

<input type="checkbox"/> Gr. 1, Dr. D. Auer	Name <u>Angelos Angelis</u>	Aufwand in h <u>9</u>
<input checked="" type="checkbox"/> Gr. 2, Dr. G. Kronberger		
<input type="checkbox"/> Gr. 3, Dr. S. Wagner	Punkte _____	Kurzzeichen Tutor / Übungsleiter _____ / _____

1. Funktionen zur Zeichenkettenverarbeitung (1 + 2 + 4 Punkte)

Entwickeln Sie weitere Operationen zur Zeichenkettenbearbeitung (auf dem Standard-Datentyp *STRING*). Verwenden Sie dafür insbesondere die bereits vorhandenen Pascal-Standardfunktionen.

- Implementieren Sie eine Funktion *Reversed*, die als Ergebnis eine Zeichenkette liefert, welche die Zeichen des Eingangsparameters *s* in umgekehrter Reihenfolge enthält.
- Implementieren Sie eine Prozedur *StripBlanks*, die alle Leerzeichen (engl. *blanks*) aus der als Übergangsparameter *s* übergebenen Zeichenkette entfernt.
- Implementieren Sie eine Prozedur *ReplaceAll*, die alle Vorkommen der Zeichenkette *old* (Eingangsparameter) in der Zeichenkette *s* (Übergangsparameter) durch die Zeichenkette *new* (ebenfalls Eingangsparameter) ersetzt und das Ergebnis im Übergangsparameter *s* liefert. Ist *old* nicht in *s* enthalten, muss *s* unverändert bleiben. Aufruf mit: *ReplaceAll(s, old, new)*.

2. Plausibilitätsprüfung von Messwerten (4 + 3*1 + 3*1 Punkte)

Bei einem Hochofen werden in unregelmäßigen Zeitabständen (alle paar Minuten) Temperaturen gemessen. Dabei wird der Zeitpunkt der Messung (in Stunden und Minuten) sowie die Temperatur (in °Celsius auf 0,1° genau) ermittelt.

- Entwickeln Sie einen Algorithmus mit Gedächtnis (in Pseudocode), der überprüft, ob ein solcher Temperaturmesswert *plausibel* ist: Ein Temperaturmesswert ist genau dann plausibel, wenn die Minimaltemperatur von 935,5° Celsius nicht unter-, die Maximaltemperatur von 1345,6° Celsius nicht überschritten wird und die Veränderung des Messwerts seit der letzten Messung +/- 11,5° Celsius pro Minute nicht übersteigt.
- Geben Sie drei unterschiedliche Realisierungsmöglichkeiten für Ihren Algorithmus mit Gedächtnis in Pascal an.
- Diskutieren Sie die Vor- und Nachteile dieser drei Realisierungen.

3. Programmierstil (2 + 2 + 3 Punkte)

In der Übung 2 haben Sie sich mit der Ausgabe von X-Mustern beschäftigt.

Im Licht Ihrer bisherigen (hoffentlich zumindest zum Teil neuen) Erkenntnisse zum Programmieren, zu Pascal und zum Thema Programmierstil:

- Geben Sie Ihre Lösung (also die Lösungsidee und das Pascal-Programm aus der zweiten Übung) völlig unverändert noch einmal an.
- Diskutieren Sie Ihre Lösung möglichst selbstkritisch. Vergleichen Sie Ihre Anmerkungen auch mit jenen, die Sie von der Tutorin / vom Tutor bekommen haben.
- Versuchen Sie nun, es besser zu machen: Geben Sie eine möglichst "gute" Lösungsidee und ein möglichst "gutes" (im Sinne des Programmierstils) Pascal-Programm an.

Hinweise:

1. Geben Sie für alle Ihre Lösungen immer eine „Lösungsidee“ an.
2. Dokumentieren und kommentieren Sie Ihre Algorithmen.
3. Bei Programmen: Geben Sie immer auch Testfälle ab, an denen man erkennen kann, dass Ihr Programm funktioniert, und dass es auch in Fehlersituation entsprechend reagiert.

