



# Konzepte-Thread

Korrektheitsnachweis (Invarianten und vollst. Indukt.)

Kryptographie

Bäume

Syntaxdiagramme

Rekursiver Abstieg

Infix, Postfix, Operatorbaum, Stack

Codegenerierung, Compiler, Interpreter

**Verzahnte** und **verwobene** Einführung konzeptioneller Aspekte und programmiersprachlicher Konstrukte

Polymorphie

Suchbäume, Sortieren

Backtracking

Spieltheorie, Minimax, Alpha-Beta

Rekursives Problemlösen

Effizienz, O-Notation

Simulation (zeit-, ereignisgesteuert)

Heap, Heapsort

Pseudoparallelität

# Java-Thread

Java: C-Level

Java-Klassen als Datenstrukturen

Klassen-Instanzen und Referenzen

Java-VM als Bytecode-Interpreter

Pakete

Klassenhierarchie

Abstrakte Klassen

Exceptions

**Programmbeispiele** dienen gleichzeitig der Einführung programmiersprachlicher Konstrukte und der Illustration von Informatikkonzepten

Threads in Java



# Themen

Programmieren in Java

1. Ein Algorithmus und seine Implementierung in Java
2. Java: Elementare Aspekte
3. Klassen und Referenzen
4. Syntaxanalyse und Compiler
5. Java: Pakete
6. Objektorientierung
7. Java: Weitere Sprachelemente
8. Binärsuche auf Arrays

Allg. Informatik-Grundkonzepte

Algorithmen und Datenstrukturen

9. Suchbäume
10. Backtracking
11. Spielbäume und Spieltheorie
12. Rekursives Problemlösen
13. Komplexität von Algorithmen
14. Modellierung und Simulation
15. Heaps
16. Parallele Prozesse / Threads

Anwendungsaspekte

# Resümee – Ziele der Vorlesung

„Eine gute Hochschule vermittelt nicht primär Wissen, sondern die Fähigkeit zu denken.“ [Lino Guzzella, früherer Rektor und Präsident der ETH Zürich]

- **Primär: Fundamentale Begriffe und Aspekte der Informatik**
  - Konzepte, Modelle, Problemlösungstechniken
  - Algorithmen und Datenstrukturen
  - Komplexität, Parallelität
- **Sekundär: Programmieren**
  - Java (→ Übungsaufgaben, Tutorien)
  - Reversi-Spielturnier
- **Auch: Techniken für qualitativ hochwertige Software**
  - Objektorientierung
  - Korrektheit



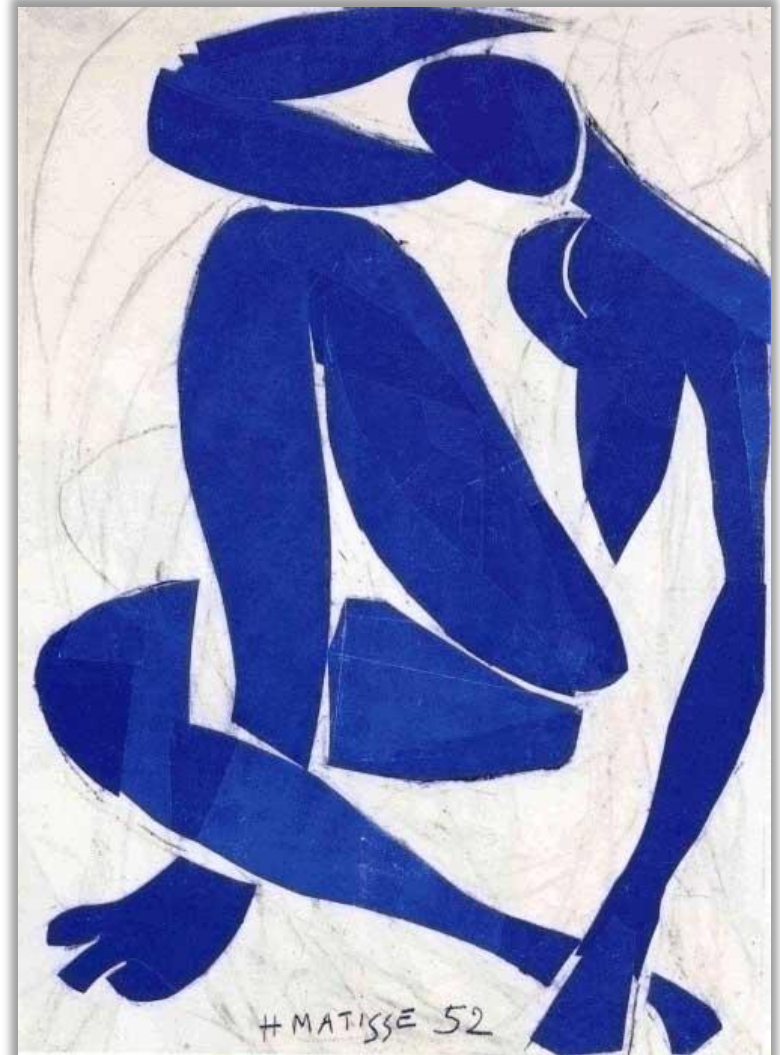
“Due to recent technological advances, everything I taught you about computers is no longer valid”

# Abstraktion

Gedankliches Verfahren, durch das von bestimmten gegebenen, jedoch als unwesentlich erachteten Merkmalen abgesehen wird, sodass sich das Augenmerk auf das Wesentliche konzentriert.

