Die Programmiersprache C
Eine Einführung

Christian Gentsch

Fakutltät IV
Technische Universität Berlin
Projektlabor

2. Mai 2014

Inhaltsverzeichnis

Einführung

Quelle: *http://commons.wikimedia.org/wiki/User:Rezonansowy*

Geschichtlicher Abriss

Geschichtlicher Abriss

- Entwickelt in der Zeit von 1969-73 von Dennis Ritchie

Geschichtlicher Abriss

- Entwickelt in der Zeit von 1969-73 von Dennis Ritchie
- Weiterentwicklung der Sprache B

Geschichtlicher Abriss

- Entwickelt in der Zeit von 1969-73 von Dennis Ritchie
- Weiterentwicklung der Sprache B
- 1978 erste Auflage „The C programming language“

Geschichtlicher Abriss

- Entwickelt in der Zeit von 1969-73 von Dennis Ritchie
- Weiterentwicklung der Sprache B
- 1978 erste Auflage „The C programming language“
- Kommittee zur Normierung der Sprache
  $\Rightarrow$ 1989 ANSI.X3.159-1989

Geschichtlicher Abriss

- Entwickelt in der Zeit von 1969-73 von Dennis Ritchie
- Weiterentwicklung der Sprache B
- 1978 erste Auflage „The C programming language“
- Kommittee zur Normierung der Sprache
  $\Rightarrow$ 1989 ANSI.X3.159-1989
- Heute ISO/IEC 9899 (C99) bzw. C11 Norm

Hauptanwendungsgebiete von C

Hauptanwendungsgebiete von C

- Systemprogrammierung; einschl. Betriebssysteme

Hauptanwendungsgebiete von C

- Systemprogrammierung; einschl. Betriebssysteme
- eingebettete Systeme

Hauptanwendungsgebiete von C

- Systemprogrammierung; einschl. Betriebssysteme
- eingebettete Systeme
- Anwendungssoftware

Hauptanwendungsgebiete von C

- Systemprogrammierung; einschl. Betriebssysteme
- eingebettete Systeme
- Anwendungssoftware
- Compiler, Programmbibliotheken und Interpreter anderer höherer Programmiersprachen oftmals in C (bspw. Java)

Hauptanwendungsgebiete von C

- Systemprogrammierung; einschl. Betriebssysteme
- eingebettete Systeme
- Anwendungssoftware
- Compiler, Programmbibliotheken und Interpreter anderer höherer Programmiersprachen oftmals in C (bspw. Java)
- Zwischencode

Vergleich zweier Programmierparadigmen

Vergleich zweier Programmierparadigmen

Objektorientiert

- Alles ist ein Objekt
- Objekte kommunizieren miteinander
- Objekte haben eigenen Speicher
- Jedes Objekt ist ein Exemplar einer Klasse
- Die Klasse modelliert das gemeinsame Verhalten
- Programm wird ausgeführt, indem dem ersten Objekt die Kontrolle übergeben wird

Vergleich zweier Programmierparadigmen

Objektorientiert

- Alles ist ein Objekt
- Objekte kommunizieren miteinander
- Objekte haben eigenen Speicher
- Jedes Objekt ist ein Exemplar einer Klasse
- Die Klasse modelliert das gemeinsame Verhalten
- Programm wird ausgeführt, indem dem ersten Objekt die Kontrolle übergeben wird

Imperativ

- Festlegung in welcher Reihenfolge was zu tun ist und wie
- Schrittweises Fortschreiten der Befehle
- Steuerung der Befehlsausführung über Kontrollstrukturen (Sequenzen, Schleifen, Verzweigungen)

Operatoren in C

Operatoren in C

- Arithmetische Operatoren

Operatoren in C

- Arithmetische Operatoren
  - Addition, Subtraktion, Multiplikation, Division, Modulo

Operatoren in C

- Arithmetische Operatoren
  - Addition, Subtraktion, Multiplikation, Division, Modulo
  - Inkrement, Dekrement

Operatoren in C

- Arithmetische Operatoren
  - Addition, Subtraktion, Multiplikation, Division, Modulo
  - Inkrement, Dekrement
  - Zuweisungen

Operatoren in C

- Arithmetische Operatoren
  - Addition, Subtraktion, Multiplikation, Division, Modulo
  - Inkrement, Dekrement
  - Zuweisungen
- Relationale Operatoren

Operatoren in C

- Arithmetische Operatoren
  - Addition, Subtraktion, Multiplikation, Division, Modulo
  - Inkrement, Dekrement
  - Zuweisungen
- Relationale Operatoren
- Logische Operatoren

Operatoren in C

- Arithmetische Operatoren
  - Addition, Subtraktion, Multiplikation, Division, Modulo
  - Inkrement, Dekrement
  - Zuweisungen
- Relationale Operatoren
- Logische Operatoren
- Bitmanipulation

Operatoren in C

- Arithmetische Operatoren
  - Addition, Subtraktion, Multiplikation, Division, Modulo
  - Inkrement, Dekrement
  - Zuweisungen
- Relationale Operatoren
- Logische Operatoren
- Bitmanipulation
- Zeigeroperatoren

