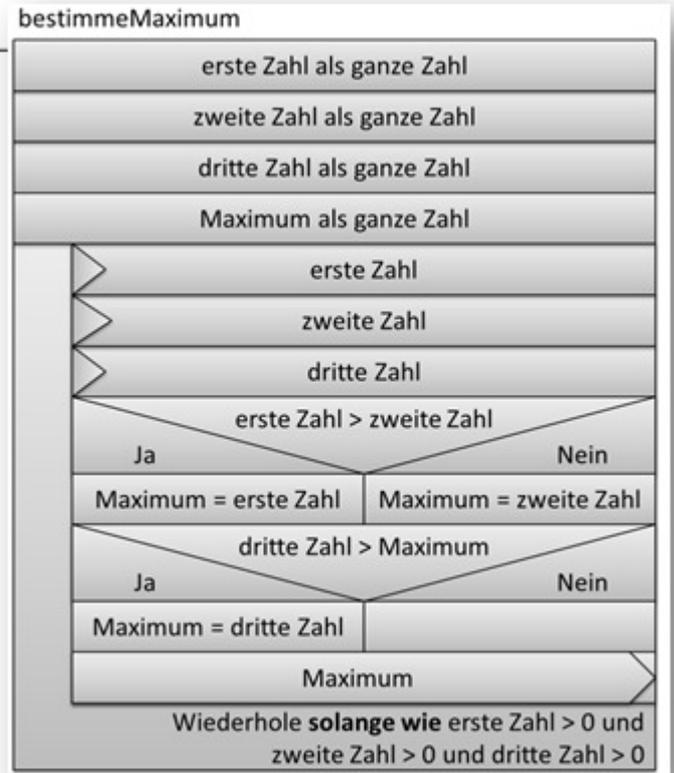
' Ausgabe: 0.5P
Debug.Print "Das Maximum ist: " & intMaximum

' 1P Loop, 1P While, 1P Ausdruck mit AND
Loop While intZahl1 > 0 And intZahl2 > 0 And intZahl3 > 0

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 irgendwie/irgendwo korrekt sind.

Wurden die Namenskonventionen nicht eingehalten: -1P



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) | |
|-----------------------|--------------------|-------------------------------|-------|
| $2.222 * -1$ | -2,222 | Single | / 1 P |
| "10" & 5 * 2 | "1010" | String | / 1 P |
| $20 \setminus 6$ | 3 | Byte | / 1 P |
| $-5 * 3 + 2 \wedge 2$ | -11 | Integer | / 1 P |
| $2 * 4 = 8$ | True | Boolean | / 1 P |

Aufgabe 4

_____ / **20 Punkte**

Frage

Ein Unternehmen möchte die täglich eingehenden Support-Anfragen besser organisieren. Stellen Sie dazu in dem Modul mdlSupport einen selbst definierten Datentyp für ein Support-Ticket bereit, der folgende Elemente umfasst: eine Ticketnummer als ganze Zahl, einen Text zur Beschreibung der Support-Meldung, ein Kennzeichen, ob das Ticket bereits erledigt ist oder nicht, und das Eingangsdatum, an dem das Ticket eingegangen ist.

Deklariert Sie dann ein Feld, in dem Sie maximal 100 Support-Tickets speichern können. Der Gültigkeitsbereich des Feldes soll das ganze Modul umfassen. Das Feld soll außerhalb des Moduls nicht sichtbar sein.

Deklariert Sie weiterhin eine Variable, deren Gültigkeitsbereich ebenfalls das Modul ist und die außerhalb des Moduls sichtbar ist. In dieser Variablen soll die aktuell vergebene Ticketnummer gespeichert werden.

Schreiben Sie eine öffentliche Funktion anlegenTicket, die drei Parameter entgegennimmt: den Text der Support-Meldung, das Eingangsdatum und das Kennzeichen, ob das Support-Ticket erledigt ist oder nicht. Als Ergebnis soll die Funktion ein Support-Ticket zurückliefern.

Deklariert Sie dazu in der Funktion ein neues Support-Ticket. Initialisieren Sie das neu deklarierte Ticket mit dem Wert der Variable, die die aktuell vergebene Ticketnummer speichert, dem als Parameter übergebenen Text, dem Eingangsdatum und dem Kennzeichen, ob das Ticket erledigt ist oder nicht.