- Immer wieder wurde im Verlauf der Vorlesung von **Abstraktion** und **Abstrahieren** gesprochen
- Wieso ist Abstraktion wichtig?

*"On the other hand, there are a number of students who are not so able. They [...] do not appreciate the utility of modeling, find it difficult to identify what is important in a problem, and produce convoluted solutions that replicate the problem complexities. Why? What is it that makes the good students so able? What is lacking in the weaker ones?" -- Jeff Kramer*



Henri Matisse: Nu bleu IV, 1952

# Abstraktionsvermögen ist der Schlüssel zum Erfolg

- Unwesentliches **ausblenden**
- **Generalisieren**
  - Aus spezifischen Instanzen wesentliche gemeinsame Aspekte herausfaktorieren und den gemeinsamen Kern freilegen
  - Generalisierte Erkenntnis auf andere, analoge Situationen übertragen
- Abstrahieren **lernt man indirekt**
  - Durch viele Beispiele
  - Im Verlaufe der Zeit
- Der Prozess ist **mühsam** – aber **Abstraktion macht frei**
  - Komplexität wird reduziert
  - Zusammenhänge werden klar

„Hidden Agenda“  
dieser Vorlesung



“Removes all detail yet conveys much”

# Abstraktion

David und Bathseba von Lucas Cranach d. Ä. (1526)  
und eine abstrakte Version von Pablo Picasso (1947)



# Geschichte und Kontext

Immer wieder sind in Form von „**Bonus-Slides**“ historische Exkurse in den Stoff der Vorlesung eingeflochten – wieso? →

### Hilbert-Kurve – raumfüllend bei IKEA

Die Hilbert-Kurve ist eine fraktale Kurve, die den gesamten Innenraum eines Quadrats ausfüllt. Sie ist nach dem Mathematiker David Hilbert benannt. In einem Werbeflyer von IKEA wird diese Kurve als 'Hilbert-Kurve' bezeichnet, die in einem Möbelstück verwendet wird.

### Der Schachtürke

(Witzgang von Kassel, 1701)

Ein berühmtes Schachspiel aus dem Jahr 1701, bei dem ein Mann in einem turbanartigen Kopfschmuck (der Schachtürke) gegen einen Menschen spielte. Die Schachfiguren waren in einem Kasten versteckt, den der Mensch nicht sehen konnte.

### Ambrosius Speiser

Ambrosius Speiser war ein österreichischer Ingenieur und Erfinder. Er entwickelte den 'Speiser-Rechner', einen mechanischen Rechner, der für die Berechnung von Werten in der Luftfahrt verwendet wurde.

### Abakus: Rechnen auf der Linie (im Sand)

Der Abakus ist eine frühe Form des Rechenrahmens. Er besteht aus einer Reihe von Kugeln oder Perlen, die auf einer Linie angeordnet sind. Er wurde zur Berechnung von Summen und anderen arithmetischen Operationen verwendet.

### Beispiel: Flugsimulator

Ein frühes Beispiel für einen Flugsimulator, das in den 1920er Jahren entwickelt wurde. Es diente dazu, Piloten zu trainieren, ohne tatsächlich in ein Flugzeug steigen zu müssen.

### Norbert Wiener (1894 - 1964), der Begründer der Kybernetik

Norbert Wiener war ein amerikanischer Mathematiker und Philosoph. Er ist bekannt für seine Arbeit an der Kybernetik, der Theorie der Steuerung und Kommunikation in lebenden Organismen und Maschinen.

### Ambrosius Speiser

Ambrosius Speiser war ein österreichischer Ingenieur und Erfinder. Er entwickelte den 'Speiser-Rechner', einen mechanischen Rechner, der für die Berechnung von Werten in der Luftfahrt verwendet wurde.

### Menschliche Rechner

Historische Aufnahmen von 'Human Calculators', Frauen, die in den 1940er Jahren in den USA eingesetzt wurden, um komplexe Berechnungen für die Luftfahrtindustrie durchzuführen.

### Carl Friedrich Gauß (1777 - 1855)

Carl Friedrich Gauß war ein deutscher Mathematiker, Physiker und Astronom. Er ist einer der größten Mathematiker aller Zeiten und hat in fast allen Bereichen der Mathematik bedeutende Beiträge geleistet.

