

## Fachhochschule Bingen

## Programmieren

## Algorithmus / Struktogramme

Prof. Dr. Maximilian Mengel,  
Professur Programmiermethodik,  
Grundlagen der Informatik und Multimedia  
Gebäude 1, Raum 212  
Tel.: 06721-409 152  
E-Mail: [mengel@fh-bingen.de](mailto:mengel@fh-bingen.de)

### Aufgabe: Zubereitung Lasagne

#### ■ Zubereitung in mehreren Schritten:

1. Zutaten besorgen
2. Zubereitung der Hackfleischseinlage
3. Zubereitung der Bechamel-Soße
4. Lasagne Schichten und Backen
5. Essen

04.10.2003

2

### Aufgabe: Zubereitung Lasagne

#### ■ Besorgen der Zutaten

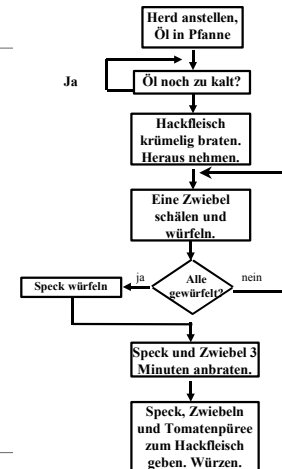
- 500 g Lasagneplatten
- 80g Mozzarella
- Hackfleischsoße:
  - 2 EL Öl
  - 300g Hackfleisch
  - 2 Zwiebel
  - 50g Speck
  - 250g Tomatenpüree
- Bechamelsoße
  - 50g Butter
  - 5 EL Mehl
  - 300 ml Milch
  - 50g Parmesan (gerieben)

04.10.2003

3

### Aufgabe: Zubereitung Lasagne

#### ■ Zubereitung der Hackfleischseinlage



04.10.2003

4

## ... Aufgabe: Zubereitung Lasagne

### ■ Zubereitung der Soße

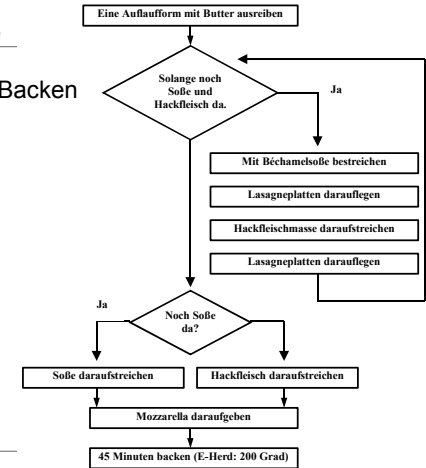


04.10.2003

5

## ... Aufgabe: Zubereitung Lasagne

### ■ Schichten und Backen



04.10.2003

6

## Algorithmus

### ■ Algorithmusbegriff

- Beschreibung einer Lösung, die von einem Computer ausgeführt werden kann
- Zu beachten sind die Regeln der
  - Syntax (formale Regeln)
  - Semantik (Bedeutung der Konstrukte untereinander)

➡ Ein Algorithmus ist eine detaillierte und explizite Vorschrift zur schrittweisen Lösung eines Problems

04.10.2003

7

## Algorithmus

### ■ Eigenschaften von Algorithmen

#### ■ Allgemeinheit

- Ein Algorithmus löst im allgemeinen eine Klasse von Problemen. Die Auswahl des Einzelfalles erfolgt meist über Parameter.
- Ein Algorithmus entsteht oft dadurch, dass ein bestimmter Fall gelöst wird und andere Situationen darauf zurückgeführt werden.

04.10.2003

8

### ■ Eigenschaften von Algorithmen

#### ■ **Determiniertheit**

- | Algorithmen sind in der Regel determiniert, d.h. bei gleichen Eingabewerten und Startbedingungen erfolgt stets dasselbe Ergebnis.
- | Beispiele für nicht determinierte Algorithmen sind z.B. stochastische Simulationen mit Hilfe von Zufallszahlen.

### ■ Eigenschaften von Algorithmen

#### ■ **Determinismus**

- | Ein Algorithmus genügt dem Determinismus, wenn zu jedem Zeitpunkt seiner Bearbeitung höchstens eine Möglichkeit der Fortsetzung besteht.

### ■ Eigenschaften von Algorithmen

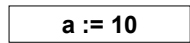
#### ■ **Terminierung**

- | In der Regel sind nur solche Algorithmen von Interesse, die für jede Eingabe nach endlichen vielen Schritten terminieren, d.h. anhalten.
- | Eine Ausnahme sind hier Betriebssystem-Funktionen und Prozeß-Steuerungen.

### ■ Beschreibungsmöglichkeiten

- Flußdiagramme
- Stuktogramme
- Programmiersprachen

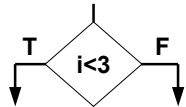
## Flußdiagramme



- Elementaroperationen
  - z.B. Zuweisungen, Operationen, ...



- Pfeile führen zur nächsten Aktion



- Test: Ist die Bedingung erfüllt folgt man dem T-Pfeil, ansonsten dem F-Pfeil

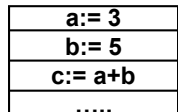


- Definierte Übergangspunkte
  - z.B. Beginn, Ende

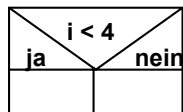
## Flußdiagramme

- Vorteile
  - Einfach zu verstehen
  - Sehr flexibel
- Nachteile
  - Schnell unübersichtlich
  - Keine strenge Strukturierung
  - Für den Computer nicht ausführbar

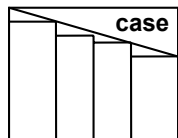
## Struktogramme



- Sequenz
  - Folge von Anweisungen / Elementaroperationen

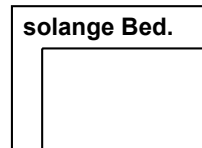


- Bedingung / Verzweigung

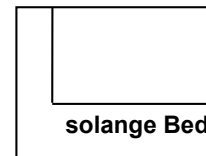


- 1-aus N Auswahl

## Struktogramme



- Abweisende Schleife
  - Solange Bedingung erfüllt bearbeite....



- Nicht abweisende Schleife
  - Solange Bedingung erfüllt wiederhole Bearbeitung ...

## Struktogramme

### ■ Vorteile

- Einfach zu verstehen
- Streng Strukturierend

### ■ Nachteile

- Platzbedarf
- Für den Computer nicht ausführbar

## Programmiersprachen: z.B. Pascal

a := 3;

b := 5;

■ Ausdrücke, Sequenz

WHILE a<3 DO ...

■ While-Schleife

IF a<3 THEN ... ELSE ...

■ Bedingte Anweisungen

## Programmiersprachen: z.B. C

a = 3;

b = 5;

■ Ausdrücke, Sequenz

WHILE ( a<3 ) ...

■ While-Schleife

IF (a<3) ... ELSE ...

■ Bedingte Anweisungen

## Programmiersprachen

### ■ Vorteile

- Von Computern ausführbar
- Je nach Programmiersprache mehr oder weniger stark strukturierend

### ■ Nachteile

- Nicht so intuitiv verständlich