Lösungsidee:

- a) Eine Funktion Reversed soll erstellt werden die einen Sting zurückgibt. Hierbei wird ein String(Eingabeparameter) in umgekehrter Reihenfolge in dem Übergabestring gespeichert. Das macht man, indem man mithilfe ein Zählschleife char für char den Eingabestring in umgekehrter Reihenfolge in dem Übergabestring speichert.
- b) Eine Prozedur StripBlanks soll programmiert werden die die Leerzeichen des Eingabestrings löscht. Hierbei wird mithilfe einer Zählschleife der String Char für Char durchgegangen und überprüft, ob ein Leerzeichen vorhanden ist. Falls ja wird das Leerzeichen mithilfe der Standartfunktion Delete gelöscht.
- c) Eine Prozedur ReplaceAll soll programmiert werden die Einen vom User eingegebenen String in einem anderen ebenfalls vom User eingegebenen String sucht und durch einen neuen ebenfalls vom User eingegebenen String ersetzt. Das habe ich mithilfe einer While Schleife programmiert. Solange der zum ersetzende String nicht mehr gefunden wird soll der String der ersetzt werden soll gesucht werden mithilfe der Standartfunktion Pos() dann soll der gelöscht werden mit der Standartfunktion Delete() und dann mit der Standartfunktion Insert ersetzt werden.

Quellltext:

PROGRAM Zeichenkettenverarbeitung;

```
FUNCTION Reversed(s : STRING) :STRING;
var
  i: Integer;
BEGIN
  for i := 1 to Length(s) do
    Reversed[i] := s [Length(s) - i +1];
  END;
```

```
PROCEDURE StripBlanks(VAR s : STRING);
VAR
  blank : CHAR;
  i : INTEGER;
BEGIN
  blank := CHAR(32);
  FOR i := 1 TO Length(s) DO BEGIN
    IF s[i] = blank THEN BEGIN
      Delete(s,i,1);
      Dec(i);
    END;
  END;
END;
```

```
PROCEDURE ReplaceAll(old,new : STRING; VAR s : STRING);
VAR
```

```
i,PosOld : INTEGER;
BEGIN
    WHILE Pos(old,s) <> 0 DO BEGIN
        PosOld := Pos(old,s);
        Delete(s,PosOld,Length(old));
        Insert(new,s,PosOld);
    END;
END;

VAR
s,saveS,old,new : STRING;
i : INTEGER;
BEGIN
    WriteLn('Welcher String soll verarbeitet werden?');
    ReadLn(s);
    saveS := s;
    WriteLn(Reversed(s));

    WriteLn;

    StripBlanks(s);
    WriteLn(s);

    WriteLn;

    s := saveS;
    WriteLn(' Bitte String eingeben das ersetzt werden soll');
    ReadLn(old);
    WriteLn(' Bitte String eingeben mit dem das Wort ersetzt werden soll');
    ReadLn(new);
    ReplaceAll(old,new,s);
    WriteLn(s);

END.
```

Testfälle:

1)

Welcher String soll verarbeitet werden?

Hallo mein name ist Angelos.

.solegnA tsi eman niem ollaH

HallomeinnameistAngelos.

Bitte String eingeben das ersetzt werden soll

Angelos

Bitte String eingeben mit dem das Wort ersetzt werden soll

Peter

Hallo mein name ist Peter.



2)

Welcher String soll verarbeitet werden?

Das sind zu viele Leerzeichen

nehciezreel eleiv uz dnis saD

DassindzuvieleLeerzeichen

Bitte String eingeben das ersetzt werden soll

viele

Bitte String eingeben mit dem das Wort ersetzt werden soll

wenige

Das sind zu wenige Leerzeichen



3)

Welcher String soll verarbeitet werden?

Hier sind einpaar Sonderzeichen :!"§\$%&/()={[]}?>

?}][{=)(/&%\$"! : nehciezrednoS raapnie dnis reiH

HiersindeinpaarSonderzeichen:!"\$%&/()={[]}?>

Bitte String eingeben das ersetzt werden soll

:!"§\$%&/()={[]}?>

Bitte String eingeben mit dem das Wort ersetzt werden soll

Keine Sonder-Zeichen

Hier sind einpaar Sonderzeichen Keine Sonder-Zeichen

2)

Lösungsidee:

a) Ein Algorithmus mit Gedächtnis soll programmiert werden der Temperatur-Messwerte überprüft. Meiner Meinung nach dem besten Weg um einen Algorithmus mit Gedächtnis zu programmieren ist es mit Hilfe von Units. In der Unit wird mit Hilfe einer Funktion Checktemps die die Messwerte vom Ofen einliest und einen Boolean ausgibt, geprüft ob die Messwerte gültig sind(True falls die Messwerte „stimmen“ und False falls nicht). Was sich der Algorithmus merken soll ist den vorherigen Messwert damit man den dann mit dem neuen Messwert vergleichen kann.

b) Die drei Realisierungsmöglichkeiten für einen Algorithmus mit Gedächtnis wären:

- 1) Units
- 2) Globale Variablen
- 3) Statische Variablen

Bei der Implementierung habe ich ja kein Ofen der mir die Messwerte gibt deshalb muss der User die Messwerte selber eingeben. Ebenfalls falls ein nicht gültiger Messwert erkannt wird, wird er sozusagen übersprungen und vom Programm nicht gemerkt.

c) Vor- und Nachteile von diesen Realisierungsmöglichkeiten:

1) Units:

Vorteile:

- Ist am einfachsten zu lesen (auch verglichen zu den anderen Möglichkeiten)
- Ist sehr sicher in dem Sinne, dass die Werte die gemerkt werden sollen nur in der Unit selber geändert werden können und von außen sozusagen nicht zugreifbar sind.

Nachteile:

- Ist meiner Meinung nach Aufwendiger zu programmieren als die anderen Möglichkeiten

2) Globale Variablen:

Vorteile:

- Ist am einfachsten zu programmieren

Nachteile:

- Sehr unlesbar
- Es kann vorkommen, dass der Wert der gemerkt werden soll ungewollt geändert wird da er von jeder Funktion im Algorithmus lesbar und bearbeitbar ist.

3) Statische Variablen:

Vorteile:

- Ist einfach zu programmieren
- Ist auch sicherer, weil der Wert der gemerkt werden soll nur innerhalb Funktion in der er deklariert wurde geändert werden kann

Nachteile:

- Ist nicht so lesbar wie Units
- Meiner Meinung nach ist die Art wie man statische Variablen in Pascal erstellt nicht so gut und lesbar wie zB in C.

Pseudocode:

```
MAIN
(*save state using Unit*)
USES UnitTempPruef;

VAR
tempsMeasurement : measurement;
BEGIN
    WHILE OvenIsOn DO BEGIN
        ReadLn(Hour,Minute,Temps);
        WriteLn(CheckTemps(Hour,Minute,Temps));
    END;
END.
-----
UNIT UnitTempPruef;

VAR
prevTemp : REAL;
prevTime,currentTime,difTime : INTEGER;

FUNCTION CheckTemps(hour,minute : INTEGER; temp : REAL) : BOOLEAN;
BEGIN
    IF prevTime = 0 THEN prevTime := 1;
    currentTime := minute + (hour*60);
    IF prevTemp = 0 THEN prevTemp := currentTime;
    difTime := currentTime - prevTime;
    IF (temp >= 935.5) AND (temp <=1345.6) AND (prevTemp - temp >= -11.5*difTime)
AND (prevTemp - temp <= 11.5*difTime)THEN BEGIN
        CheckTemps := TRUE;
        prevTemp := temp;
        prevTime := currentTime;
    END
    ELSE BEGIN
        CheckTemps := FALSE;
    END
END;
```

Quellcode:

Variant emit Unit:

```
PROGRAM TempPruefung;
(*save state using Unit*)
USES UnitTempPruef;

TYPE
measurement = RECORD
Hour : INTEGER;
Minute : INTEGER;
Temps : REAL;
END;

VAR
tempsMeasurement : measurement;
ovenState : STRING;
BEGIN
  WriteLn('Soll die Temperatur weiter gemessen werden? Bitte mit ja / nein Antworten');
  ReadLn(ovenState);
  WHILE ovenState = 'ja' DO BEGIN

    WriteLn('Bitte Stunden, Minuten , Temperatur der Messung eingeben');
    ReadLn(tempsMeasurement.Hour,tempsMeasurement.Minute,tempsMeasurement.Temps);

    WriteLn(CheckTemps(tempsMeasurement.Hour, tempsMeasurement.Minute, tempsMeasurement.Temps));

    WriteLn('Soll die Temperatur weiter gemessen werden? Bitte mit ja / nein Antworten');
    ReadLn(ovenState);
  END;
  WriteLn('Ok das Messen hört auf');
END.
UNIT UnitTempPruef;

INTERFACE

FUNCTION CheckTemps(hour,minute : INTEGER;temp : REAL) : BOOLEAN;
```

IMPLEMENTATION

VAR

```
prevTemp : REAL;  
prevTime,currentTime,difTime : INTEGER;
```

```
FUNCTION CheckTemps(hour,minute : INTEGER; temp : REAL) : BOOLEAN;  
BEGIN
```

```
  IF prevTime = 0 THEN prevTime := 1;  
  currentTime := minute + (hour*60);  
  IF prevTemp = 0 THEN prevTemp := currentTime;  
  difTime := currentTime - prevTime;  
  IF (temp >= 935.5) AND (temp <=1345.6) AND (prevTemp - temp >= -  
11.5*difTime) AND (prevTemp - temp <= 11.5*difTime) THEN BEGIN  
    CheckTemps := TRUE;  
    prevTemp := temp;  
    prevTime := currentTime;  
  END  
  ELSE BEGIN  
    CheckTemps := FALSE;  
  END  
END;
```