### Machina Arithmeticae Dyadicae (Leibniz, 1679)

Die 'Machina Arithmeticae Dyadicae' war ein mechanischer Rechner, den Gottfried Wilhelm Leibniz im Jahr 1679 erfand. Er war ein Vorläufer moderner elektronischer Rechner.

### Machina Arithmeticae Dyadicae (Leibniz, 1679)

Leibniz'sche Handschrift zur Darstellung des binären Systems (Zwei-System). Er zeigte, wie man alle Zahlen und Operationen mit den Ziffern 0 und 1 darstellen kann.

### Al-Chwarizmi

Al-Khwarizmi war ein persischer Mathematiker, Astronom und Geograph. Er ist bekannt für seine Arbeit an der Algebra und der Entwicklung des Dezimalsystems.





«Ziel der ETH ist es, ihre Studierenden zu kritischen und unabhängig denkenden Persönlichkeiten auszubilden. Im Laufe ihrer Ausbildung sollen sie nicht nur Methodenkompetenz und disziplinäres Wissen erlernen...» [Critical-Thinking-Initiative der ETH]

Man mag diese „Geschichten“ zunächst ignorieren – sie scheinen von den eigentlichen Themen der Vorlesung abzulenken und kommen schliesslich auch nicht in der Prüfung vor. Aber befasst man sich später intensiver mit einer Sache, stellt man unweigerlich Querbezüge zu anderen Teilthemen fest; auch zu Dingen, die man z.B. noch vom Mathematikunterricht der Schule her kennt oder die man sonst irgendwo bei der Beschäftigung mit naturwissenschaftlichen oder geschichtlichen Aspekten erfahren hat. Die Frage, wie diese Aspekte zusammenhängen und wo eigentlich der Ursprung liegt, taucht dann fast von selbst auf.

Aber auch unabhängig von der individuellen Neugier gilt: Die Informatik hat eine Reihe sehr mächtiger Prinzipien, Paradigmen und Ideen hervorgebracht – man denke etwa an das Prinzip des Algorithmus, an Programmiersprachen, Objektorientierung, aber auch abstraktere Konzepte wie Komplexitätsklassen, Rekursion, Backtracking, Verifikation mit Invarianten etc. Die zugrundeliegenden Ideen sind oft mit sehr alten mathematischen, aber auch kulturgeschichtlichen Fragen und Anstrengungen verbunden, viele kluge Geister der vergangenen Jahrhunderte, manchmal gar Jahrtausende, haben sich zum Beispiel abgemüht, astronomische

## Geschichte und Kontext (2)

„Für mich hat der Dozent viel zu viel Zeit mit unnötigen Geschichten und zusätzlichen nicht prüfungsrelevanten Fakten verbraucht anstatt die Theorie etwas praktischer zu erklären.“ [Aus „Meine Kommentare“ beim Vorlesungsfeedback]

Phänomene effizient berechenbar zu machen, das Rechnen für die Kaufleute zu vereinfachen oder die Logik des Denkens zu automatisieren. Die Erkenntnisse, die dabei gewonnen wurden (man vergegenwärtige sich z.B. die Stellenschreibweise von Zahlen inklusive der Erfindung der Null oder die unterschiedlichen mechanischen Rechenhilfsmittel), bereiteten den Boden für viele der grundlegenden heutigen Informatikkonzepte, stellvertretend seien hier das Konzept der Programmvariablen, das Dualsystem und die Schaltalgebra genannt.

Auch über den mit der Informatik eng verbundenen Aspekt der Automatisierung eröffnet sich eine interessante historische Perspektive. Dass einem Arbeit abgenommen wird und man durch Automaten, die stattdessen die Mühsal verrichten, erst noch produktiver wird, ist ein uralter Wunsch und damit Antrieb, Technik zu betreiben. Windmühlen gab es schon lange vor der eigentlichen industriellen Revolution, und Wasserräder trieben nicht nur Getreidemühlen an, sondern Sägewerke, Blasebälge und schliesslich ganze Textilfabriken. Wir wissen, wie es weiterging: Dampfmaschinen, Elektrizität und Benzinmotoren vervielfachten das Potential – leider nicht nebeneffektfrei hinsichtlich Umwelt und Gesellschaft, wie uns im Laufe der Zeit immer bewusster wurde. Und nun hoffen wir, dass die „smarte“ Technik des 21. Jahrhunderts die Probleme des 20. Jahrhunderts löst!

