Kapitel 1 - Einführung

Ziele der Veranstaltung

Organisatorisches

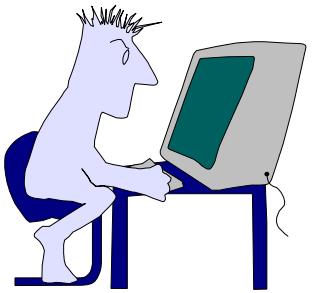
Die Sprache C++

Was ist Programmieren ?

? "Programmierer" ist ein Lehrberuf ?

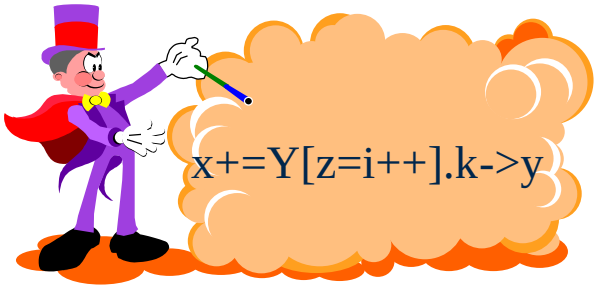
? Verlagerung der Programmierung in Billiglohnländer ?

- Strukturierung komplexer Probleme
  - divide and conquer (teile und herrsche)
  - Übung für viele andere Aufgaben
- äußerst kreative Tätigkeit
  - Entwurf und Realisierung
  - nicht nur kodieren vorgegebener Abläufe
- Handwerkliche Fertigkeiten
  - Beherrschung der Ausdrucksmittel einer Programmiersprache
- Kundenorientierung

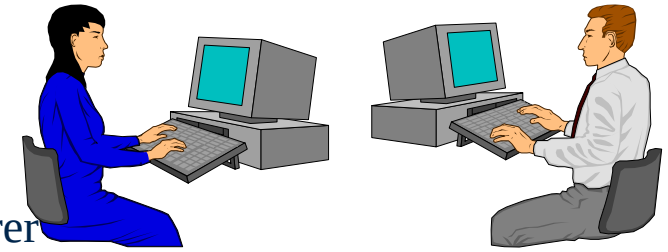


Das Bild des Programmierers

- früher: Hacker, Guru, Wizard
  - tricky programming
  - Hauptsache, der Compiler versteht mich
  - findet sich teilweise auch in der Literatur
  - siehe: annual contest on obfuscated C



- heute: Informatiker, Ingenieur
  - Programmieren als eine Tätigkeit im systematischen Entwurfsprozeß
  - Teamwork, nicht Einzelkämpfer
  - Wiederverwendung von Code
    - leichte Verständlichkeit für andere Programmierer



Stroustrup zu "Programmieren"

Gedanken aus B. Stroustrup:
Die C++ Programmiersprache

- Programm-Design und Programmieren sind menschliche Aktivitäten.

- Es gibt kein "allgemeines Rezept", das Intelligenz, Erfahrung und "guten Geschmack" ersetzen kann.
- Das Experimentieren ist für alle nicht-trivialen Software-Projekte essentiell.
- Entwurf und Programmieren sind iterative Aktivitäten.

- Die Systeme, die wir entwickeln, tendieren stets zu einer Komplexitäts-Obergrenze, die durch uns selbst und unsere Werkzeuge gesetzt ist.

Aufgaben einer Programmiersprache

- vordergründig:
  spezifiziert Arbeitsanweisung für Computer
- spiegelt das Denken wider und prägt es zugleich
- ist Verständigungsmittel
  - Mensch • Mensch
  - Mensch • Computer

es gibt gute und schlechte Ausdrucksweise, genau wie in jeder natürlichen Sprache

- Konsequenzen:
  - ein Programm, das nur der Erfinder und sein Computer versteht, ist schlecht
  - "Hauptsache, es tut" wird nicht akzeptiert !

Ihre Vorerfahrungen interessieren mich !

... um daran anzuknüpfen

- Wer hat einen privaten PC, bzw. Zugang zu einem ?
  - oder Mac, Amiga, Atari, ...
- Wer hat schon programmiert ?
  - als Hobby
  - Informatik AG in der Schule
  - als Job für eine Firma
- in welcher Programmiersprache ?
  - (Visual) Basic, Delphi, Turbo Pascal, C, C++, Java, ...
- ... und welche Projekte, inhaltlich gesehen ?