Weisen Sie das neu angelegte Ticket im Feld an der Stelle mit seiner Ticketnummer zu. Erhöhen Sie anschließend den Wert der Variablen für die aktuell vergebene Ticketnummer um 1. Geben Sie das Ticket außerdem als Ergebnis der Funktion zurück.

Antwort

Das Modul mdlSupport beginnt hier:

```
Option Compare Database
Option Explicit

' 0.5P Type verwendet, 0.5P sinnvoller Bezeichner
' 1P Eigenschaften inkl. Bezeichner und passendem Datentyp
Type TSupportTicket
    intNr As Integer
    strText As String
    bolErledigt As Boolean
    datEingang As Date
End Type
' 0.5P End Type korrekt
' 1P gesamte Typdefinition steht auf Modulebene

' 0.5P Public
' 0.5P sinnvoller Bezeichner, 0.5P Datentyp
' 1P steht auf Modulebene
Public intLetzteNummer As Integer

' 0.5P Private
' 0.5P Bezeichner korrekt
' 1P Indexgröße korrekt (0-99)
' 1P richtiger Datentyp
' 1P steht auf Modulebene
Private tckTickets(99) As TSupportTicket

' 0.5P Public Function korrekt
' 1.5P Parameter korrekt deklariert
' 1P Rückgabewert
Public Function anlegenTicket(pstrText As String, pdatEingang As Date, pbolErledigt As Boolean) As TSupportTicket

    ' 1P lokale Variable vom Typ TSupportTicket korrekt deklariert
    Dim tckNeu As TSupportTicket

    ' 1P Initialisierung aller vier Felder im With-Block oder ohne With
    With tckNeu
        Let .intNr = intLetzteNummer
        Let .strText = pstrText
        Let .datEingang = pdatEingang
        Let .bolErledigt = pbolErledigt
    End With

    ' 1P Zugriff auf Arrayelement korrekt
    ' 0.5P Let-Zuweisung korrekt
    Let tckTickets(intLetzteNummer) = tckNeu

    ' 0.5P korrektes Inkrement mit Let
    Let intLetzteNummer = intLetzteNummer + 1

    ' 1P Rückgabe des Ergebnisses mit Let korrekt
    Let anlegenTicket = tckNeu
End Function

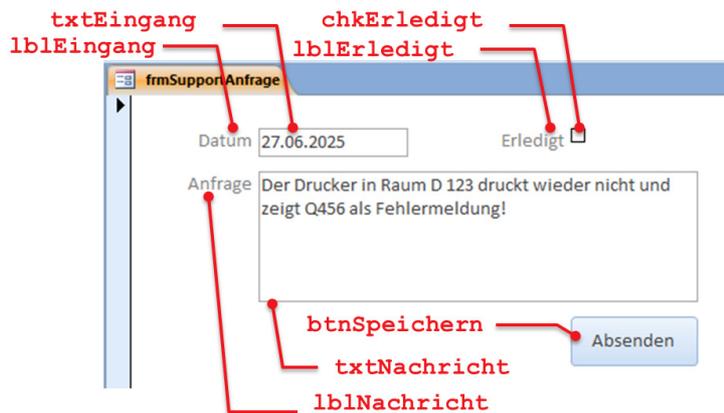
'2P Beschriebener Algorithmus klar umgesetzt
```

Aufgabe 5

_____ / 20 Punkte

Frage

Gegeben ist das folgende Formular "frmSupportAnfrage", in dem eine neue Anfrage an den Support erfasst werden kann, mit einem Eingangsdatum, einem Nachrichtentext und einem Kennzeichen, ob das Problem schon erledigt ist oder nicht. Die Namen der Steuerelemente sind in der Abbildung angegeben.



Gehen Sie im Folgenden davon aus, dass der Benutzer in allen Feldern gültige Werte einträgt oder auswählt. Schreiben Sie eine Ereignisprozedur, die aufgerufen wird, wenn der Benutzer auf die Schaltfläche "Absenden" klickt.

Deklarieren Sie Variablen mit geeigneten Datentypen, um entsprechende Eingaben verarbeiten zu können. Ermitteln Sie dann die in den Oberflächenelementen eingegebenen oder ausgewählten Werte und wandeln Sie diese bei Bedarf mit einer Typumwandlungsfunktion um, um sie in den zuvor deklarierten Variablen zu speichern.