# Geschichte und Kontext (3)

„Besonders die Bonusfolien sind sehr interessant und ein klarer Mehrwert.“  
„Ich würde die Zusatzfolien (violett) unbedingt beibehalten.“ [Feedback]

Dampfkraft, Elektrizität und Elektronik, letztere im Verbund mit der damit ermöglichten (zunächst noch analogen) Kommunikations- und Regelungstechnik, waren die wesentlichen Treiber der verschiedenen Stufen der industriellen Revolution. Bis dann die Digitaltechnik auf den Plan trat. Diese erlaubte eine ganz andere Art der Informationsspeicherung und Verarbeitung; zur neuen treibenden Kraft wurde (schleichend und zunächst fast unbemerkt, aber unaufhaltsam) ab den 1950er-Jahren der Digitalcomputer, zusammen mit einem durch ihn induzierten ganzen „Ökosystem“ an Software und Diensten. Mit digitalisierter Information, die verlustfrei und effizient übertragen und verarbeitet werden kann, lassen sich die unterschiedlichsten Prozesse in Wirtschaft, Industrie und Verwaltung auf informierte (oder gar „smarte“) Art steuern und automatisieren – viel umfänglicher, genauer und schneller als es auf herkömmliche Weise ging. Wenn derzeit allenthalben von selbstfahrenden Autos, dem Internet der Dinge sowie einer umfassenden „Digitalisierung“ wirtschaftlicher und gesellschaftlicher Bereiche gesprochen wird, dann ahnen wir, dass wir bisher erst ein Anfangsstück des Phänomens gesehen und erlebt haben.

Das Prinzip der Automatisierung, insbesondere auch der Automatisierung intellektueller und informationsverarbeitender Prozesse, hat nun ebenfalls historische Vorläufer – zumindest in der Ideengeschichte, denn tatsächlich realisieren liess sich früher Vieles mit den gegenüber heute eher bescheidenen

# Geschichte und Kontext (4)

„Kritisches Denken in den Wissenschaften“ – das klingt wie eine reine Tautologie! Wissenschaft und Forschung haben es doch immer mit kritischer Reflektion zu tun. In der Wis- →

technischen Mitteln noch nicht. Wie in der Vorlesung skizziert, mühten sich beispielsweise Leibniz, Babbage und Lewis Fry Richardson nach Kräften ab und konnten doch bestenfalls nur zeigen, dass ihre Ideen im Prinzip oder vielleicht einmal in Zukunft mit praktischem Gewinn umsetzbar sein sollten; viele Zeitgenossen liessen sich jedenfalls nicht überzeugen. Doch der Traum, das Rechnen – und vielleicht sogar Teile der Mathematik einschliesslich der Mathematiker – zu automatisieren (und langwierige Rechnungen dadurch fehlerfrei zu machen), logische Formeln automatisch auf „wahr“ oder „falsch“ prüfen zu lassen oder eine schachspielende Maschine zu bauen, die dann auch andere schwierige strategische Probleme mit ihrer künstlichen Intelligenz lösen kann, hat viele Denker angetrieben und etliche Erkenntnisse hervorgebracht.

Die Ideengeschichte der Informatik ist nun allerdings nicht so einfach zugänglich – alte Schriften zu studieren und sich in die Vorstellungswelt und den andersartigen Kontext früherer Zeiten und vergangener Geistesgrössen hineinzudenken, dabei den Bezug zu den in modernerem Vokabular ausgedrückten heutigen Konzepten herzustellen, ist etwas mühsam. Ganz abgesehen davon, dass es das Alte und Klassische bei einem sich besonders schnell entwickelnden und nach Innovationen gierenden Gebiet grundsätzlich etwas schwerer hat, sich bemerkbar zu machen und als relevant angesehen zu werden. An dieser Stelle setzen die eingestreuten historischen Exkurse an. Sie sollen zunächst einmal beispielhaft aufzeigen, was für

# Geschichte und Kontext (5)

senschaft wird untersucht und hinterfragt, statt die Dinge einfach so zu nehmen, wie sie scheinen. Oder ist das vielleicht nur ein Ideal?“ [N. Sieroka, V. I. Otto, G. Folkers]

traditionelle Ideen es gibt und verdeutlichen, dass für ihre Entwicklung und Durchsetzung meist langwierige historische Prozesse relevant waren. Aber auch darauf hinweisen, dass es gelegentlich herausragende Personen waren, die eine geniale Idee hatten oder ein Konzept (wie etwa das Rechnen „auf der Feder“ anstelle des Rechnens „auf der Linie“) mit besonderem Nachdruck verfolgten und diesem so zum Durchbruch verhalfen.

Man könnte die historischen Exkurse auch mit dem traditionellen Bildungsauftrag einer Universität oder neomodischen und neudeutschen Ableitungen hiervon („critical thinking“) rechtfertigen: Das Heutige in einen historischen Kontext zu setzen, bildet. Das ist natürlich kein Selbstzweck, sondern schärft den Blick dafür, wie Neues entstehen kann und welche Faktoren dafür relevant sind. Es hilft vielleicht auch ein bisschen dabei, die heutige Situation einzuschätzen und zu bewerten. Und es macht bescheiden: Was vergangene Geistesgrößen gefunden und geschaffen haben, fand meist unter deutlich widrigeren Umständen statt als heute, wo wir diese Dinge mit viel weniger Mühe nachvollziehen oder wiedererfinden können. Oder anders ausgedrückt: Man spürt, was Fortschritt in seiner historischen Dimension bedeutet und welche Strapazen manche auf sich genommen haben, um erkenntnismässig etwas beizutragen und damit dem Fortschritt zu dienen. Wir verdanken diesen Personen viel, auch den vielen Namenlosen und Vergessenen!