Ablauf einer Praktikumsübung

- Ausgabe der Aufgabenstellung
- Vorbesprechung und Entwurf in Hörsaalübung
- selbständig zu Hause oder in den Labors
  - Verfeinerung des Entwurfs
  - Ausarbeitung des Programms inklusiv Eintippen
  - setzt Editor voraus, aber nicht unbedingt Compiler
- im Praktikum
  - Übersetzung, Inbetriebnahme, Test mit MS Visual C++
  - Vorführung und Erläuterung durch Sie
  - Abnahme und Testat durch mich (die innere Struktur interessiert mich mehr als die Funktion nach außen !)

Regeln zum Praktikum (1)

- Teilnahme an Praktikumsterminen ist Pflicht
  - Testat gibt es nur zum jeweiligen Termin (+ Folgetermin) !
- Bewertung: erfolgreich / nicht erfolgreich
  - keine Noten
  - Es zählt ob Sie Ihr eigenes Programm verstanden haben und erklären können. D.h. abschreiben hilft nur wenig.
  - Es gibt IMMER viele Lösungen für ein Problem.
- Erfolgreiche Teilnahme am Praktikum ist wichtig für Klausur:
  - Praktikum / Klausur geht im Verhältnis 1/3 in die Note ein
- Gruppeneinteilung & Termine
  - Im Anschluss an die Vorlesung

Regeln zum Praktikum (2)

- Gruppenarbeit: jeweils 2 StudentInnen an einem PC
  - erarbeiten gemeinsam ein Programm
  - keine Arbeitsteilung: jede(r) muß alles beherrschen
  - gemeinsame Vorbereitung ist empfehlenswert
- Modell "Fahrschule":
  Autofahren lernt man nicht als Beifahrer
  - "Fahrer" bedient Tastatur und Maus
  - "Beifahrer" beobachtet, denkt mit, berät
  - in jeder Übung werden die Rollen getauscht !
- Abschreiben und Kopieren ist schlecht!
  - Programmieren lernt man nur durch eigenes Üben

suchen Sie sich möglichst gleichwertige Partner, sonst halten Sie das nicht aus !

Termine Praktikum (3)

		Block2	Block3	Block4	Block5
Montag		10:15-11:45	12:00-13:30	14:15-1545	16:00-17:30
09.10.2007	x	V	NK		
16.10.2007	y	V			
23.10.2007	x	V	V		
30.10.2007	y	Keine Vorlesung/Praktikum			
06.11.2007	x	V	V	P1	P2
13.11.2007	y	V	P3	P4	
20.11.2007	x	V	V	P1	P2
27.11.2007	y	V	P3	P4	
04.12.2007	x	V	V	P1	P2
11.12.2007	y	V	P3	P4	
18.12.2007	x	V	V	P1	P2
25.12.2007		Weihnachten			
01.01.2008					
08.01.2008	y	V	P3	P4	
15.01.2008	x	V	V	P1	P2
22.01.2008	y	V	P3	P4	
29.01.2008	x	V	V		
05.02.2008			Klausur ?		
12.02.2008					

Hörsaalübung

- Termine
  - in lockerer Folge in die Vorlesung eingestreut
- Form
  - Programmentwürfe mit Papier und Bleistift
  - Zusammentragen der Ergebnisse an der Tafel
  - Programmentwicklung live über Beamer
- Inhalt
  - orientiert am jeweils aktuellen Abschnitt der Vorlesung
  - Vorbesprechung und Grobentwurf der Praktikumsaufgaben

Klausur

- am Ende des 1. Semesters

- schriftliche Prüfung im Hörsaal
  - Dauer – 1 Vorlesungsblock
  - Verständnisfragen
  - Programmanalyse

Persönliche Studienorganisation

- Selbstverantwortliches Lernen
  - **Fragen stellen**
  - Üben, üben, üben ...
  - Wissenslücken identifizieren und systematisch füllen
- Gruppenarbeit
  - muß allen Beteiligten nützen
  - nicht für Trittbrettfahrer
- Besuch der Vorlesungen
  - wird nicht kontrolliert
  - ist aber dringend zu empfehlen

... und außerdem:

- Sprachkenntnisse
  - Mindestanforderung Englisch
  - unbedingt pflegen !
  - auch Originalliteratur lesen
  - Fachbegriffe lernen
- heute Standard:
  - Arbeit mit Textsystem
  - nützlich: Maschinenschreiben im 10-Finger-System

hatten Sie schon Ihren Kurs an der Volkshochschule ?

