ETH Zürich
Chair of Software Engineering

Einführung in die Programmierung

Prof. Dr. Bertrand Meyer

Vorlesung 1: Übersicht & Willkommen

Ziele der Vorlesung

Nach erfolgreichem Abschluss dieser Vorlesung werden Sie:

- Die Schlüsselkonzepte des Programmierens kennen
- Viele verschiedene Programmierprobleme aus verschiedenen Bereichen lösen können
- Die grundsätzlichen Hardware- und Softwarewerkzeuge kennen
- Eine Programmiersprache beherrschen: Eiffel
- Die Grundkonzepte des Designs, der Implementierung und der Wartung von Softwaresystemen kennen ("software engineering").

Das Team der Assistenten

Nadia Polikarpova (Back Office)

Julian Tschannen (Koordinator)

Marco Piccioni

Mischael Schill

Andrei Rusakov

Yu Pei (Max)

Valentin Wüstholz

Oliver Probst

Daniel Schweizer

Florian Köhl

Sabina Schellenberg

Jascha Grübel

Back Office

Nadia Polikarpova

Koordinator

Julian Tschannen

Gruppe Ada Lovelace: Mischael Schill

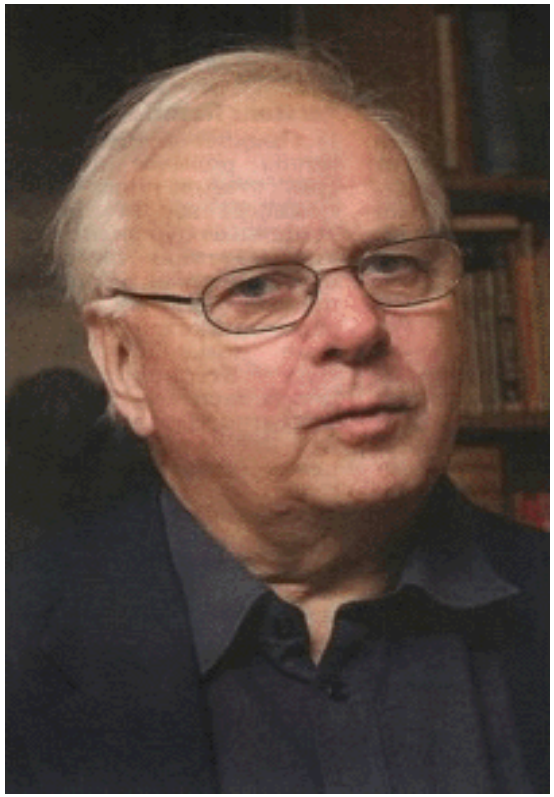
Mailingliste: se-info1-lovelace@lists.inf.ethz.ch

E-mail: mischael.schill@inf.ethz.ch
Büro: RZ J3
Telefon: 044 632 02 68
Sprache: Deutsch

Gruppe Kristen Nygaard: Julian Tschannen

Mailingliste: se-info1-nygaard@lists.inf.ethz.ch

E-mail: julian.tschannen@inf.ethz.ch
Büro: RZ J3
Telefon: 044 632 44 49
Sprache: Deutsch



Gruppe Edsger Dijkstra: Yu Pei (Max)

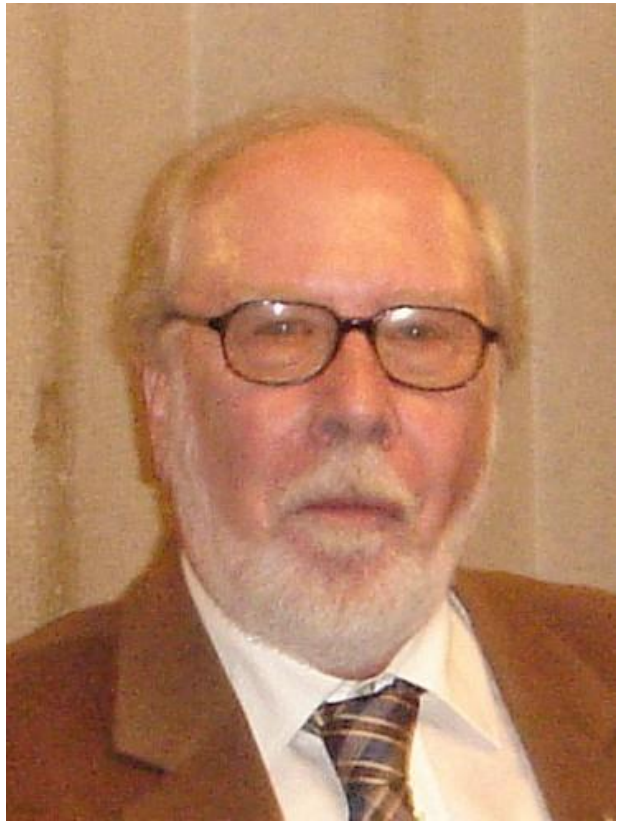
Mailingliste: se-info1-dijkstra@lists.inf.ethz.ch

E-mail: yu.pei@inf.ethz.ch
Büro: RZ J3
Telefon: 044 632 89 02
Sprache: English

Gruppe Niklaus Wirth: Marco Piccioni

Mailingliste: se-info1-wirth@lists.inf.ethz.ch

E-mail: marco.piccioni@inf.ethz.ch
Büro: RZ J9
Telefon: 044 632 65 32
Sprache: Englisch



Gruppe Tony Hoare: Valentin Wüstholz

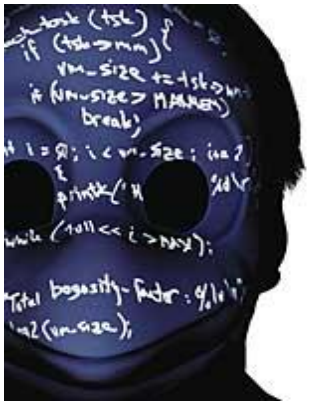
Mailingliste: se-info1-hoare@lists.inf.ethz.ch

E-mail: valentin.wuestholz@inf.ethz.ch
Büro: CAB F39
Telefon: 044 632 79 42
Sprache: Deutsch

Gruppe Linus Torvalds: Andrey Rusakov

Mailingliste: se-info1-torvalds@lists.inf.ethz.ch

E-mail: andrey.rusakov@inf.ethz.ch
Büro: RZ J3
Telefon: 044 632 44 68
Sprache: Englisch



Gruppe Adele Goldberg: Sabina Schellenberg

Mailingliste: se-info1-goldberg@lists.inf.ethz.ch

E-mail: sabinasc@student.ethz.ch

Sprache: Deutsch

Gruppe Donald Knuth: Florian Köhl

Mailingliste: se-info1-knuth@lists.inf.ethz.ch

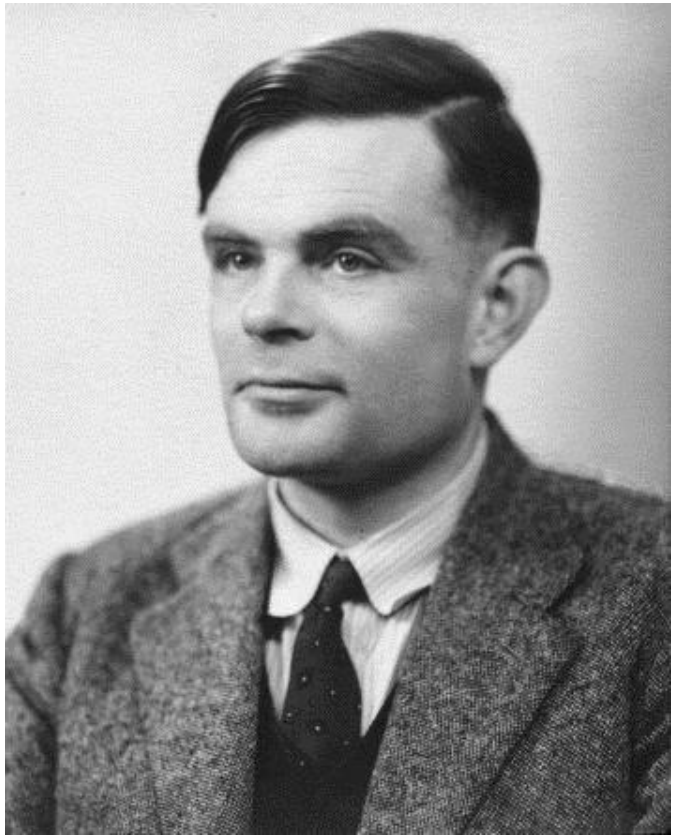
E-mail: koehlf@student.ethz.ch
Sprache: Deutsch

Gruppe Alan Turing: Daniel Schweizer

Mailingliste: se-info1-turing@lists.inf.ethz.ch

E-mail: daschwei@student.ethz.ch

Sprache: Deutsch



Gruppe Barbara Liskov: Jascha Grübel

Mailingliste: se-info1-liskov@lists.inf.ethz.ch

E-mail: jgruebel@student.ethz.ch
Sprache: Deutsch

Gruppe John von Neumann: Oliver Probst

Mailingliste: se-info1-neumann@lists.inf.ethz.ch

E-mail: oprobst@student.ethz.ch
Sprache: Deutsch

Über mich

An der ETH seit Ende 2001, Professor für Software Engineering

Grossteil meiner Karriere in der Industrie, zuletzt bei *Eiffel Software* in Santa Barbara, Kalifornien, seit 1985. Heute "Chief Architect"

Assoc. Prof. an der University of California, Santa Barbara in den 80ern

Autor mehrerer Bücher, insbesondere *Object-Oriented Software Construction* (2. Ausgabe: 1997)

Forschungsschwerpunkte: Software Engineering, Programmiersprachen, Objekt-orientierte Programmierung, Nebenläufige Programmierung (*concurrency*), Programbeweise, Testen, Entwicklungsumgebungen, Persistenz

Kontaktdaten:

- E-mail: Bertrand.Meyer@inf.ethz.ch, Büro: RZ J22
- Sekretariat: Claudia Günthart, 044 632 83 46
  Claudia.Guenthart@inf.ethz.ch, Büro: RZ J7

Sprechstunden: Mittwochs während des Semesters, kontaktieren Sie Frau Günthart

Übungsgruppen

Sie füllen eine Umfrage aus. Den Link dazu finden sie auf der Webseite, oder wird Ihnen morgen per E-Mail geschickt. Sie wählen zwischen

- Drei Niveaus
- Zwei Sprachen

Tragen Sie sich bis morgen auf mystudies.ethz.ch für den Kurs ein, um das Email zu erhalten.