# Geschichte und Kontext (6)

„So, I think history matters, even though IT is always about the future and no one wants to believe that history matters. And there's more to history than the firsts and lone geniuses.“ [Thomas Haigh]

Wir leben in spannenden Zeiten und sind mit unserem Fach mittendrin im Geschehen! Wo die Reise hingehet, wissen wir nicht. Aber vielleicht hilft der Blick zurück – auf einige frühere Stationen unserer Reise – beim Erkennen, worauf es trotz aller Offenheit und Unwägbarkeit des Kommenden etwas mehr und worauf es etwas weniger ankommt? Also: Was ist wichtig? Was macht uns stark für die nächste Reiseetappe? Was hat Bestand? Welche Ideen sind nachhaltig? Was soll man lernen? Was soll die ETH lehren?

Vergegenwärtigt man sich in dieser Hinsicht nochmals die kleinen eingeflochtenen Geschichten der Vorlesung, dann mag man sich unter anderem an folgendes erinnern:

1) An erster Stelle steht sicherlich der für die Informatik so wesentliche **Algorithmusbegriff**, für den al-Chwarizmi namensgebend war, der mit Beginn der Neuzeit (und seiner aufkommenden Buchdruckindustrie) in vielen Rechenrezepten der Rechenmeister seine Ausprägung findet, der aber auch schon viel früher, etwa beim euklidischen Algorithmus der Pythagoreer, auftaucht. Die Erklärung in der Enzyklopädie von Diderot und d'Alembert demonstriert die Mitte des 19. Jahrhunderts noch etwas unscharfe Auffassung des Begriffs. In eher ungewöhnlicher Weise führt uns die Osterformel von Gauß vor Augen, dass man Algorithmen auf sehr unterschiedliche Art formulieren kann – die jahrhundertealte Methode zur

# Geschichte und Kontext (7)

„Voraussetzung einer Öffnung nach vorne ist [...] ein stärkeres Verhältnis zur Geschichte – oder vielleicht genauer: überhaupt einmal ein Bewusstsein von Geschichte. [...] Es gibt eine Sym- ➔

Bestimmung des Ostertags, die im Messbuch („De anno et ejus partibus“) umständlich und seitenlang beschrieben ist, kommt ganz ohne Mathematik aus; sie wurde formuliert, als sich im zugehörigen Kulturkreis noch nicht einmal die Null als Zahl oder Ziffer etabliert hatte. Das dahinterliegende Modell einer vereinfachten Himmelsmechanik aus Sonne, Erde und Mond mit ihren zyklischen Prozessen ist aber eigentlich nicht kompliziert, und Gauß kann mit einer inzwischen entsprechend „weit“ entwickelten (aus heutiger Sicht aber elementaren) Formelsprache der Mathematik den Algorithmus in wenigen Zeilen notieren.

2) Der Schritt vom Algorithmus zum **Computerprogramm** forderte ebenfalls einiges ab. Nun gibt es also eine Maschine, die mechanisch (wie bei Babbage) oder elektromechanisch (wie bei Zuse) Folgen von Einzelrechnungen durchführen kann. Wie bringt man sie aber dazu, genau die vorgesehenen Operationen auszuführen? Babbage, Menabrea und Ada geben sich viel Mühe und müssen nebenbei einige Konzepte erfinden; später machten sich andere (wir schilderten etwas ausführlicher die Arbeiten von Rutishauser dazu, praktisch zeitgleich gab es aber weitere Programmierpioniere), Gedanken, wie man die „Rechenplanfertigung“ auf eine höhere Ebene heben kann – mit algorithmischen Programmiersprachen, wozu man aber erst noch, quasi nebenbei, das Konzept des Compilers erfinden muss, um die Idee praktikabel zu machen.

# Geschichte und Kontext (8)

metrie zwischen dem Zukunftshorizont und dem historischen Erfahrungs- und Bewusstseinsraum. Wo Traditionszusammenhänge schrumpfen, kann sich auch keine Zukunftsphantasie ent- →

3) Die **Objektorientierung** als Programmierparadigma vor allem für nicht-numerische Anwendungen war ein weiterer wesentlicher Meilenstein. Wie wir gesehen haben, zurückgehend auf eine wegweisende Einsicht von Dahl und Nygaard Mitte der 1960er-Jahre, wo Modellierungspraktiken und Ingenieursprinzipien zusammenfanden. Und dennoch dauerte es etwa 25 Jahre, bis dieses wichtige Konzept in grösserem Stil Anerkennung in der Praxis fand!

