

Fachhochschule Bingen

Programmieren

Wiederholung Rekursion

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

Rekursive Funktionen (direkte Rekursion)

- In einer rekursiven Funktion wird
 - In bestimmten Fällen das Ergebnis der Funktion direkt ausgerechnet bzw. bestimmt
 - In bestimmten Fällen die Funktion selbst wieder aufgerufen
 - Wird die Funktion selbst wieder aufgerufen, so muß dies so geschehen, daß sich der Aufwand zur Berechnung der Funktion verringert hat

21.03.2004

2

Beispiel einer Rekursiven Funktion

■ N!

```
int fak(int N)
{
    int erg;
    if (N < 2) erg = 1;
    else erg = N * fak(N-1);
    return erg;
}

main()
{
    printf(" %d! = %d\n", 7, fak(7));
}
```

21.03.2004

3

Beispiel einer Rekursiven Funktion

■ ggT (A, B)

```
int ggt(int A, int B)
{
    int erg;
    if (A == B) erg = A;
    else
        if (A > B) erg = ggt(A-B, B);
        else erg = ggt(A, B-A);
    return erg;
}

main()
{
    printf("ggT(%d, %d) = %d\n", 140, 78, ggt(140,78));
}
```

21.03.2004

4

Rekursive Prozeduren

- Neben rekursiven Funktionen zur Berechnung von mathematischen Funktionen existieren auch sonstige Probleme die elegant rekursiv lösbar sind
 - Ausgabe einer (Dezimal-)Zahl als Dualzahl
 - Die Türme von Hanoi

Horner Schema Rekursion

- Um eine Dezimalzahl als Dualzahl auszugeben kann folgende Prozedur benutzt werden

```
WriteDual(int A)
{
    if (A > 1)
    {
        WriteDual((int)(A / 2));
        WriteDual(A % 2);
    }
    else printf("%d",A);
}
```

Türme von Hanoi

```
void hanoi(int Anzahl, int *Start, int *Lager, int *Ziel)
{
    if (Anzahl == 1) move(Start,Ziel);
    else
    {
        hanoi(Anzahl-1, Start, Ziel, Lager);
        move(Start, Ziel);
        hanoi(Anzahl-1, Lager, Start, Ziel);
    }
}
```

Indirekte Rekursion

- Bei der indirekten Rekursion existieren mehrere Funktionen bzw. Prozeduren die sich wechselseitig aufrufen

```
int odd(int Zahl)
{
    if (Zahl == 1) return 1;
    else return even(Zahl-1);
}

int even(int Zahl)
{
    if (Zahl == 1) return 0;
    else return odd(Zahl-1);
}
```

Backtracking

- A) Festlegen eines möglichen Lösungsschrittes
- B) Rekursiver Aufruf mit der Problemgröße $N-1$
- C) Überprüfen ob Erfolg
 - Ja: Lösung gefunden
 - Nein: Lösungsschritt revidieren und anderen Lösungsschritt ausprobieren => A)
- Wenn kein Lösungsschritt zum Erfolg führt, dann ist das Problem nicht lösbar
- Beispiele: Damenproblem, Springerproblem, Labyrinth