Literaturempfehlungen

das Original vom Erfinder der Sprache;
mit vielen Hintergrundinformationen;
hervorragend, aber
für Anfänger sehr schwer verdaulich

- Bjarne Stroustrup
  Die C++ Programmiersprache

leichter verständliche Einführungen:

- Ulrich Breymann
  C++: Eine Einführung
- Stanley B. Lippman
  C++ Einführung und Leitfaden
- Helmuth Erlenkötter
  C++ Objektorientiertes Programmieren von Anfang an

Für Kenner einer Programmiersprache (z.B. C)

- **U. Kirch-Prinz, P. Prinz**
  OOP mit C++

Warum gerade C++ ?

- Programmieren kann mit jeder Sprache gelernt werden.
  - Wichtig für uns heute ist aber das verstehen der objektorientierten Denkweise
  - Gerade im (elektro-)technischen Bereich hat C durch seine HW-nähe einen gewissen Scharm. (Segen und Fluch)
- hohe Praxisrelevanz: C und C++ haben sich durchgesetzt
  - trotz vieler Haken und Ösen (die wir weitgehend vermeiden)
- neben Konzepten auch handwerkliche Fertigkeiten lernen
  - ... und dann gleich eine Sprache, die Sie später vermutlich noch brauchen können

Warum nicht gleich Java ?

- C++ hat zusätzliche Konzepte, die Informatiker kennen sollten
- Java ist eng verwandt mit C++
  - syntaktisch weitgehend identisch
  - einige Sprachkonstrukte sind weggelassen;
    dadurch einfacher, sicherer, aber auch weniger effizient
  - umfangreiche Standard-Bibliotheken, auch für Aufgaben, die sonst das Betriebssystem übernimmt
  - wer C++ beherrscht, wird mit Java keine Probleme haben
    (umgekehrt schon eher)

Das Erfolgsrezept von C

- C als Systemprogrammiersprache mit UNIX groß geworden
- nicht herstellerspezifisch
- anfangs kostenlose Weiterverbreitung im amerikanischen Hochschulbereich
  - kein Support, aber hacking the kernel für jeden
  - ganze Generation von Absolventen mit UNIX und C aufgewachsen
  - diese sorgte später in der Industrie für Verbreitung
- Implementierung
  - zunächst im technisch-wissenschaftlichen Bereich (DEC PDP-11)
  - dann im gesamten Workstation Sektor
  - mittlerweile auf praktisch allen Plattformen von Mikro bis Mainframe

... und C++

- C++ ist eine Obermenge von C
  - alle C++ Compiler können genauso gut C
- sanfter Übergang möglich
  - C > C++, prozedural > objektorientiert
- aber auch Etikettenschwindel möglich
  - man kann offiziell C++ programmieren
    - Objektorientierung ist "in"
  - und in Wirklichkeit ist es ganz konventionelles C
    - Umdenken ist nämlich gerade für alte Hasen manchmal schwierig

einerseits mit üblen Erbschaften belastet, andererseits gerade dadurch breite Akzeptanz

