

Imperative Sprachen

Tail Rekursion:

- Tail Call bei rekursiven Funktionen
- Tail Call: letzte Anweisung in Funktionsrumpf von funk:
Funktionsaufruf ohne weitere Berechnungen
„drumherum“
funkt := g(...);
- bei Rekursion: funk := funk(...);
- → pro Rekursionsaufruf einen Activation Record,
- aber: Ergebnis durchreichen nach „ganz oben“
- → Activation records der vorherigen Durchläufe können gelöscht werden (bzw. durch neuen überschrieben werden)
- Alternative Optimierung:
funkt := „iterative Formulierung der Funktion“
benötigt gleichfalls nur diesen einen Activation Record!

Imperative Sprachen

- Beispiel:

```
FUNCTION fakul(n,a:integer):integer;  
begin  
    if n <= 1 then fakul := a  
    else fakul:=fakul(n-1, n*a)  
  
end;
```

- Iterativ:

```
FUNCTION fakul2(n,a:integer):integer;  
begin  
    while (n>0) do begin  
        a := n*a;  
        n := n-1;  
    end;  
    fakul2 := a;  
  
end;
```

