**Aufgabe 1:** Definiere den Begriff *Algorithmus*. Welche vier Eigenschaften sind für den Algorithmusbegriff bestimmend? (2P)

**>> 1 BE Algorithmusbegriff**

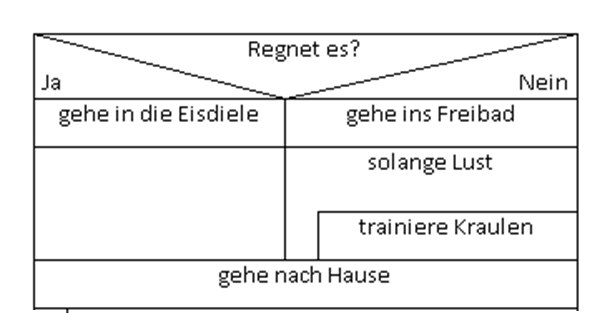
**Ein Algorithmus ist eine endliche Folge von eindeutigen und ausführbaren Anweisungen zur Lösung eines allgemeinen Problems.**

**>> 1 BE vier Eigenschaften**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Aufgabe 2:** Kevin sitzt am Frühstückstisch und plant seinen Tag: "Wenn es nicht regnet, gehe ich ins Freibad. Solange ich Lust habe, trainiere ich dann die 25-Meter-Kraulen-Disziplin. Sollte es regnen, dann gehe ich mit meiner Freundin in die Eisdiele. In beiden Fällen gehe ich anschließend nach Hause.“

Zeichne Kevins Tagesplan als Struktogramm. (4P)



**>> 1 BE richtige Zeichnung**

**>> 1 BE Bedingte Anweisung**

**>> 1 BE normale Anweisungen**

**>> 1 BE Schleife**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Aufgabe 3:** Finde und berichtige die Fehler in dem folgenden Programm zur Berechnung am Kreis. Ergänze weiterhin die Lücken sinnvoll, sodass ein funktionsfähiges TP-Programm entsteht! (4P)

program**~~m~~** Berechnungen\_am\_Kreis**; I**

uses crt**;**

VAR rad**,** flae**, umf:** real**; II**

BEGIN**~~;~~ I**

Clrs**cr; II**

**writeLn** (**'**Berechnungen am Kreis**'**)**; II**

wri**~~e~~**te(**'Radius** = ? **'**)**; II**

readLn(**~~'~~rad~~'~~**)**; II**

**flae:**= **pi \* rad \* rad** **; III**

umf**:**=**2** \* pi \* **rad; II**

writeLn(**'**Kreisfläche = **',** flae**:**10**:**2 **, '** FE**'**)**; II**

writeLn(**'**Umfang = **'** **,** umf**:**10**:**2 **,** **'** **LE'**)**; II**

**readln; II**

END**~~;~~ I**

**24 Lücken und Fehler 🡪 für jeweils 2,5 gefundene Fehler/ausgefüllt Lücken gibt es einen halben Punkt (abrunden). Punktabzüge sind möglich bei vielen falschen „Fehlervermutungen“ oder Ähnlichem. *[Bsp: 17 🡪 3/4, 18 🡪 3,5/4]***

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Aufgabe 4:** ***(a)*** Wozu dienen in Turbo-Pascal die Befehle „read“ und „write“? (1P)

**read = Dateneingabe, write = Datenausgabe**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***(b)*** Notiere die (theoretische) Ausgabe, die von folgenden Code-Segmenten erzeugt wird und benenne den Unterschied. (1P)

|  |  |
| --- | --- |
| Writeln(‚Hallo‘); Writeln(‚Welt‘); | Write(‚Hallo‘);  WriteLn(‚Welt‘); |
| **Hallo**  **Welt** | **Hallo Welt** |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***(c)*** Wofür wird bei Turbo-Pascal häufig die Zeile „ReadLn;“ vor dem „End.“-Befehl verwendet? (1P)

**Damit das Programm nicht sofort beendet wird**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Aufgabe 5:** ***(a)*** Beschreibe, wofür *Bedingte Anweisungen* und wofür *Schleifen* in der Programmierung benötigt werden. (2P)

**>> Je 1 BE Bedingte Anweisungen und Schleifen**

**Bedingungen bieten die Möglichkeit, verschiedene Anweisungen abhängig von vorangegangenen Schritten auszuführen**

**Schleifen bieten die Möglichkeit, einen Anweisungsblock mehrmals auszuführen**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***(b)*** Nenne zwei Variablentypen aus Turbo-Pascal und beschreibe die Unterschiede zwischen Ihnen. (1P)

**z.B. integer(Ganze Zahlen) und real (Kommazahlen)**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Aufgabe 6:** Ergänze den folgenden Code, sodass ein Turbo-Pascal-Programm entsteht, welches zwei (von einem Benutzer eingegebene) Zahlen addiert und dann deren Summe ausgibt. Im Kästchen auf der rechten Seite findest du die Ausgabe, die dein Programm erzeugen soll. *(Dabei sind die Werte 47 und 14 nur Beispielwerte, die von einem Benutzer eingegeben wurden)* (5P)

Addition zweier Zahlen

===================

1. Summand: 47

2. Summand: 14

Summe = 61

program Addition**;**

uses crt**;**

VAR zahl1**,** zahl2**,** summe**:**integer**;**

BEGIN

**Clrscr;** *optional*

**writeLn(‚Addition zweier Zahlen‘);**

**writeLn(‚=================== ‘);**

**writeLn;**

**write(‚1. Summand: ‘);**

**readLn(zahl1);**

**write(‚2. Summand: ‘);**

**readLn(zahl2);**

**writeLn;**

**summe:=zahl1+zahl2;**

**writeLn(‚Summe = ‘, summe);**

readln**;**

END**.**

**>> 1 BE Ausgabe ähnlich der Vorgegebenen**

**>> 0,5 BE Begrüßung/Erklärung**

**>> 1,5 BE Einlesen**

**>> 1 BE Verarbeitung**

**>> 1 BE Ausgabe**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Zusatzaufgabe:** Notiere den Turbo-Pascal-Code für eine Schleife, die alle Quadratzahlen der Zahlen von 1 bis 20 ausgibt. (1ZP)

**PROGRAM for\_schleife;**

**VAR i:INTEGER;**

**BEGIN**

**FOR i:=1 TO 20 DO**

**WriteLn(i, ' ',i\*i);**

**ReadLn;**

**END.**

**Es reicht auch nur die Schleife, oder nur die Quadratzahlen, ohne die Zahl, die quadriert wird.**