Zeittafel C und C++

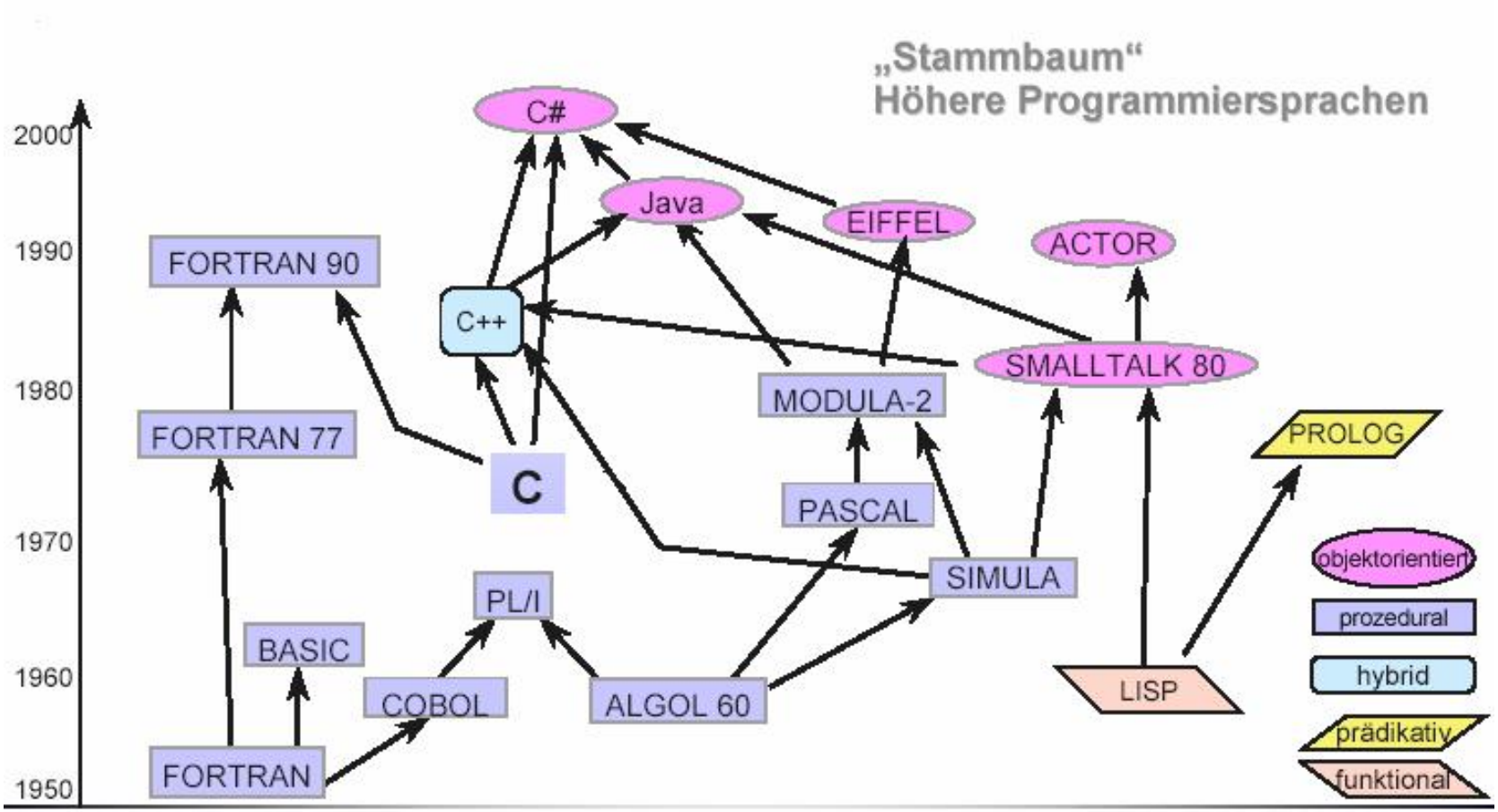
Algol und Pascal nicht in Ahnengalerie

- 1967 BCPL
- 1970 B
- 1972 erste C Implementierung

- 1978 Kernighan, Ritchie: The C Programming Language (First Edition)
- 1983 ANSI Komitee gegründet
- 1988 ANSI C Standard

- 1980 C mit Klassen implementiert bei AT&T
- 1983 Prägung des Namens: C++
- 1989 ANSI Komitee gegründet
- 1990 Ellis, Stroustrup: The Annotated C++ Reference Manual
- 1997 ISO / ANSI C++ Standard

Historischer Überblick



„Hello World“ Ein Klassiker unter den Programmen

- http://www.roesler-ac.de/wolfram/hello.htm

```
// Hello World in Java import java.io.*;
class HelloWorld
{
        static public void main( String args[] )
        {
                System.out.println( "Hello World!" );
        }
}
```

```
// Hello World in C++
#include <iostream.h>
main()
{
                cout << "Hello World!" << endl;
                return 0;
}
```

```
10 REM Hello World in BASIC
20 PRINT "Hello World!"
```