

Programmiersprachen

Schluss

Berthold Hoffmann

Studiengang Informatik
Universität Bremen

Wintersemester 2004/2005
(Vorlesung am 14. Februar 2005)

Berthold Hoffmann

Programmiersprachen

Zusammenfassung Ausblick Bewertung Organisatorische Konzepte Fallstudie

Konzepte

Module & Schnittstellen		
abstrakter Datentyp	Klasse	generisches Modul
Überladen Polymorphie Vererbung	Sprung Ausweg Ausnahme	
Abstraktion & Parameter		
Funktion	Prozedur	generische Vereinbarung
Datenstrukturen Ausdrücke	Speicher Befehle	Vereinbarungen Blöcke
<i>funktional</i>	<i>imperativ</i>	<i>deklarativ</i>

Berthold Hoffmann

Programmiersprachen

Zusammenfassung Ausblick Bewertung Organisatorische Konzepte Fallstudie

Meilensteine

- LISP: Listen und Rekursion
- PASCAL: Datenstrukturen und Kontrollstrukturen
- SIMULA: Klassen
- ML: Polymorphie und Typinferenz
- SMALLTALK: Klassen und dynamisches Binden
- MIRANDA: *pattern matching* und Listenumschreibungen
- EIFFEL *design by contract*
- JAVA: Plattformunabhängigkeit

Berthold Hoffmann

Programmiersprachen

Zusammenfassung Ausblick Bewertung Organisatorische Konzepte Fallstudie

Quicksort imperativ (ADA)

```

type Item is ...;
type ItemSequence is array (Integer range <>) of Item;

procedure sort (items : in out ItemSequence) is
  left, right : Integer;
begin
  if items'first < items'last
  then partition (items, left, right);
    sort (items(items'first..right));
    sort (items(left..items'last));
  end if;
end;

```

Berthold Hoffmann

Programmiersprachen

1 Zusammenfassung

- Konzepte
- Fallstudie

2 Ausblick

3 Bewertung

4 Organisatorische

Berthold Hoffmann

Programmiersprachen

Zusammenfassung Ausblick Bewertung Organisatorische Konzepte Fallstudie

Prinzipien

- Typvollständigkeit von Operationen (Orthogonalität)
- Qualifizierung von Vereinbarungen
- Abstraktion von Berechnungen
- Korrespondenz von Parametermechanismen zu Vereinbarungen
- Verbergen von Implementierungsdetails in Modulen

Berthold Hoffmann

Programmiersprachen

Zusammenfassung Ausblick Bewertung Organisatorische Konzepte Fallstudie

Paradigmen

- Imperativ
 - Befehle verändern Zustände (Variablenbelegungen)
- Objektorientiert
 - Objekte tauschen Nachrichten aus
- Funktional
 - Ausdrücke werden ausgewertet
- Logisch
 - Zielklauseln werden abgeleitet

Berthold Hoffmann

Programmiersprachen

Zusammenfassung Ausblick Bewertung Organisatorische Konzepte Fallstudie

Prozedur zum Aufteilen des Feldes

```

procedure partition (items : in out ItemSequence;
  left, right : out Integer) is
  l : Integer := items'first;
  r : Integer := items'last;
  pivot : constant Item := items(l);
  procedure swapitems is new swap (Item);
  begin loop while items(l)<pivot loop l:=l+1; end loop;
    while pivot<items(r) loop r:=r-1; end loop;
    exit when l > r;
    swapitems (items(l), items(r));
    l := l+1; r := r-1; exit when l > r;
  end loop;
  left := l; right := r;
end;

```

Berthold Hoffmann

Programmiersprachen

Quicksort funktional (ML)

```

fun sort (nil) = nil
  | sort [single] = [single]
  | sort (first::others) =
    let val (smalls, bigs) = partition(first, others)
    in sort (smalls) @ [first] @ sort (bigs)
    end
fun partition (pivot, nil) = (nil, nil)
  | partition (pivot, first::others) =
    let val (smalls, bigs) = partition (pivot, others)
    in if first < pivot
       then (first::smalls, bigs)
       else (smalls, first::bigs) end
    end

```

Quicksort logisch (PROLOG)

```

sort([], []).
sort([Single], [Single]).
sort([First|Others], Sorted) :-
  partition(First, Others, Smalls, Bigs),
  sort(Smalls, SortedSmalls), sort(Bigs, SortedBigs),
  append(SortedSmalls, [First|SortedBigs], Sorted).

partition(P, [], [], []).
partition(P, [First|Others], [First|Smalls], Bigs) :-
  First < P, partition(P, Others, Smalls, Bigs).
partition(P, [First|Others], Smalls, [First|Bigs]) :-
  not(First < P), partition(P, Others, Smalls, Bigs).