Rufen Sie dann die Funktion anlegenTicket im Modul mdlSupport (aus Aufgabenteil a) auf und übergeben Sie die Variablenwerte als Parameter. Nehmen Sie das neu angelegte Ticket als Rückgabewert der Funktion entgegen und speichern Sie es in einer weiteren deklarierten Variablen.

Antwort

```
Option Compare Database
Option Explicit

' 0.5P Private, 0.5P Sub, 1P Bezeichner, 0.5P Ereignis
Private Sub btnSpeichern_Click()

' Für alle Dim 0.5P, Bezeichner 0.5P, As + Datentyp 1P
Dim strText As String
Dim datEingang As Date
Dim bolErledigt As Boolean

Dim tckNeu As TSupportTicket ' Zusätzlich 0.5P für Datentyp

' Für alle Let 1P, Bezeichner 0.5P, 1P Zuweisung,
' 2P Zugriff auf Oberfläche, 1P Typumwandlung (z.B. CBool, CDate)
Let strText = Me.txtNachricht.Value ' Kein Val()! 0.5P
Let datEingang = CDate(Me.txtEingang.Value)
Let bolErledigt = CBool(Me.chkErledigt.Value)

' 1P Zuweisung mit Let, 0.5P Modul, 0.5P Funktionsname, 0.5P Klammern, 1.5P Parameter korrekt übergeben
Let tckNeu = mdlSupport.anlegenTicket(strText, datEingang, bolErledigt)

End Sub ' 0.5P

Bonuspunkte für klar erkennbare Algorithmus und Einhaltung der Namenskonventionen: 4,5P
```

Aufgabe 6

_____ / **8 Punkte**

Eine Bank bietet ein Jugendkonto an, das nur eröffnet werden kann, wenn die kontoführende Person mindestens 18 Jahre und maximal 25 Jahre alt ist. Auf der Benutzeroberfläche zur Kontoeröffnung befindet sich ein Feld zur Eingabe des Geburtsdatums. Nach der Eingabe überprüft eine Funktion das eingegebene Datum.

Gehen Sie davon aus, dass in dem Eingabefeld immer ein gültiges Datum eingetragen ist (d. h. das Feld ist nicht leer und enthält immer ein korrektes Datum im korrekten Format).

Frage

Welche vier Testfälle müssen Sie für einen Black Box-Test mindestens durchführen? Leiten Sie geeignete Äquivalenzklassen für die Eingabe des Geburtsdatums ab. Nennen Sie für jede Klasse jeweils die konkret einzugebenden Testdaten (Datum im Format TT.MM.JJJJ) und begründen Sie Ihre Auswahl. Geben Sie auch an, was das erwartete Ergebnis ist. Ziel ist eine möglichst vollständige Testabdeckung mit wenigen, gezielten Testfällen.

Antwort

| Äquivalenzklassen (0.5P) (4x 0.5P) | Testdaten und Begründung (1P) (4 x 1P) | Ergebnis (0.5P) (4x 0.5P) | Punkte |
|---|--|------------------------------|--------|
| Alter zwischen mindestens 18 und höchstens 25 | Wert: 01.03.2005 Begründung: Warum dieser Wert? Person ist 20 Jahre alt zum Zeitpunkt der Eröffnung | Kontoeröffnung möglich | |
| Alter ist unter 18 | Wert: 30.05.2020 Begründung: Warum dieser Wert? Person ist 5 Jahre alt zum Zeitpunkt der Eröffnung, was nicht möglich sein soll | Kontoeröffnung nicht möglich | |
| Alter ist größer 25 | Wert: 31.01.1990 Begründung: Warum dieser Wert? Person ist 35 Jahre alt zum Zeitpunkt der Eröffnung | Kontoeröffnung nicht möglich | |

Bitte wenden.

| | | |
|--|---|-------------------------------------|
| Geburtsdatum liegt in der Zukunft | Wert: 30.08.2035 Begründung: Warum dieser Wert? Geburtsdatum, das in der Zukunft liegt, ist zwar ein korrektes Datum, darf aber trotzdem nicht auftreten | Kontoeröffnung nicht möglich |
|--|---|-------------------------------------|

Weitere Testfälle: Person hat heute Geburtstag und ist genau 18 Jahre bzw. genau 25 Jahre alt.

Platz für Notizen und Lösungen