4) Die Geschichte der **Digitalrechner** und ihrer Einführung lässt sich an der hier episodenhaft dargestellten Verwendung von Zuses Z4 an der ETH sowie der ERMETH-Entwicklung nachvollziehen. Dass vor allem auch in den USA (und dann z.B. auch in Grossbritannien, Deutschland, Schweden oder den Niederlanden) erste Digitalrechner entwickelt wurden, konnte hier allerdings nur am Rande Erwähnung finden.

5) Bezüglich massgeblicher **Ideen und Konzepte** sei an Leibniz und „sein“ Dualsystem und Bestreben, die Logik zu arithmetisieren, erinnert; zu Letzterem haben wir auch andere wichtige Persönlichkeiten erwähnt wie George Boole und Claude Shannon, die in späteren Jahrhunderten diese interessante Idee konkreter vorantreiben konnten. Programme selbst als mathematische Konstrukte aufzufassen und dafür einen Beweiskalkül anzugeben, das war vor knapp 50 Jahren schliesslich eine wesentliche Leistung von Tony Hoare.



# Geschichte und Kontext (9)

falten. In der kritischen Kenntnis der Geschichte findet der in die Zukunft gerichtete Gestaltungswille seinen Hebel und sein unentbehrliches Widerlager.“ [Roman Bucheli, NZZ]

6) Die Geschichte und die Geschichten rund um **Computerschach** und ähnliche Spielprogramme führen zur Motivation etwas anders gelagerter Informatikprinzipien: der Automatisierung kluger Entscheidungen. Die mathematische Modellierung durch die Spieltheorie löst das Problem noch nicht wirklich; trickreiche Algorithmen (etwa Alpha-Beta und Varianten davon) stossen bei der immensen Grösse des Suchraumes auch bald an ihre Grenzen – das Imitieren menschlicher Intelligenz durch schiere Rechenpower ist offenbar nicht so einfach! An den Duellen menschlicher Spielexperten mit Schachprogrammen (oder in jüngerer Vergangenheit mit Go-Programmen) kann man auch erkennen, wie bei diesem Streben nach Simulation (oder gar Realisierung?) menschlicher Intelligenz emotionale Aspekte, vielleicht sogar ethische oder philosophische Aspekte, anklingen. Auf die Ideengeschichte der künstlichen Intelligenz konnten wir aber in dieser Vorlesung nicht näher eingehen; ein der Sache angemessenes historisches Kapitel dazu wird man wohl auch erst in einiger Zukunft schreiben können!

7) Was hat es mit der Meteorologie und der **Simulation** auf sich, denen in der Vorlesung historische Exkurse gewidmet sind? Beides sind wichtige Anwendungsgebiete, die enorme Rechenleistung verschlingen; mit noch feineren Modellen kann man auch zukünftige Supercomputer stets gut auslasten. Richtig angewandt, ist der Nutzen jedenfalls gross. Das Simulationsmodell zu den „Grenzen des Wachstums“ ist inzwischen schon von kulturhistorischem Interesse, auch wenn es

# Geschichte und Kontext (10)

unsere jüngere Vergangenheit betrifft. Jenseits aller Detailkritik am Modell und den konkreten Ergebnissen hatte diese Computeranwendung zweifelsohne eine enorme und vor allem nachhaltige Wirkung auf die Öffentlichkeit. Für eine adäquate Beurteilung braucht es aber wohl weniger eine technische als eher eine kulturhistorische Perspektive.

8) Bei der **Meteorologie** wird noch etwas anderes deutlich: Neben der rechen-technikbezogenen „Übererfüllung“ des Traums von Lewis Fry Richardson, das zukünftige Wetter anhand numerischer Modelle ausrechnen zu können, hat die Informatik auch jenseits des schnellen Rechnens gewirkt und Prozesse verändert. Beispielhaft zeigt sich das nicht nur bei der Wetterstation auf dem Säntis, sondern vor allem daran, wie das Ergebnis der Wetterprognose immer besser aufbereitet und „automatisierter präsentiert“ werden kann, etwa mit Mitteln der Computergraphik und -animation. Meteorologen im Fernsehen braucht man nicht mehr, das Smartphone liefert einem jetzt die Wetterpiktogramme passend zum Ort, wo man sich gerade befindet oder hinfährt. Das Wetter dient dem Fernsehen eigentlich nur noch als Vorwand, die „Presenter“ als Werbeträger zur Schau zu stellen. Deren Qualifikationsprofil hat sich im Zuge der informatikgetriebenen Automatisierung radikal geändert.