Falls Sie gute Gründe für einen Gruppenwechsel haben: fragen Sie Julian

Sprachen

Unsere Assistenten sprechen verschiedene Sprachen:

- Deutsch
- Englisch
- Italienisch
- Chinesisch
- Russisch
- …

Die Übungsgruppen werden auf deutsch oder auf englisch gehalten.

Sprache

Die Sprache für diese Vorlesung ist deutsch

Geschwindigkeit auf Deutsch

Der Turbo-Knopf (nicht noch gefunden)

Auf der positiven Seite

30 Sekunden

Stundenplan

Vorlesungen:

- Montags, 13:15 – 15:00, HG F1
- Dienstags, 8:15 – 10:00, HG F1

Übungsstunden:

- 11 Gruppen
  - Mittwoch, 8:15 – 10:00, in verschiedenen Räumen
  - Mittwoch, 15:15 – 17:00, in verschiedenen Räumen

Vorlesungsausfall und Ersatzvorlesung

Die Vorlesung nächste Woche am Dienstag 25. September fällt aus.

Dafür wird diese Woche am **Freitag 21. September** von **13:15 bis 15:00** eine Ersatzvorlesung im **HG E7** gehalten.

(diese Information finden Sie auch auf der Webseite)

Webseite der Vorlesung

Webseite:

http://se.ethz.ch/courses/2012b_fall/eprog/

→ Zweimal wöchentlich anschauen

Deutsche Version ist vorhanden, aber die englische ist meistens aktueller

Vorlesungsunterlagen:

- Folien der Vorlesung
- Buch: *Touch of Class*

  Siehe nächste Folie

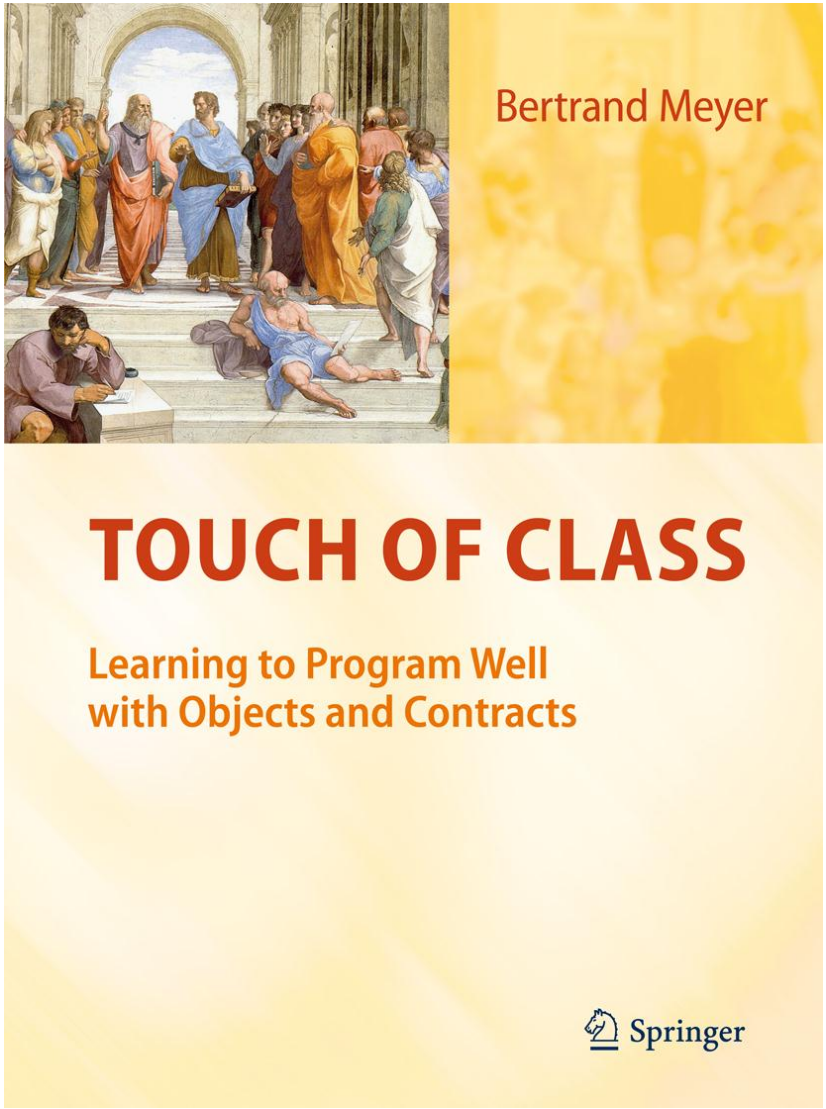
Übungsunterlagen:

- Übungen
- Musterlösungen

Auch vorhanden:

Videos der Vorlesung!

Das Buch zur Vorlesung



Online-Version des Buches

Möglich aus dem Netz der ETH

URL: siehe Vorlesungswebseite

Buchverkauf: nächsten Montag (in der Pause)

ETH-Zentrum
MM B 96
8092 Zürich

Öffnungszeiten

Montag – Donnerstag
9:30 – 16:30 Uhr
Freitag
9:30 – 15:30 Uhr

In der Buchhandlung akzeptieren wir
Bargeld, Maestro, Postcard, Visa und Eurocard/Mastercard

Meyer

Touch of Class

Fr. 58.00

Büchertisch vor dem Hörsaal
Nur Barbezahlung möglich!

Polybuchhandlung
ETH Zentrum MM B96
Öffnungszeiten: Mo-Do 9.30 – 16.30, Fr. 9.30-15.30
Internet: www.books.ethz.ch, Email: shop@books.ethz.ch

Elektronische Foren

Diskussionsforen:
Hilfeforum für die gesamte Vorlesung:
http://forum.vis.ethz.ch/

Mailingliste für jede Übungsgruppe

Ratschläge und Regeln:

- Benutzen Sie das VIS-Forum und die Mailinglisten! Programmieren zu lernen ist schwierig: Nutzen Sie jede Hilfe, die Ihnen angeboten wird.
- Es gibt keinen Grund, schüchtern zu sein. Es gibt keine dummen Fragen.
- Kritik ist willkommen, seien Sie aber immer freundlich und halten Sie sich an die Etiquette.

Falls Sie einen Laptop brauchen…

Das NEPTUN-Programm der ETH verkauft Laptops zu guten Preisen

Thinkpad (Lenovo, ex-IBM), HP oder Apple

Sie wählen das Betriebssysstem: Windows, Linux, MacOS

Zeitlich begrenzter Verkauf: siehe www.neptun.ethz.ch

Übungen

Die Übungen sind ein wichtiger Bestandteil der Vorlesung

- Eine Übung pro Woche (ca 10 insgesamt)
- Zwei Präsenzübungen

Für Ihre Übungsabgabe sollten Sie:

- nachweisen, dass Sie die Aufgaben zu lösen versucht haben.
- die Umfrage ausfüllen.

Absenzen wegen Militärdienst oder Krankheit: kontaktieren Sie Ihren Assistenten.

Benotung

Die Grundregeln sind von der ETH diktiert, die Feinheiten für das Testat von uns bestimmt:

Die Note beruht nur auf der Leistung in der Prüfung vom kommenden September

Aber: damit Sie zur Prüfung zugelassen werden müssen Sie das Testat erhalten. Das bedeutet Sie müssen

- Alle wöchentlichen Übungen ausser einer abgeben
- An beiden Präsenzübungen teilnehmen

Damit diese akzeptiert werden, müssen Sie:

- nachweisen, dass Sie die Aufgaben zu lösen versucht haben.
- die Umfrage ausfüllen.

Absenzen wegen Militärdienst oder Krankheit: kontaktieren Sie Ihren Assistenten.

Repetenten: das Testat vom letzten Jahr ist gültig, aber wir empfehlen Ihnen die Übungen trotzdem zu machen.