BEGIN

```
WriteLn('UnitTempPruef Loaded');  
END.
```

Variante mit globaler Variable:

```
PROGRAM TempPruefung;
```

TYPE

```
measurement = RECORD
```

```
Hour : INTEGER;  
Minute : INTEGER;  
Temps : REAL;  
END;
```

```
(*Save state using global Variables*)
```

VAR

```
prevTemp : REAL;  
prevTime,currentTime,difTime : INTEGER;
```

```
FUNCTION CheckTemps(hour,minute : INTEGER; temp : REAL) : BOOLEAN;  
BEGIN
```

```
  IF prevTime = 0 THEN prevTime := 1;  
  currentTime := minute + (hour*60);
```

```

    IF prevTemp = 0 THEN prevTemp := currentTime;
    difTime := currentTime - prevTime;
    IF (temp >= 935.5) AND (temp <= 1345.6) AND (prevTemp - temp >= -
11.5*difTime) AND (prevTemp - temp <= 11.5*difTime) THEN BEGIN
        CheckTemps := TRUE;
        prevTemp := temp;
        prevTime := currentTime;
    END
    ELSE BEGIN
        CheckTemps := FALSE;
    END
END;

VAR
tempsMeasurement : measurement;
ovenState : STRING;
BEGIN
    WriteLn('Soll die Temperatur weiter gemessen werden? Bitte mit ja / nein Antwor
ten');
    ReadLn(ovenState);
    WHILE ovenState = 'ja' DO BEGIN

        WriteLn('Bitte Stunden, Minuten , Temperauter der Messung eingeben');
        ReadLn(tempsMeasurement.Hour, tempsMeasurement.Minute, tempsMeasurement.Temps);

        WriteLn(CheckTemps(tempsMeasurement.Hour, tempsMeasurement.Minute, tempsMeasu
rement.Temps));

        WriteLn('Soll die Temperatur weiter gemessen werden? Bitte mit ja / nein Antw
orten');
        ReadLn(ovenState);
    END;
    WriteLn('Ok das Messen hoert auf');
END.

```

Variante mit Statischer Variable:

```
PROGRAM TempPruefung;
```

```
TYPE
```

```
measurement = RECORD
```

```
Hour : INTEGER;
```

```
Minute : INTEGER;
```

```
Temps : REAL;
```

```
END;
```

```
(*save state Using static variable*)
FUNCTION CheckTemps(hour,minute : INTEGER; temp : REAL) : BOOLEAN;
CONST
prevTemp : REAL = 0;
prevTime : INTEGER = 0;
currentTime : INTEGER = 0;
difTime : INTEGER = 0;
BEGIN
  IF prevTime = 0 THEN prevTime := 1;

  currentTime := minute + (hour*60);

  IF prevTemp = 0 THEN prevTemp := currentTime;

  difTime := currentTime - prevTime;

  IF (temp >= 935.5)
    AND (temp <=1345.6)
    AND (prevTemp - temp >= -11.5*difTime)
    AND (prevTemp - temp <= 11.5*difTime) THEN BEGIN
    CheckTemps := TRUE;
    prevTemp := temp;
    prevTime := currentTime;
  END

  ELSE BEGIN
    CheckTemps := FALSE;
  END

END;

VAR
tempsMeasurement : measurement;
ovenState : STRING;
BEGIN
  WriteLn('Soll die Temperatur weiter gemessen werden? Bitte mit ja / nein Antworten');
  ReadLn(ovenState);
  WHILE ovenState = 'ja' DO BEGIN

    WriteLn('Bitte Stunden, Minuten , Temperauter der Messung eingeben');
    ReadLn(tempsMeasurement.Hour,tempsMeasurement.Minute,tempsMeasurement.Temps);
```

```
WriteLn(CheckTemps(tempsMeasurement.Hour, tempsMeasurement.Minute, tempsMeasurement.Temps));
```

```
WriteLn('Soll die Temperatur weiter gemessen werden? Bitte mit ja / nein Antworten');  
ReadLn(ovenState);  
END;  
WriteLn('Ok das Messen hoert auf');  
END.
```