Operatoren in C

- Arithmetische Operatoren
  - Addition, Subtraktion, Multiplikation, Division, Modulo
  - Inkrement, Dekrement
  - Zuweisungen
- Relationale Operatoren
- Logische Operatoren
- Bitmanipulation
- Zeigeroperatoren
  - Adressoperator &

Operatoren in C

- Arithmetische Operatoren
  - Addition, Subtraktion, Multiplikation, Division, Modulo
  - Inkrement, Dekrement
  - Zuweisungen
- Relationale Operatoren
- Logische Operatoren
- Bitmanipulation
- Zeigeroperatoren
  - Adressoperator &
  - Verweisoperator $*$

Operatoren in C

- Arithmetische Operatoren
  - Addition, Subtraktion, Multiplikation, Division, Modulo
  - Inkrement, Dekrement
  - Zuweisungen
- Relationale Operatoren
- Logische Operatoren
- Bitmanipulation
- Zeigeroperatoren
  - Adressoperator &
  - Verweisoperator $*$

```
1 int a;
2 int *zeiger;
3 zeiger = &a; //Zeiger verweist nun auf a
4 *zeiger = 10; //setzt den Wert von a auf
      10
```

If, else Anweisungen

If, else Anweisungen

- Verhalten analog zu Java

If, else Anweisungen

- Verhalten analog zu Java

```
1 if( a > 0 ){
2 printf("positiv");
3 }else if( a < 0 ){
4 printf("negativ");
5 }else{
6 printf("null");
7 }
```

Switch

Switch

- Wird verwendet, wenn unter vielen Bedingungen ausgewählt werden soll

Switch

- Wird verwendet, wenn unter vielen Bedingungen ausgewählt werden soll
- break-Anweisung notwendig

Switch

- Wird verwendet, wenn unter vielen Bedingungen ausgewählt werden soll
- break-Anweisung notwendig

```
1  int x;
2  scanf("%d", &x);
3  switch (x) {
4  case 1: y = 1;
5  break;
6  case 2: y = 4;
7  break;
8  case 3; y = 9;
9  break;
10 .
11 .
12 default: y = 0;
13 }
```

Beispiel: Flächeninhaltsfunktion

Beispiel: Flächeninhaltsfunktion

```
1  double flaeche_inhalt(double a, double b){
2  double flaeche;
3  flaeche = a * b;
4  return(flaeche);
5  }
6
7  int main(void){
8  double a = 5;
9  double b = 3;
10 flaeche_inhalt(a,b);
11 }
```

Strukturen

Strukturen

- Zusammenfassen mehrerer primitiver oder komplexer Datentypen

Strukturen

- Zusammenfassen mehrerer primitiver oder komplexer Datentypen
- Variablen dürfen unterschiedliche Datentypen besitzen

Strukturen

- Zusammenfassen mehrerer primitiver oder komplexer Datentypen
- Variablen dürfen unterschiedliche Datentypen besitzen

```
1 struct datum{
2 int tag;
3 char monat[12];
4 int jahr;
5 } geburtstag, feiertag; //Variablen vom Typ
    datum werden erzeugt
6
7 struct datum geburtstag = {19, "Januar",
    1994}
8 feiertag.jahr = 1989;
```

Aufzählungen

Aufzählungen

- Definition vieler Konstanten, die miteinander verwandt sind

Aufzählungen

- Definition vieler Konstanten, die miteinander verwandt sind
- werden als Integer interpretiert

Aufzählungen

- Definition vieler Konstanten, die miteinander verwandt sind
- werden als Integer interpretiert

```
1 enum woche {Mo=1, Di, Mi, Do, Fr, Sa, So};
```

Programmbeispiel

Rekursive Fakultät programmiert in C

```c
#include <stdio.h>

int fak(int n){
int erg;
if(n == 0){
erg = 1;
}else{
erg = n * fak(n-1);
}
return(erg);
}
int main(void){
int n;
printf("Bitte eine Zahl eingeben!\n");
scanf("%d", &n);
printf("%d!=%d\n", n, fak(n));
}
```

Vergleich von C und Java

- Programmbeispiel in C

```c
#include <stdio.h>

int fak(int n){
int erg;
if(n == 0){
erg = 1;
}else{
erg = n * fak(n-1);
}
return(erg);
}
int main(void){
int n;
printf("Bitte eine Zahl
      eingeben!\n");
scanf("%d", &n);
printf("%d!=%d\n", n, fak(n));
}
```

- Programmbeispiel in Java

```java
public class Fakultaet{

public static int fak(int n){
int erg;
if(n == 0){
erg = 1;
}else{
erg = n * fak(n-1);
}
return erg;
}
public static void main (
      String args[]){

System.out.println(fak(5));
}

}
```

Zusammenfassung

Zusammenfassung

- C wurde seit den 70ern oftmals überarbeitet und standardisiert

Zusammenfassung

- C wurde seit den 70ern oftmals überarbeitet und standardisiert
- C und Java haben starke syntaktische Übereinstimmungen

Zusammenfassung

- C wurde seit den 70ern oftmals überarbeitet und standardisiert
- C und Java haben starke syntaktische Übereinstimmungen
- Wer Java bereits kennt hat meist einen leichten Umstieg auf C

Quellenangabe

- Vorlesung: Einführung in die Informatik I (Technikorientiert) - Prof. Dr.-Ing. Olaf Hellwich
- http://de.wikibooks.org/wiki/C-Programmierung