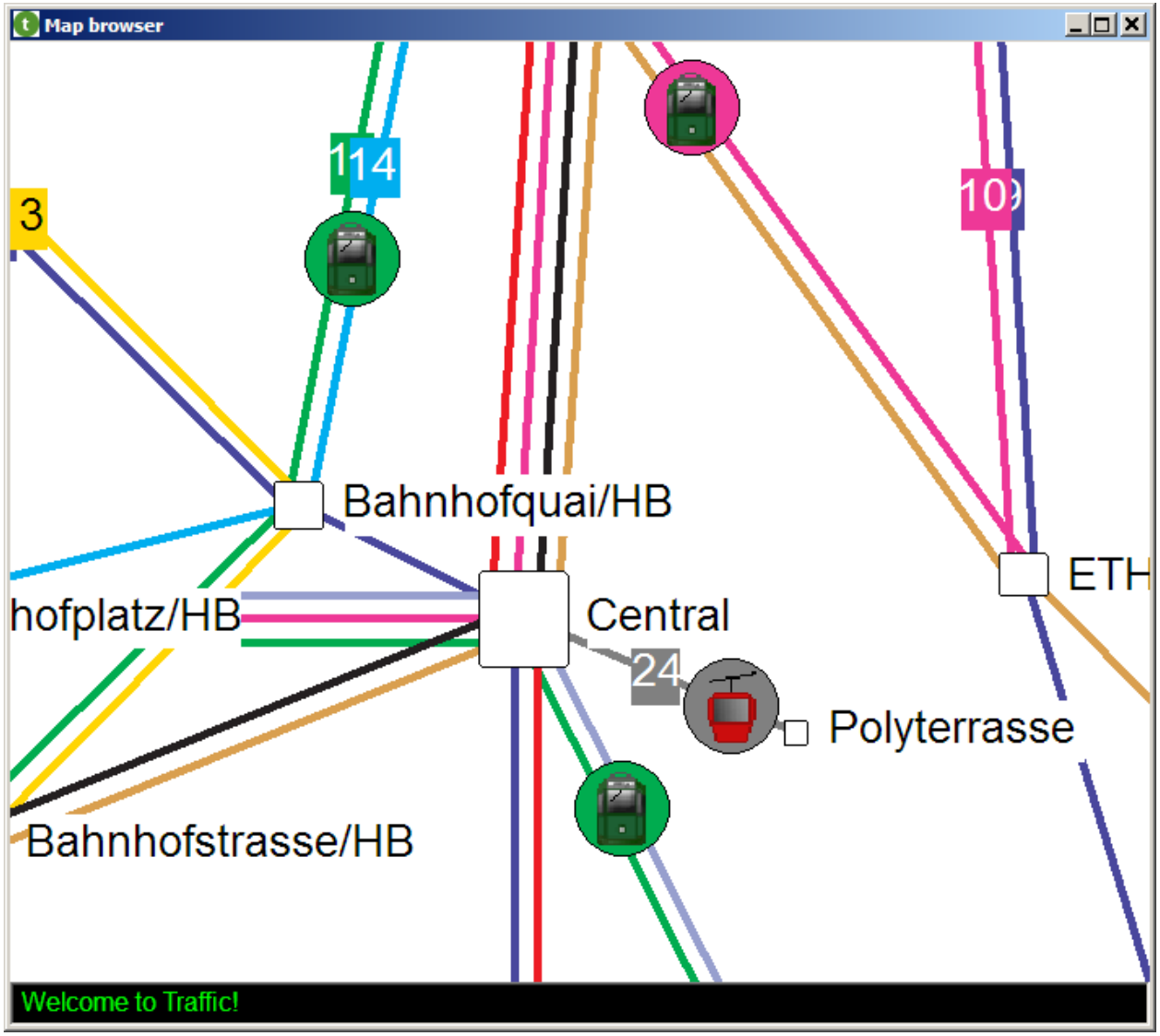
Die Software

Die Übungen bauen auf der Bibliothek Traffic auf

Anwendungsgebiet: öffentlicher Verkehr in einer Stadt (benutzt Zürich als Beispiel)

Übung 1 (auf der Webseite) führt Sie durch die Installation von EiffelStudio und zeigt Traffic

Entdecken Sie Traffic



Warnung

Natürlich ist nicht alles perfekt.

Traffic beinhaltet wahrscheinlich Fehler ("bugs"), und das Buch wahrscheinlich auch.

(Fehlerliste: http://touch.ethz.ch -> *Errata*)

Aber:

- Wir versuchen, die Fehler so schnell wie möglich zu korrigieren.
- Schieben Sie beim Ausprobieren die Schuld jeweils nicht zuerst der Software in die Schuhe. Vielleicht folgt sie bloss Ihren Anweisungen.

Weshalb diese Lehrmethode?

Mit anderen Lehrmethoden würden Sie nur mit kleinen selber geschriebenen Programmen arbeiten

Wir verwenden ein vorgegebenes Softwaresystem; benutzen Sie dieses als Beispiel und Inspiration

Sie benutzen die Software durch ihre abstrakten Schnittstellen (auch bekannt als **Verträge** (contracts))

Sie verwandeln sich vom Konsumenten zum Produzenten: outside-in

Traffic ist grafisch und macht Spass!

Im besten Fall verstehen Sie am Ende die gesamte Software.

Dann können Sie auch neues hinzufügen

Ratschläge

- Besuchen Sie alle Vorlesungen
- Lesen Sie die Unterlagen — das Buch und die Folien — jeweils vor den Vorlesungen
  (Bem.: Folien werden häufig nach der Vorlesung aktualisiert)
- Nehmen Sie eine Druckversion der Folien mit und machen Sie sich Notizen
- Besuchen Sie alle Übungsstunden
- Machen Sie alle Übungen
  (Sie brauchen sie für das "Testat")
- Falls Sie etwas nicht verstehen, fragen Sie nach
  (es gibt keine dummen Fragen)

Programmiererfahrung

Falls Sie bereits programmiert haben, nutzen Sie diesen Vorteil, aber seien Sie auch offen für eine neue Sichtweise; erkunden Sie Traffic

Falls Sie noch nie programmiert haben, keine Angst; es kann anfangs schwierig sein, aber Sie werden es schaffen.

Mathematisches Wissen ist genauso nützlich wie Programmiererfahrung

Vorkenntnisse eines Informatikstudenten im ersten Semester (2003-2008)

	2008	Frühere Jahre
Erfahrung mit Computern	≤ 2 Jahre: 1%	(0%, 0%, 0%, 1%)
	2 bis 4 Jahre: 3%	(1%, 3%, 4%, 1%, 6%)
	5 bis 9 Jahre: 34%	(39%, 35%, 48%, 35%)
	≥ 10 Jahre: 62%	(61%, 62%, 48%, 63%)
Programmier-erfahrung	Keine: 18%	(12%, 19%, 18%, 14%)
	Keine OOP: 22%	(20%, 26%, 33%, 38%)
	≥ 100 Klassen: 17%	(8%, 11%, 15%, 10%, 5%)

Die Industrie der reinen Ideen

Ziele der Vorlesung

Nach erfolgreichem Abschluss dieser Vorlesung werden Sie:

- Die Schlüsselkonzepte des Programmierens kennen
- Viele verschiedene Programmierprobleme aus verschiedenen Bereichen lösen können
- Die grundsätzlichen Hardware- und Softwarewerkzeuge kennen
- Eine Programmiersprache beherrschen: Eiffel
- Die Grundkonzepte des Designs, der Implementierung und der Wartung von Softwaresystemen kennen ("software engineering").

Themen

- Was ist Software?
- Objekte & Programme
- Schnittstellen und das Klassenkonzept
- Logik und Verträge (contracts)
- Das Laufzeitmodell: Objekterzeugung, Referenzen
- Syntaxbeschreibung
- Kontrollstrukturen
- Vererbung
- Generik
- Rekursion
- Datenstrukturen
- Ereignisgesteuerte Programmierung & Agents
- Topologisches Sortieren
- Einführung ins Software Engineering

Softwareingenieure bauen Maschinen

Diese Maschinen kann man nicht berühren, treten oder fallen lassen: sie sind immateriell

Aber es sind trotzdem Maschinen

Wir nennen sie Programme oder Systeme

Um ein Programm auszuführen, benötigt man eine materielle Maschine: einen Computer

Computer und technische Geräte: Hardware

Programme und der damit verbundene intellektuelle Wert: Software

Software, wohin man auch schaut

Banken: verwaltet Millionen von Konti

Handel: entscheidet über Kauf und Verkauf

Verkehr: kontrolliert Züge, überwacht Flugzeuge...

- Einige Autos beinhaltet Software mit Millionen von Zeilen Programmcode

Reisen: Flug-, Zug-, und Hotelbuchungen

Kommunikation: Telefonie, Internet, ...

Behörden: verwaltet Steuern, überwacht Gesetze...

Gesundheitswesen: verwaltet Krankenakten, überwacht Geräte

Unterricht

Unterhaltung

Information

usw.

Computer überall…

Banken

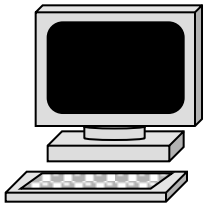
Flugzeuge, Autos…

Waschmaschinen

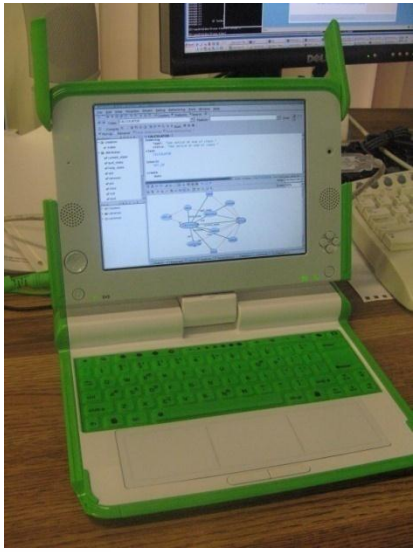
Mobiltelefone (> 70% des Werts)

Drucker

Morgen: Ihr T-Shirt…



Computer in allen Grössen, Farben und Formen



Computer

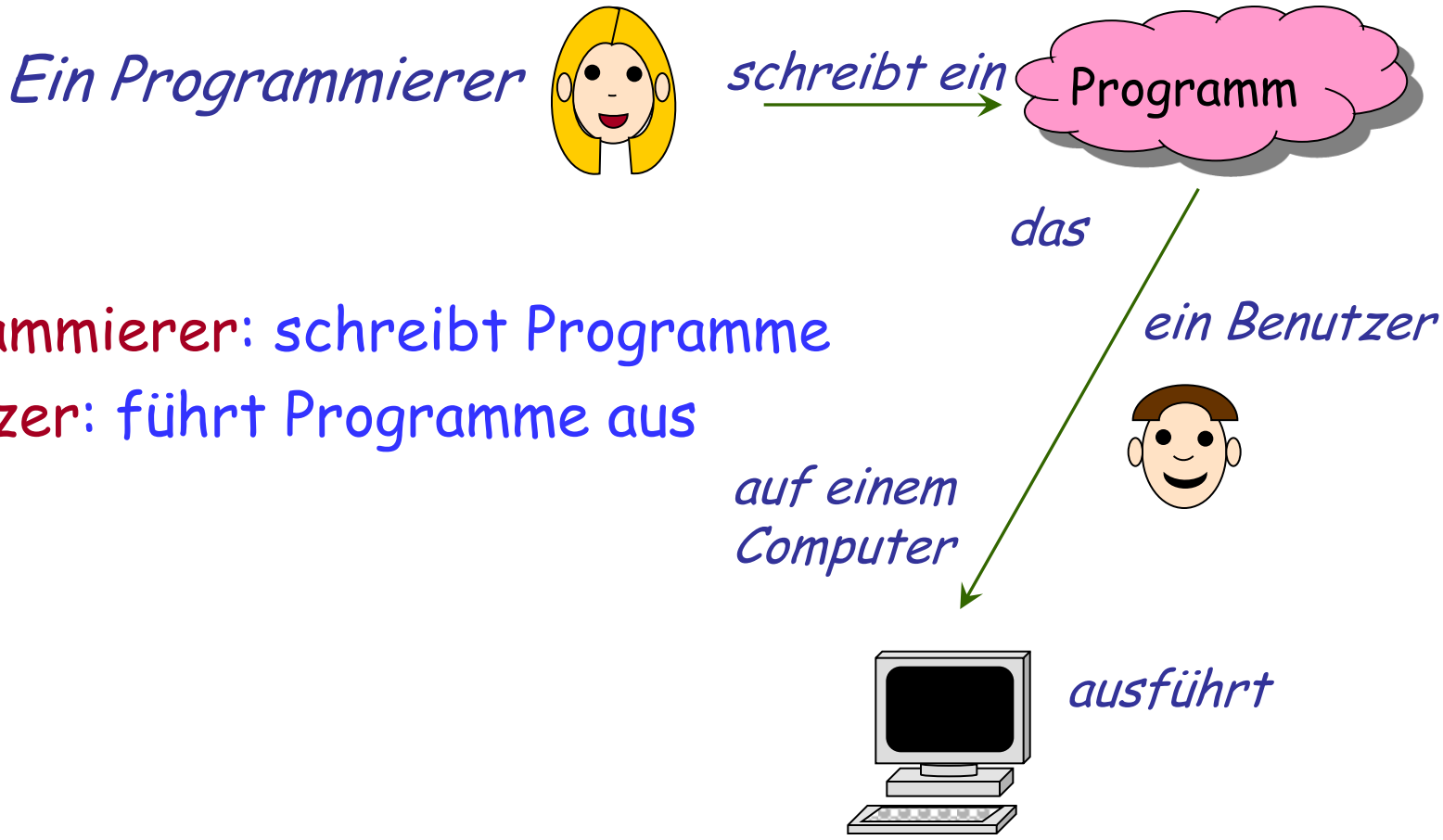
Computer sind universelle Maschinen. Sie führen das Programm aus, das Ihnen gegeben wird.