```

Schluss (14. Februar 2005)

- 1 Zusammenfassung
- 2 **Ausblick**
- 3 Bewertung
- 4 Organisatorische

Schluss (14. Februar 2005)

- 1 Zusammenfassung
- 2 Ausblick
- 3 **Bewertung**
 - Ziele
 - Evaluation
 - Veränderung
- 4 Organisatorische

Quicksort funktional (HASKELL)

```

sort [] = []
sort [x] = [x]
sort (p:r)
  = sort [l | l <-r, l<p] ++ [p] ++ sort [l | l <-r,

```

Vergleich der Stile

- Imperativ
 - wenig Speicherplatzverbrauch (Nur $\log(n)$ Indizes)
 - Korrektheit nicht trivial
- Funktional
 - mehr Speicherplatzverbrauch (für sortierte Teillisten)
 - Korrektheit ziemlich einfach zu sehen
 - Listenumschreibungen sind sehr intuitiv (aber ineffizient)
- Logisch
 - mehr Speicherplatzverbrauch (für sortierte Teillisten)
 - Korrektheit auch ziemlich einfach zu sehen
 - Prädikate sind etwas umständlicher als Funktionen
- **Fazit:** ein Paradebeispiel für funktionales Programmieren

Tendenzen

Verlagerung des Schwerpunkts des Programmierens

- Algorithmen
- Datenmodellierung
- Prozessmodellierung
- Spezifikation und Verifikation?

Abwägungen beim Sprachentwurf

- Effizienz
- Sicherheit
- Programmierkomfort

Meine Lehrziele / Eure Lernziele?

- 1 Die Prinzipien von Programmiersprachen kennen
- 2 Den Kern der Paradigmen verstehen
 - Mit einer semantisch geprägte Terminologie und Klassifikation
- 3 Neue Sprachen schnell und gründlich lernen können
- 4 Programmiersprachen beurteilen und vergleichen können
- 5 Sprachen entwerfen können (??)
- 6 Einige neuartige Sprachen kennen lernen

Eure Meinung zur Veranstaltung

- Wurden die Lehrzeile erreicht?
- Was hättet Ihr gerne noch gelernt?
- Was hättet Ihr lieber nicht gelernt?

Auswertung der Bögen

-

Mein Eindruck von den Studierenden

- + Regelmäßige Teilnahme
- Regelmäßiges "Vergessen" der Übungsaufgaben
- + Einige haben sich viel beteiligt
- ? Haben die Anderen auch viel gelernt?

Eure Wünsche an mich

-

Meine Wünsche an die Studierenden

- Bessere Vorbereitung der Übungsaufgaben
- Selbstständiges Stöbern in Sprachbeschreibungen
- Ausprobieren neuer Sprachen

Meine Pläne für die Veranstaltung

- Noch klarere Trennung von
 - Essenz der Konzepte und Paradigmen (Semantik)
 - Phänomene der Realisierungen in konkreten Sprachen
- Beispielprogramme zur Vorlesung *online* zum Ausprobieren

Schluss (14. Februar 2005)

- 1 Zusammenfassung
- 2 Ausblick
- 3 Bewertung
- 4 **Organisatorische**
 - Prüfung
 - Werbepause

Prüfungstermine (Ort: MZH 8130)

Zeit	MI 16.2.	DO 3.3.	FR 4.3.	DI 29.3.
8:00	wmachert		pbecker	
8:30			tommy01	
9:00	barann		bjenny	
9:30	fharjes		riky2008	xiangyu
10:00	jck		marcelk	miro
10:30	margo		tica	manuel
11:00	roscoe		twarode	dkramer
11:30	fxl		astein	leisun
12:00	fransder		jessie	kraetsch
12:30			hunecker	

Noch unentschieden: tfreese und dambroth

Gegenstand der Prüfung

- Der Stoff der Veranstaltung
 - Die allgemeinen Konzepte und Prinzipien ...
 - ... und ihre Ausformung in einer wählbaren Sprache
- Was kommt ziemlich sicher dran?
 - Zusammengesetzte und rekursive Datentypen
 - Parameterübergabe
 - Abstrakte Datentypen und Klassen
 - Überladen, Polymorphie und Vererbung

Berthold Hoffmann

Programmiersprachen

Zusammenfassung Ausblick Bewertung Organisatorische Prüfung Werbepause

Nachlese(n)

 David A. Watt.
Programming Language Design Concepts.

Chichester: Wiley & Sons, 2004.

Wichtige Konferenzen zum Thema Programmiersprachen

- *Principles of Programming Languages*
ACM SIGPLAN / SIGACT.
<http://www.cs.princeton.edu/dpw/pop1/05/>
- *Programming Language Design and Implementation*
ACM SIGPLAN.
<http://research.ihost.com/pldi2005/>

Tagungsbände erscheinen bei SIGPLAN-Notices / ACM Press.

Berthold Hoffmann

Programmiersprachen

Fortsetzung im Sommersemester

Übersetzer

- Montags 8 – 12 in MZH 1380
- Implementierungstechniken für Programmiersprachen
 - Interpreter, Übersetzer, Umgebungen, Phasenmodell
 - lexikalische, syntaktische, kontextuelle Analyse
 - Werkzeuge lex und yacc
 - Codeerzeugung für abstrakte Maschinen

Übersetzerpraktikum

- Dienstags 13–15 in MZH 4194 (von Günter Feldmann)
- Vertiefte Anwendung von Übersetzerbau-Techniken
 - Vorgegeben: PL0-System (in Oberon oder in ML?)
 - Ziel: Schrittweise Erweiterung der Sprache (bis PL5)

Berthold Hoffmann

Programmiersprachen

Zusammenfassung Ausblick Bewertung Organisatorische Prüfung Werbepause

Tschüß

Schöne vorlesungsfreie Zeit!

Berthold Hoffmann

Programmiersprachen