# Geschichte und Kontext (11)

„Fang immer bei den alten Römern an und gib stets, wovon du auch sprichst, die geschichtlichen Hintergründe der Sache. Das ist nicht nur deutsch – das tun alle Brillenmenschen ... Man ver- →

9) Die in Exkursen geschilderten **Weltraummissionen** illustrieren wiederum etwas anderes: Ein einziger unentdeckter Fehler wie bei der Ariane-Rakete setzt Milliarden in den Sand, und auf dem Mars mit seinen anderen Umweltbedingungen können auch gut getestete parallele Programme versagen. Alarmierend ist aber, dass solche versteckten Fehler auch hier auf der Erde lauern – wie beschrieben, war ein nichtdeterministisch auftretender Programmier- und Synchronisationsfehler eine Ursache für den grossen Blackout in den USA im Jahr 2003. Zukünftige cyberphysikalische Systeme (wie z.B. „automatische“ Autos) stellen noch höhere Anforderungen an fehlerfreie und fehlertolerante Informatiksysteme!

10) **Déjà-vu**: Manchmal war etwas früher schon einmal so ähnlich da. Mitte der 1950er-Jahre gab es z.B. eine grosse Automatisierungsdebatte: Computer und Roboter würden uns die Arbeitsplätze wegnehmen. 60 Jahre später keimt eine ähnliche Debatte hoch: Selbstfahrende Autos und Algorithmen für fast alles werden auch die gut Ausgebildeten arbeitslos machen. Kann man Parallelen ziehen? Oder sind die Voraussetzungen heute so anders, dass man sich besser nicht auf die seinerzeitige Debatte rückbesinnt?

11) **Menschen und Einzelschicksale**: Nur gelegentlich schimmert durch, dass es die seinerzeitigen Helden auch nicht immer einfach hatten. In Kurzbiographien fällt diesbezüglich meist nur ein früher oder ungewöhnlicher Tod auf (wie

# Geschichte und Kontext (12)

steht es ja sonst nicht, wer kann denn das alles verstehen, ohne die geschichtlichen Hintergründe ... Sehr richtig! Die Leute sind doch nicht in deinen Vortrag gekommen, um lebendiges Leben zu →

beispielsweise George Boole, der eine Vorlesung in völlig durchnässten Kleidern hielt und 49-jährig an Fieber starb). In der Foliensammlung wird beispielhaft an verschiedenen Stellen Dietrich Prinz erwähnt, der Leiter der Programmierung beim Computerhersteller Ferranti war, u.a. den NIM-Spielcomputer entwickelte und Politikern in Berlin präsentierte sowie später ein rudimentäres Schach-Programm erstellte. Prinz studierte in den 1920er-Jahren an der Berliner Universität, promovierte in Physik und arbeitete anschliessend bei Telefunken in Berlin. Er teilt das Schicksal vieler deutsch-jüdischer Wissenschaftler: 1935 verlässt er Deutschland aufgrund des immer virulenter werdenden Nationalsozialismus und der damit verbundenen Diskriminierung und Verfolgung der Juden; er geht nach England. Dort wird er allerdings zu Beginn des Zweiten Weltkriegs als Deutscher und potentieller Feind interniert und nach Kanada gebracht. Glücklicherweise kann er dort seine Fähigkeiten unter Beweis stellen und wird nach dem Krieg, 1947, von Ferranti angestellt. Josef Kates, der hier ebenfalls mit Bezug auf Computerspiele erwähnt wird, erleidet ein ähnliches Schicksal: Mit 17 aus Österreich emigriert, ohne Möglichkeit, in der Schweiz zu bleiben, wird auch er von England aus nach Kanada geschickt und interniert, wo er aber schliesslich eine neue Heimat findet und Karriere machen kann.

Die Einzelschicksale machen betroffen und stehen doch nur stellvertretend für Viele, die Grossartiges geleistet haben und gleichzeitig viel erleiden mussten.

# Geschichte und Kontext (13)

hören, sondern das, was sie auch in Büchern nachschlagen können... Sehr richtig! Immer gib ihm Historie, immer gib ihm.“ [Kurt Tucholsky, 1930]

Porträtfotos und Lebensdaten sollen ihnen, wie auch vielen anderen in der Vorlesung erwähnten Personen, ein Gesicht geben. Denn hinter der abstrakten Wissenschaft und deren verwertbaren Erkenntnissen, die die uns täglich dienen, stecken Menschen – die vielleicht auch einmal in einem ähnlichen Hörsaal saßen und sich ganz ähnlichen Gedanken und Gefühlen hingaben wie die heutige Generation von Studierenden.

12) **Sonderbare Vergangenheit:** Jemand, der „simuliert“ denkt nach oder täuscht gleich eine Geisteskrankheit vor; ein Rechner oder Computer war damals ein Mensch und kein Ding aus Blech, politisch korrekt war anfangs wohl die Bezeichnung „programmgesteuerte Rechenmaschine“; das Wort „Automatisierung“ gab es vor 1945 einfach noch nicht; und statt „Informatik“ hätte es fast „Informatiorik“ geheißen. Sonderbar auch die Leute damals, in den 1970er-Jahre: Rauchten im ETH-Hörsaal, nutzten Computer mit 1.7 MHz Taktfrequenz statt 1.7 GHz und 4 kB RAM statt 4 GB. Geht das überhaupt? Und Programme auf Musikkassetten abspeichern – wie kann man nur auf so eine hanebüchene\*) Idee kommen? Da kann man doch eigentlich froh sein, **dass man im Hier und Heute lebt!**

---

\*) Auch wieder so ein Wort, das es eigentlich schon lange nicht mehr gibt...