Ihre Vorstellungskraft ist die einzige Grenze

Die guten Nachrichten:

- Ihr Computer tut genau das, was in Ihrem Programm steht

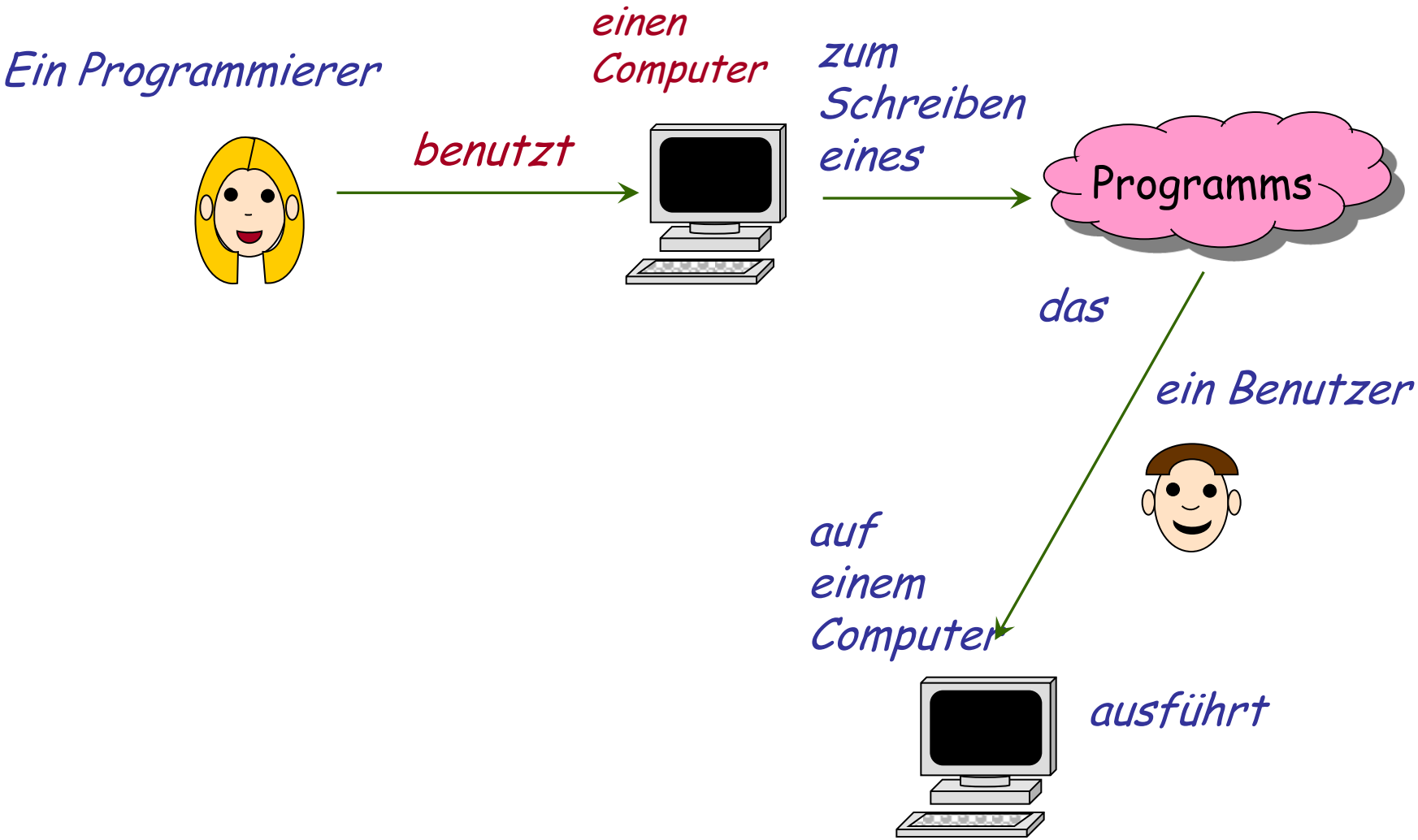
Programme erstellen und ausführen



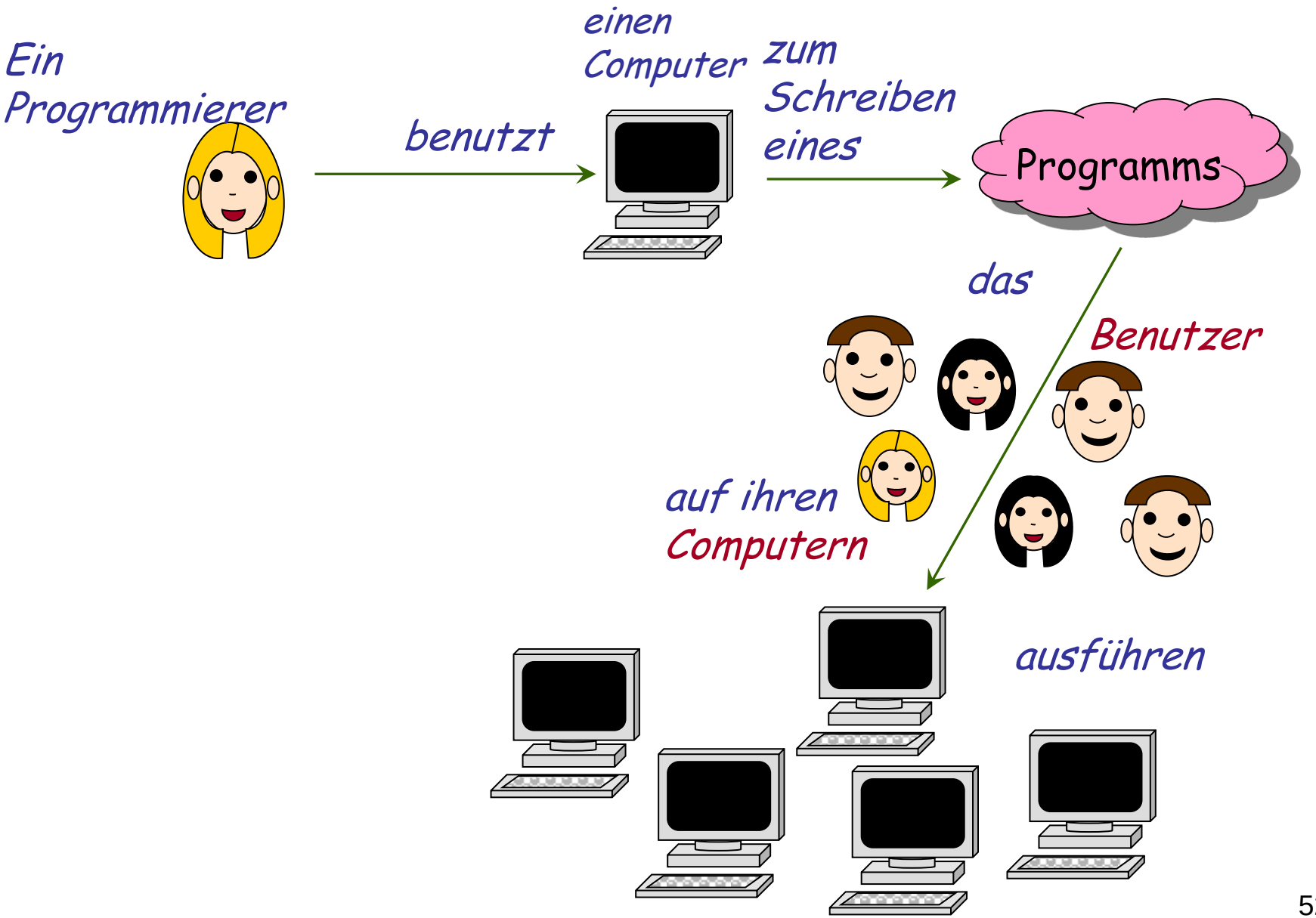
Programmierer: schreibt Programme

Benutzer: führt Programme aus

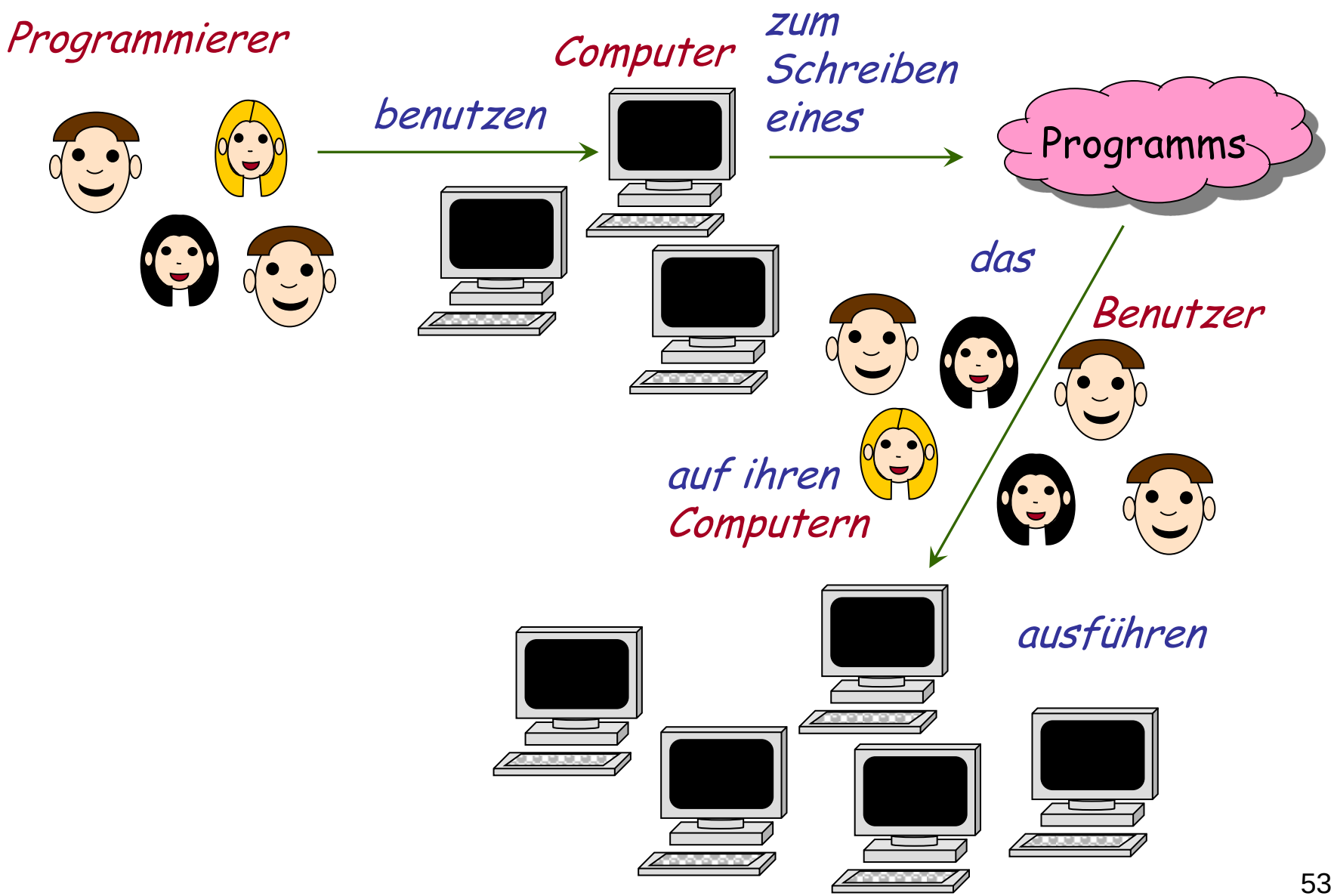
Programme erstellen und ausführen



Programme erstellen und ausführen



Programme erstellen und ausführen



Computer