TESTFÄLLE:

```
1)  
UnitTempPruef Loaded  
Soll die Temperatur weiter gemessen werden? Bitte mit ja / nein Antworten  
ja  
Bitte Stunden, Minuten , Temperauter der Messung eingeben  
12 30 1000  
TRUE  
Soll die Temperatur weiter gemessen werden? Bitte mit ja / nein Antworten  
ja  
Bitte Stunden, Minuten , Temperauter der Messung eingeben  
12 31 1009  
TRUE  
Soll die Temperatur weiter gemessen werden? Bitte mit ja / nein Antworten  
ja  
Bitte Stunden, Minuten , Temperauter der Messung eingeben  
12 40 1100  
TRUE  
Soll die Temperatur weiter gemessen werden? Bitte mit ja / nein Antworten  
ja  
Bitte Stunden, Minuten , Temperauter der Messung eingeben  
12 41 1300  
FALSE  
Soll die Temperatur weiter gemessen werden? Bitte mit ja / nein Antworten  
ja  
Bitte Stunden, Minuten , Temperauter der Messung eingeben  
12 41 1090  
TRUE  
Soll die Temperatur weiter gemessen werden? Bitte mit ja / nein Antworten  
nein  
Ok das Messen hH||rt auf  
█
```

2)

UnitTempPruef Loaded

Soll die Temperatur weiter gemessen werden? Bitte mit ja / nein Antworten
ja

Bitte Stunden, Minuten , Temperauter der Messung eingeben

12 30 9999

FALSE

Soll die Temperatur weiter gemessen werden? Bitte mit ja / nein Antworten
nein

Ok das Messen h||rt auf



3)

a)

PROGRAM Schachbrett;

VAR i,j,n,z,g : INTEGER;

begin

WriteLn('Geben sie bitte die GroeÙe des Feldes an');

ReadLn(n);

IF (n < 1) OR (n > 9) THEN BEGIN

WriteLn('Falsche Zahl eingegeben');

WHILE TRUE DO BEGIN

END;

END;//Überprüfen ob n>9

WriteLn;

z:= 0;

g:= 0;

FOR i:=0 TO n DO BEGIN

Write(g);

Inc(g);//Y-Achse (Dient nur zur Darstellung)

FOR j:=1 TO n DO BEGIN

IF i = 0 THEN BEGIN

Inc(z);

Write(' ', z);

END;//X-Achse (Dient nur zur Darstellung)

IF (j > 0) AND (i > 0) THEN BEGIN //Kein X oder . auf der X Achse

IF (i = j) OR (i + j = n + 1) THEN BEGIN //X

Write(' X');

```

        END
        ELSE // .
            Write(' .')
        END;
    END;

    WriteLn;
END;

end.

```

B)

Der Code hat Programmierstiltechnisch sehr viele Probleme. Erst mal der Name des Programms an sich. Schachbrett repräsentiert überhaupt nicht was das Programm macht. XMuster würde da eher passen. Dann wären die Variablenamen überhaupt nicht gut gewählt. Dann habe ich zum Beispiel das erste BEGIN kleingeschrieben und dann wieder Groß. Solche Unregelmäßigkeiten sollte man insgesamt sein lassen. Die Endlos-Schleife beim Angeben einer „falschen“ Zahl ist auch zu „overkill“. Abgesehen davon ist mir nichts anderes Kritikwürdiges aufgefallen.

C)

```

PROGRAM XMuster;

VAR
    i,j : INTEGER;
    fieldLength,xFigure,yFigure : INTEGER;
BEGIN
    WriteLn('Geben sie bitte die Groeße des Feldes an');
    ReadLn(fieldLength);

    IF (fieldLength < 1) OR (fieldLength > 9) THEN BEGIN
        WriteLn('Falsche Zahl eingegeben');
        Halt;
    END; //Überprüfen ob n>9

    WriteLn;
    xFigure:= 0;
    yFigure:= 0;

    FOR i:=0 TO n DO BEGIN
        Write(yFigure);
        Inc(yFigure); //Y-Achse (Dient nur zur Darstellung)

        FOR j:=1 TO n DO BEGIN

```

```
IF i = 0 THEN BEGIN
    Inc(xFigure);
    Write(' ', xFigure);
END; //X-Achse (Dient nur zur Darstellung)
IF (j > 0) AND (i > 0) THEN BEGIN //Kein X oder . auf der X Achse
    IF (i = j) OR (i + j = fieldLength + 1) THEN BEGIN //X
        Write(' X');
    END
    ELSE
        Write(' .')
    END;
END;
WriteLn;
END;

END.
```