Computers sind universelle Maschinen. Sie führen das Programm aus, das Ihnen gegeben wird.

Ihre Vorstellungskraft ist die einzige Grenze

Die guten Nachrichten:

- Ihr Computer tut genau das, was in Ihrem Programm steht
- Er tut es sehr schnell

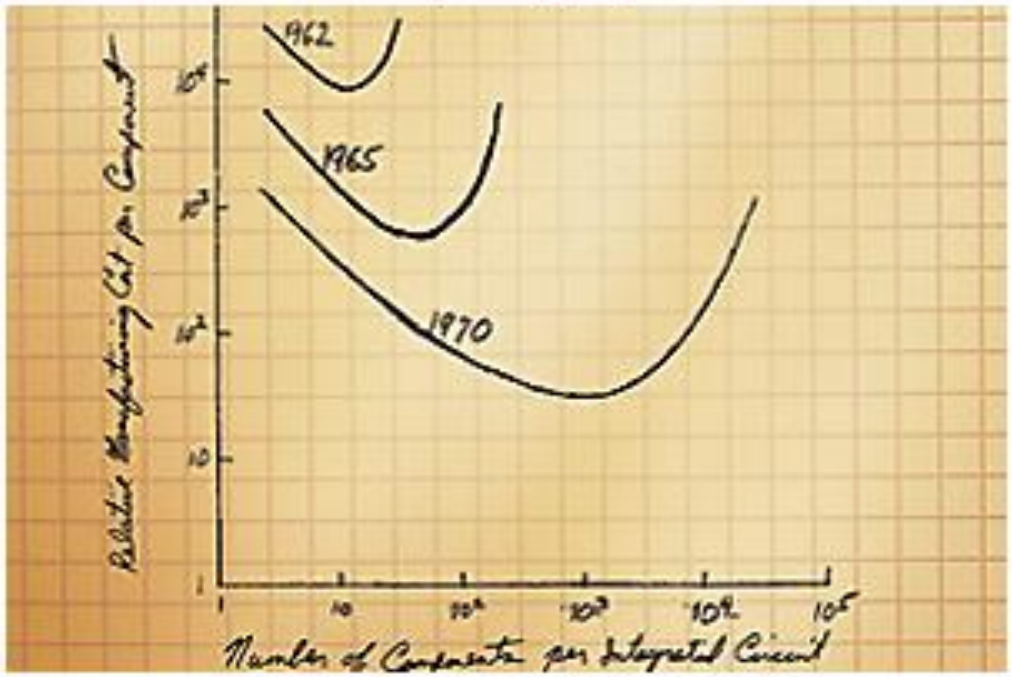
Moore's "Gesetz"

Etwa alle 18 Monate: Verdopplung der Rechenleistung bei gleichbleibendem Preis

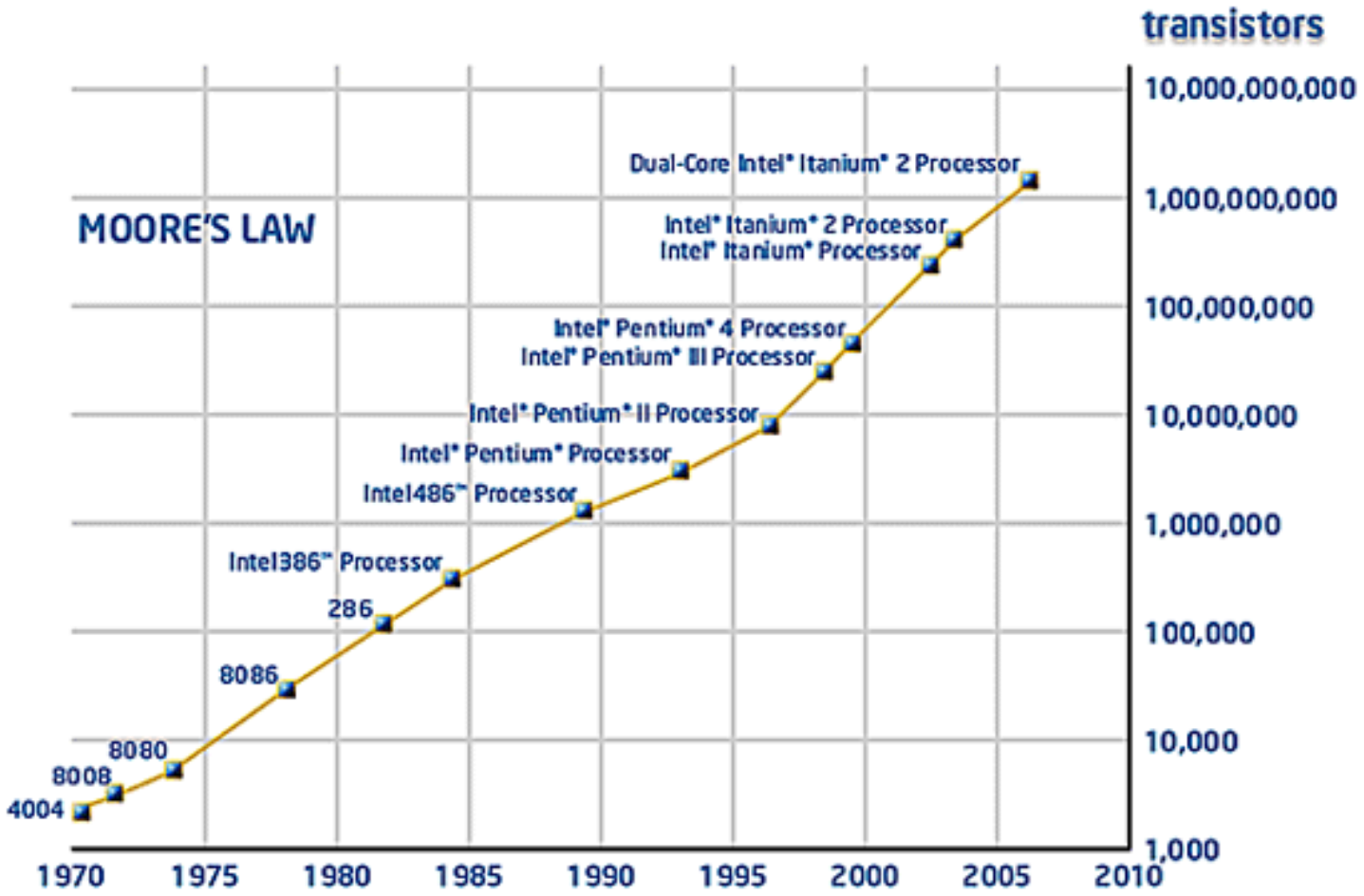
(Ist das die Aussage von Moore's Gesetz?)

(Nein: Verdopplung der Anzahl Transistoren)

Gordon Moore's original graph from 1965

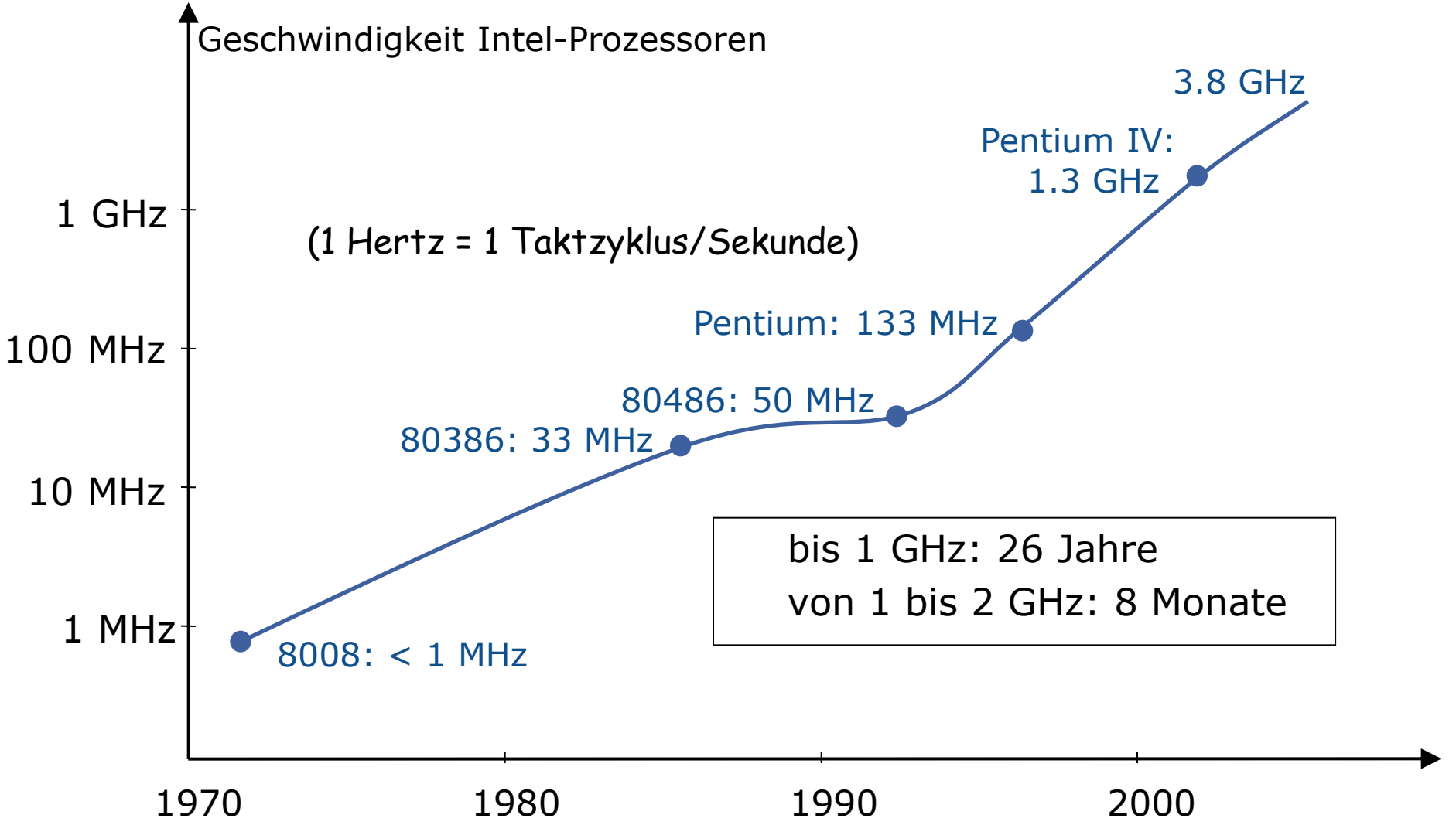


Moore's Gesetz (Quelle: Intel)

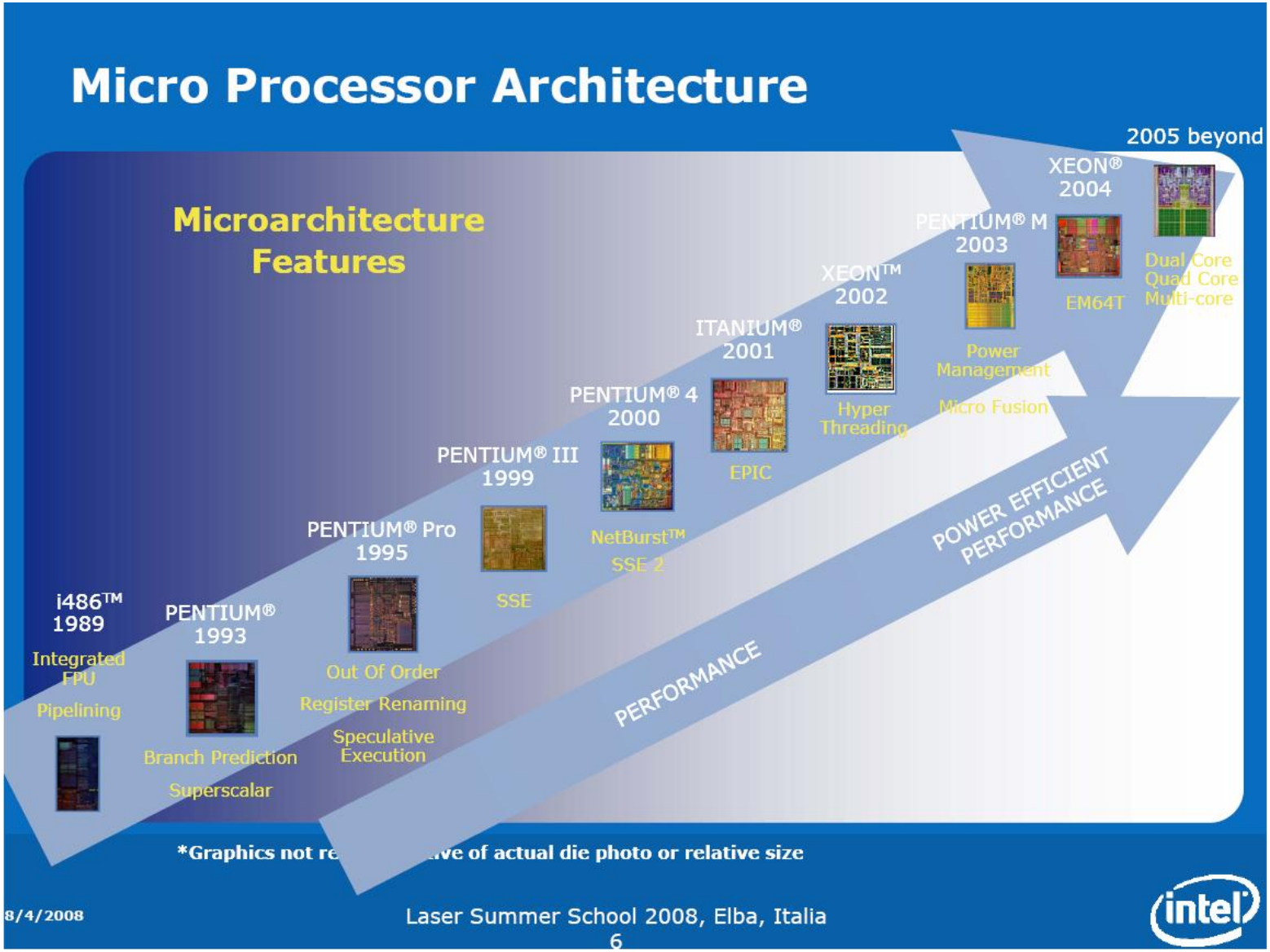


Moore's "Gesetz"

Etwa alle 18 Monate: Verdopplung der Rechenleistung bei gleichbleibendem Preis



Microprozessoren (Quelle: Intel)



Legenden und Entschuldigungen

"Computer sind intelligent"

Tatsache: Computer sind weder intelligent noch dumm. Sie führen Programme aus, die von Menschen entwickelt wurden. Diese Programme spiegeln die Intelligenz ihrer Autoren wieder.

Die Grundoperationen eines Computers sind elementar (speichern eines Wertes, addieren zweier Zahlen…).

"Der Computer ist abgestürzt"

"Der Computer erlaubt das nicht"

"Der Computer hat Ihre Datei verloren"

"Der Computer hat Ihre Datei kaputt gemacht"

Computer machen keine Fehler *....

- Programme machen auch keine Fehler
- Programmierer machen die Fehler

*Hardware kann zwar defekt sein, aber das ist viel seltener der Fall als Fehler in der Software

Computer

Computers sind universelle Maschinen. Sie führen das Programm aus, das Ihnen gegeben wird.

Die einzige Grenze ist Ihre Vorstellungskraft und Ihre Sorgfalt

Die guten Nachrichten:

- Ihr Computer tut genau das, was in Ihrem Programm steht
- Er tut es sehr schnell

Die schlechten Nachrichten:

- Ihr Computer tut genau das, was in Ihrem Programm steht
- Er tut es sehr schnell

" To err is human, but to really mess things up takes a computer"

Der “Blue Screen Of Death”

Software zu schreiben ist schwierig

Programme "stürzen ab"

Wenn Programme nicht abstürzen, bedeutet das nicht umbedingt, dass sie richtig funktionieren

Fehlerhafte Programme haben Menschen umgebracht, z.B. medizinische Geräte

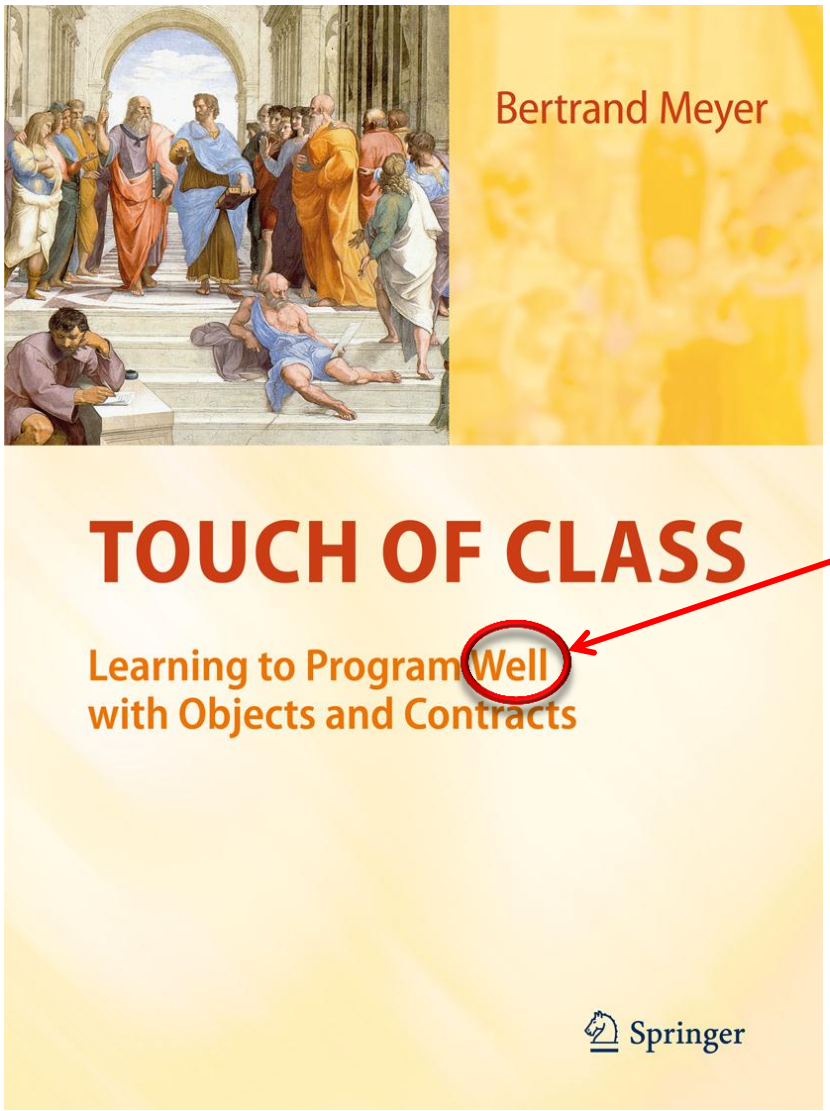
Ariane 5 Rakete, 1996: $10 Mia. verloren, wegen einem einfachen Programmfehler

US Patriot-Raketen, 1990: 28 Soldaten getötet, wegen einem numerischen Fehler

Programmierer sind für die korrekte Funktionsweise ihrer Programme verantwortlich

Der Zweck dieser Vorlesung ist es nicht nur Ihnen programmieren beizubringen, sondern dass Sie gut programmieren lernen.

Learning to program *well*



Ariane 5

http://www.youtube.com/watch?v=kYUrqdUyEpI

Programm-Fehler: sehen

- Jean-Marc Jézéquel & Bertrand Meyer: *Design by Contract: The Lessons of Ariane*, in *Computer* (IEEE), vol. 30, no. 1, January 1997, pages 129-130, archive.eiffel.com/doc/manuals/technology/contract/ariane/.

Persönliche Ratschläge

Erfolgreich studieren (insbesondere an der ETH):

- Sie bestimmen selbst, was Sie wann tun
- Nutzen Sie die Möglichkeiten der ETH
  - Talks gehalten von Forschern von anderen Unis
  - Konferenzen
  - Bibliotheken
  - Experimente
  - Projekte
- Sprechen Sie mit Professoren und Assistenten
- Lesen Sie die Webseiten des Departements und des Chair of Software Engineering
- Halten Sie Ausschau nach Vorlesungen mit Projekten und anderen Möglichkeiten, individuell zu arbeiten

Noch mehr persönliche Ratschläge

- Besuchen Sie die Vorlesungen
- Besuchen Sie die Übungsstunden
- Lesen und drucken Sie die Folien vor der Stunde
- Machen Sie sich Notizen
- Schliessen Sie sich zu Studiengruppen zusammen
- Besuchen Sie auch Vorlesungen zu informatikfremden Themen, vor allem während den ersten zwei Jahren
- Bereiten Sie sich früh auf die Prüfung vor

- Bewahren Sie eine kritische, forschende Einstellung

Was Computer tun

Speichern und wieder abrufen

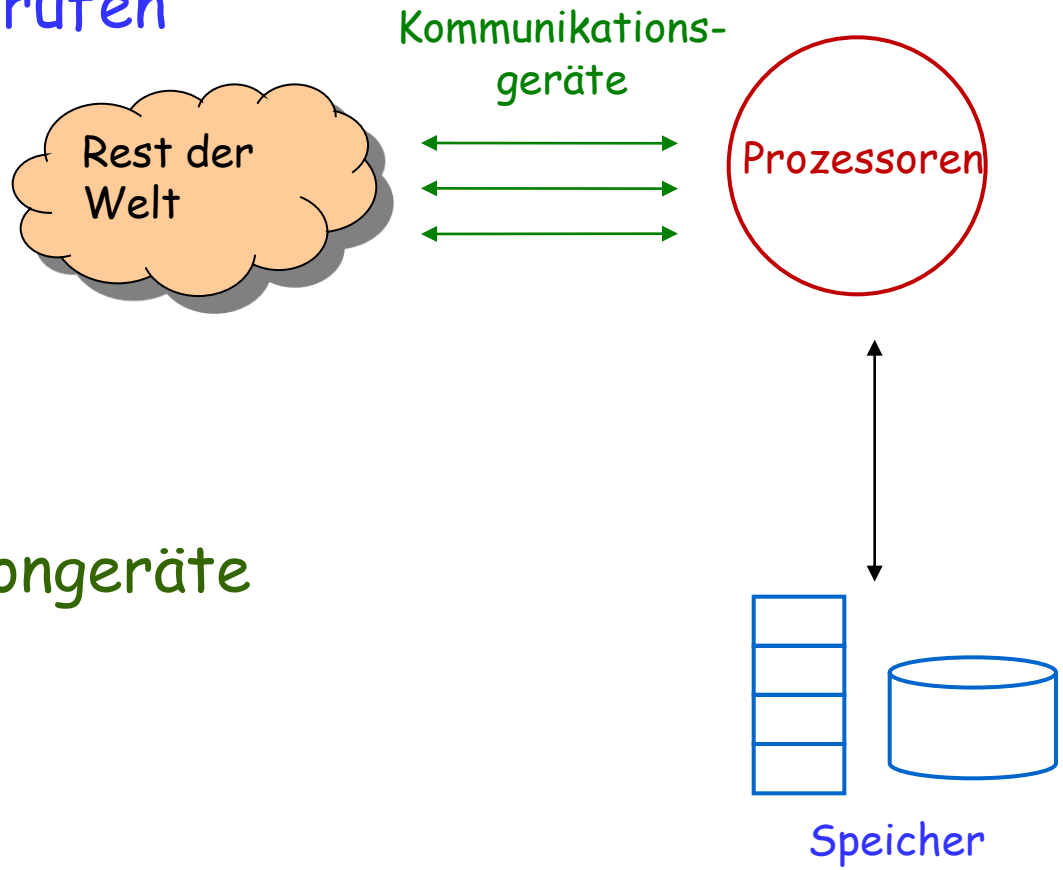
- Speicher

Operationen ausführen

- Prozessoren

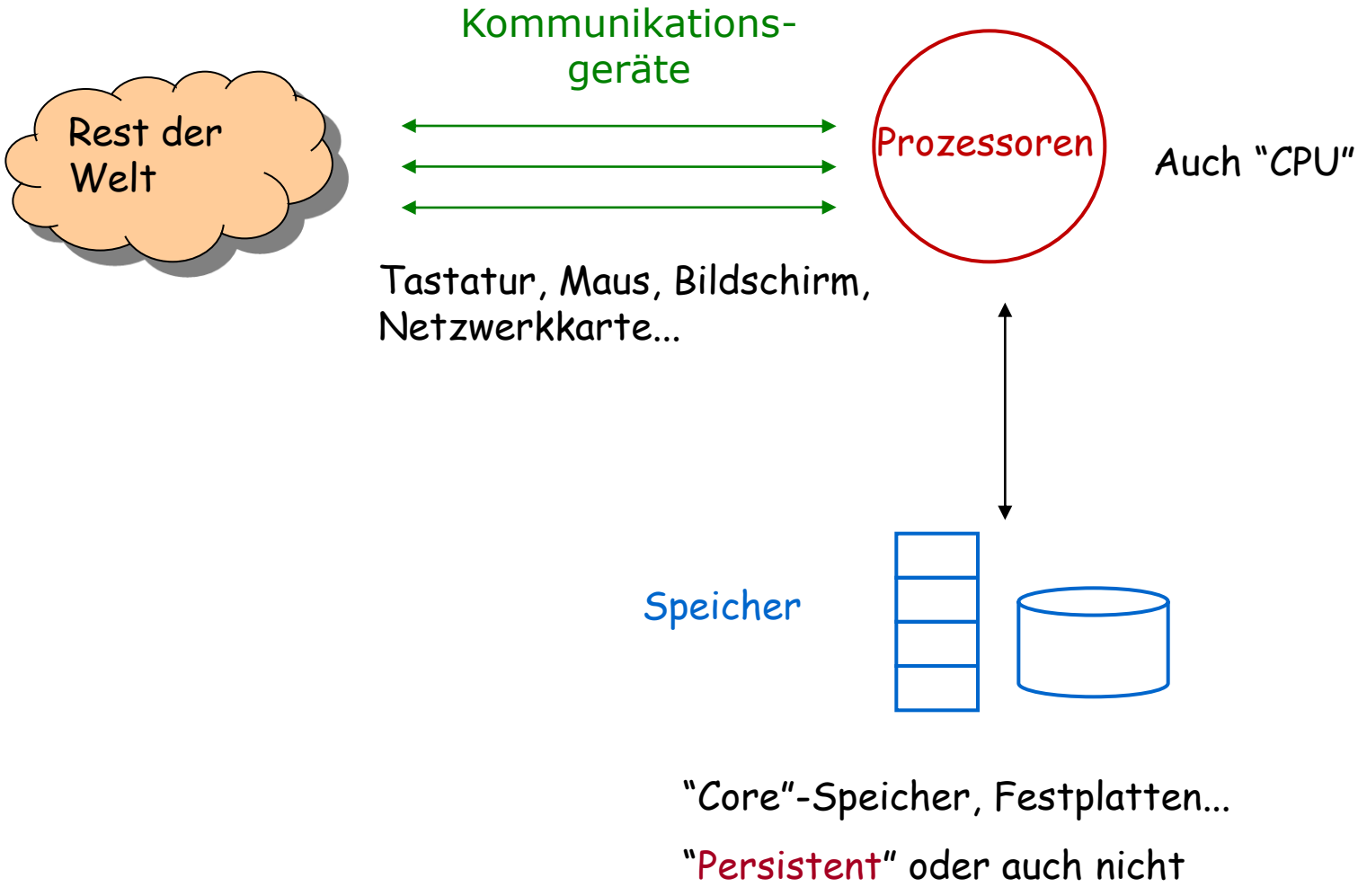
Kommunikation

- Kommunikationgeräte



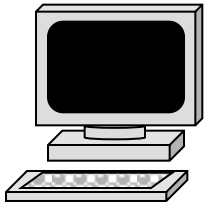
Speicher, Prozessoren und Kommunikationsgeräte gehören zur Hardware

Genereller Aufbau



Computer

Computer sind universelle Maschinen. Sie führen das Programm aus, das Ihnen gegeben wird.



\+

Programm

=

Spezialisierte Maschine

(Universelle Maschine)

Software

Information und Daten

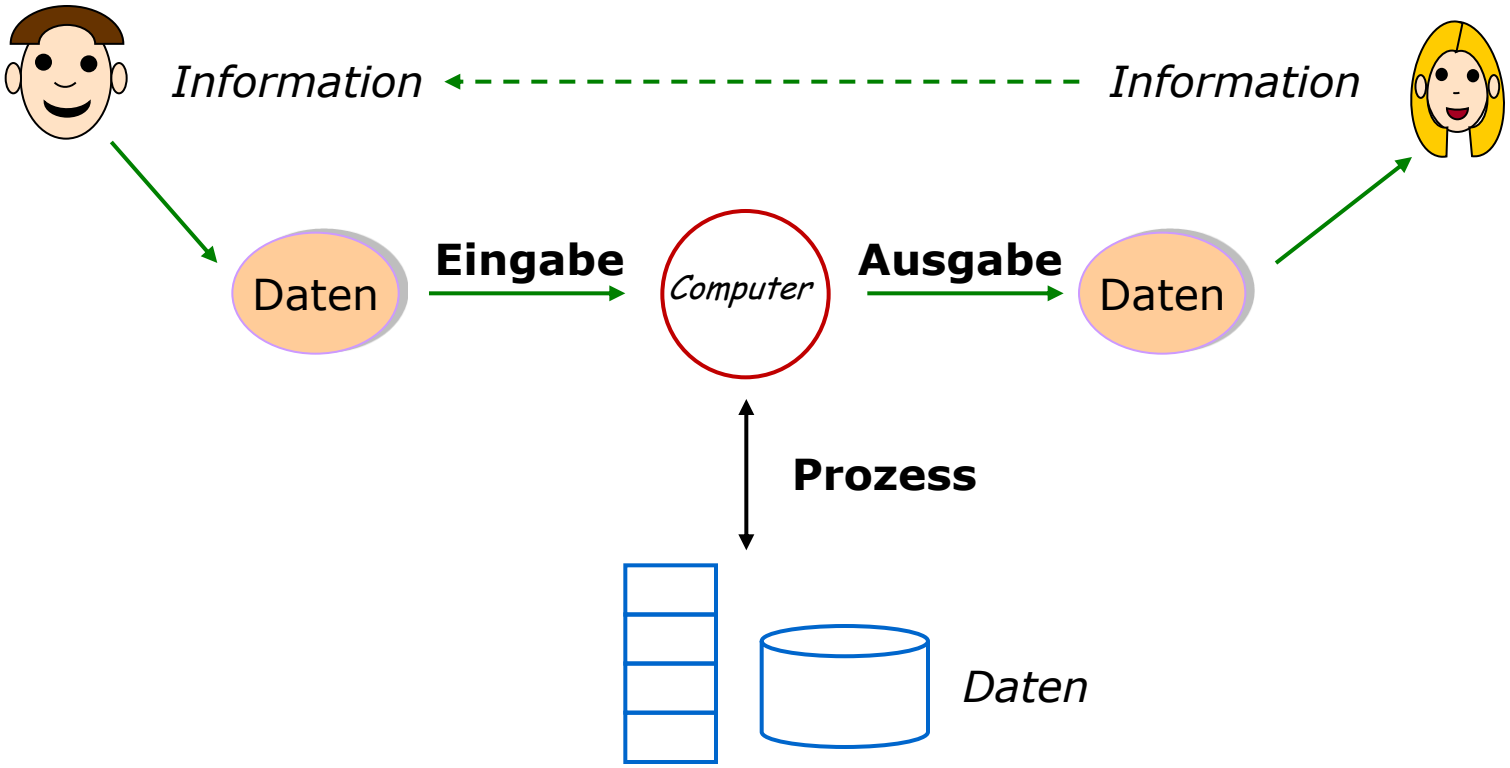
Information ist, was Sie wollen, z.B. Text oder Musik

Daten sind Informatinen, die für den Computer kodiert sind, z.B. im MP3-Audioformat

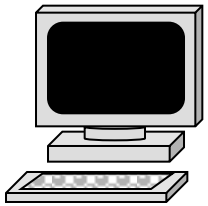
- Daten: Menge von Symbolen auf dem Computer gespeichert
- Informationen: Interpretation der Daten für menschliche Zwecke

Verarbeitung von Daten und Information

Daten werden im Speicher gehalten
Eingabegeräte produzieren Daten aus Informationen
Ausgabegeräte produzieren Informationen aus Daten



Wo ist das Programm?



\+

Programm

=

Spezialisierte Maschine

(Universelle Maschine)

Wo befindet sich ein Programm?

Speicherprogrammierte Computer: Das Programm befindet sich im Speicher

- "Executable data"

Ein Programm kann im Speicher in verschiedenen Formen vorhanden sein:

- Quellcode: von Menschen lesbar (Programmiersprache)
- Maschinencode: vom Computer ausführbar

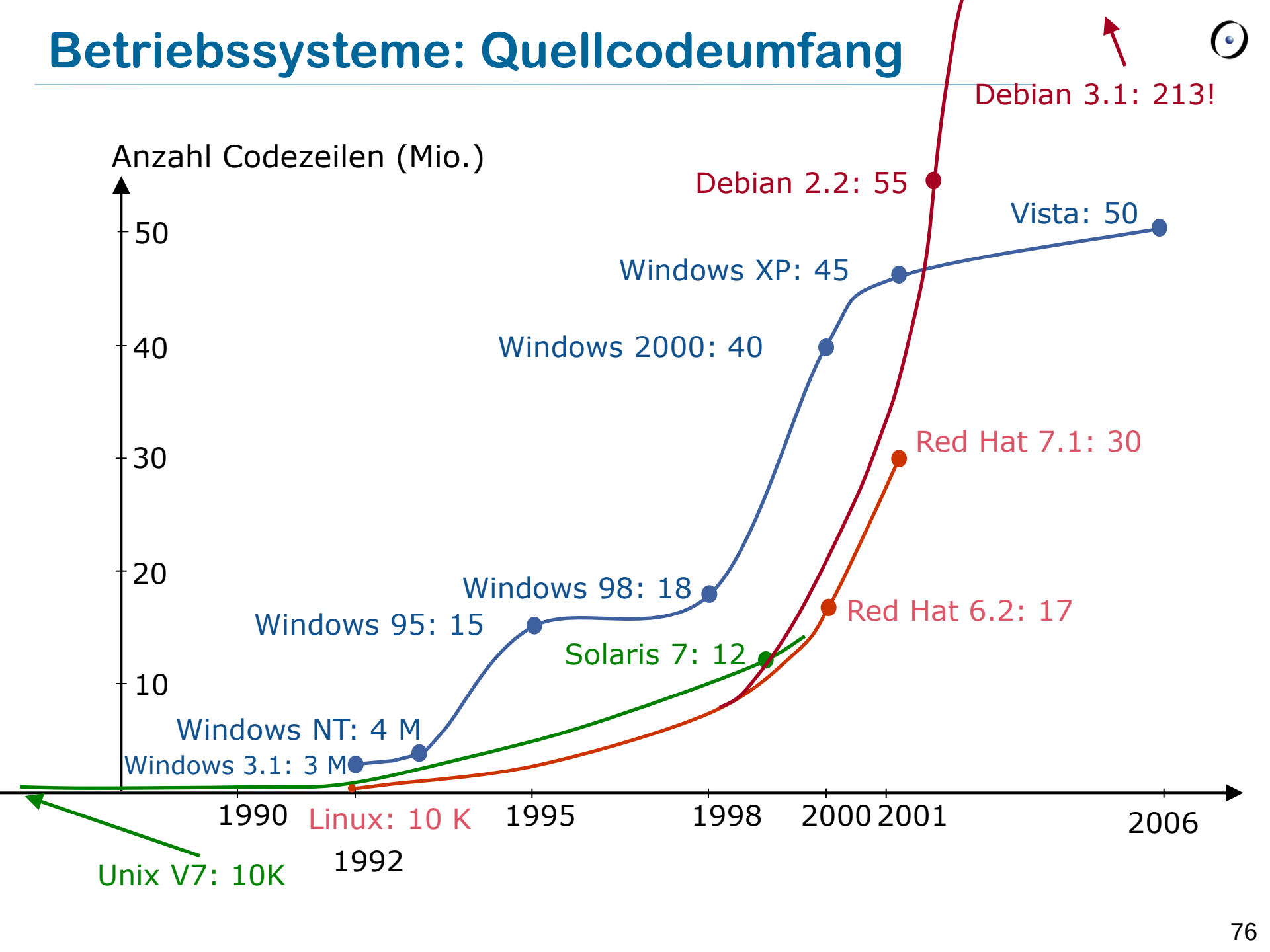
Ein Compiler übersetzt Quellcode in Maschinencode

Der Computer
(genauer: die Plattform aus Computer + Betriebssystem)
findet Ihr Programm im Speicher und führt es aus

Schreiben von Software, die folgende Eigenschaften besitzt:

- Korrekt
  Macht was sie soll!
- Erweiterbar
  Einfach zu ändern!
- Lesbar
  Von Menschen!
- Wiederverwendbar
  Das Rad nicht wiedererfinden!
- Robust
  Reagiert angemessen auf Fehler
- Sicher
  Gegen Angreifer

Betriebssysteme: Quellcodeumfang



Software zu schreiben ist schwierig

Es ist schwierig, ein korrektes Programm zu schreiben

"Trial-and-error" ist sehr ineffizient

Software zu schreiben macht Spass

Entwerfen und entwicklen Sie Ihre eigenen Maschinen

Leben Sie Ihre Kreativität und Ihren Ideenreichtum aus

Programme retten Leben und helfen, die Welt zu einem besseren Ort zu machen

Erleben Sie das Gefühl, wenn ein Programm, das Sie geschrieben haben, funktioniert

Bis zur nächsten Vorlesung

Lesen Sie Kapitel 1 und 2 von *Touch of Class*

Schauen Sie sich die Folien der nächsten zwei Vorlesungen (2 und 3) an

Ersatzvorlesung:

Freitag 21. September, 13:15 -15:00 im HG E7