**Warnung!**  
**Geschichte kann zu  
Einsichten führen  
und verursacht  
Bewusstsein.**

# Was will die Critical-Thinking-Initiative der ETH?

„Mit der Critical-Thinking-Initiative will die ETH Zürich ihre institutionelle Diversität erhöhen, eine Kultur des disziplinenübergreifenden Austauschs, des kritischen Denkens und eigenverantwortlichen Handelns fördern. [...]

Wenn es das Ziel der ETH ist, hochqualifizierte Fachkräfte auszubilden und ihnen dabei eine breite Bildung mitzugeben, die sie fit macht für die Herausforderungen der Zukunft, muss sie den Absolvierenden neben der Vermittlung von Fachwissen und Forschungskompetenz ein hinreichendes Mass an Kontextverständnis zur Einordnung ihres Wissens mitgeben. [...]

Ausgehend von diesem Ansatz geht es unter dem Begriff „critical thinking“ um die Förderung der Fähigkeit von Studierenden, verschiedene Positionen, Sichtweisen und Anspruchshaltungen zu unterscheiden. [...]“

Ganzer zitierter Text: [www.ethz.ch/services/de/organisation/schulleitung/praesident/critical-thinking/ueber-cteth.html](http://www.ethz.ch/services/de/organisation/schulleitung/praesident/critical-thinking/ueber-cteth.html) [Zugriff am 13.2.2020]

„Auf der Hochschulstufe ist kritisches Denken ein Phänomen, weil es überhaupt Thema ist. Aus der Tradition und aus dem universitären Selbstverständnis heraus müsste es eigentlich eine Selbstverständlichkeit sein. Dass es dies nicht ist, macht es auffällig.“  
[Peter Treppe, Balthasar Eugster, 2018]



# Prüfungsvorbereitung

The lyf so short, the  
craft so long to lerne.  
-- Geoffrey Chaucer,  
ca. 1382

Also lautet ein Beschluss,  
dass der Mensch  
was **lernen** muss.

*„Es ist ja immer dasselbe Spiel. Du arbeitest zwei Wochen daraufhin, den Rasen top vorzubereiten. Und dann kommen 90 Minuten, in denen dein Werk wieder zerstört wird. Das ist bitter.“ – Marvin Guse, Head-Greenkeeper des Sportklubs VfL Bochum.*





# Die Prüfung

Der Dozent verschränkte die Hände auf dem Rücken, verlagerte sein Gewicht auf das gesunde Bein und fragte: „Was täten Sie, Kadett, wenn Sie bei einem Patrouillenflug auf das Schiff eines fremden Planeten stießen?“

Pirx öffnete den Mund, als wollte er die darin enthaltene Antwort vertreiben. Er sah aus wie der letzte Mensch – der letzte Mensch auf Erden, der zu erklären wüsste, was er zu tun hat, wenn er Raketen von fremden Planeten begegnet.

„Ich würde näher heranfliegen“, sagte er mit dumpfer, merkwürdig rauher Stimme. Die Lehrgangsteilnehmer wurden still. Sie witterten eine willkommene Abwechslung.

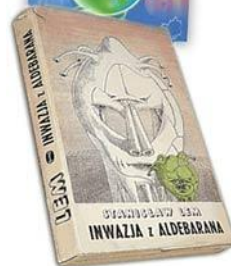
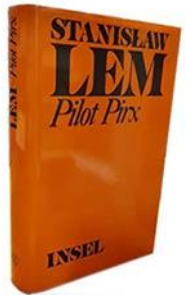
„Sehr gut“, sagte Eselswiese väterlich, „aber was weiter?“

„Ich würde stoppen“, platzte Kadett Pirx heraus, denn er fühlte, dass er im Niemandsland umhertappte, weit vor der vordersten Linie seiner Kenntnisse. Fieberhaft durchsuchte er sein leeres Hirn nach Paragraphen für das Verhalten im Raum. Irgendwann muss ich mal was darüber gelesen haben, dachte er. Bescheiden senkte er den Blick und sah, dass Smiga ihm etwas vorsagen wollte – er bewegte dabei nur die Lippen. Pirx begriff und wiederholte laut, bevor ihm der Sinn der Worte klar wurde: „Ich würde mich ihnen vorstellen.“

Das Auditorium brüllte wie ein Mann. Eselswiese kämpfte eine Sekunde mit sich, lachte dann auch, wurde aber gleich wieder ernst.

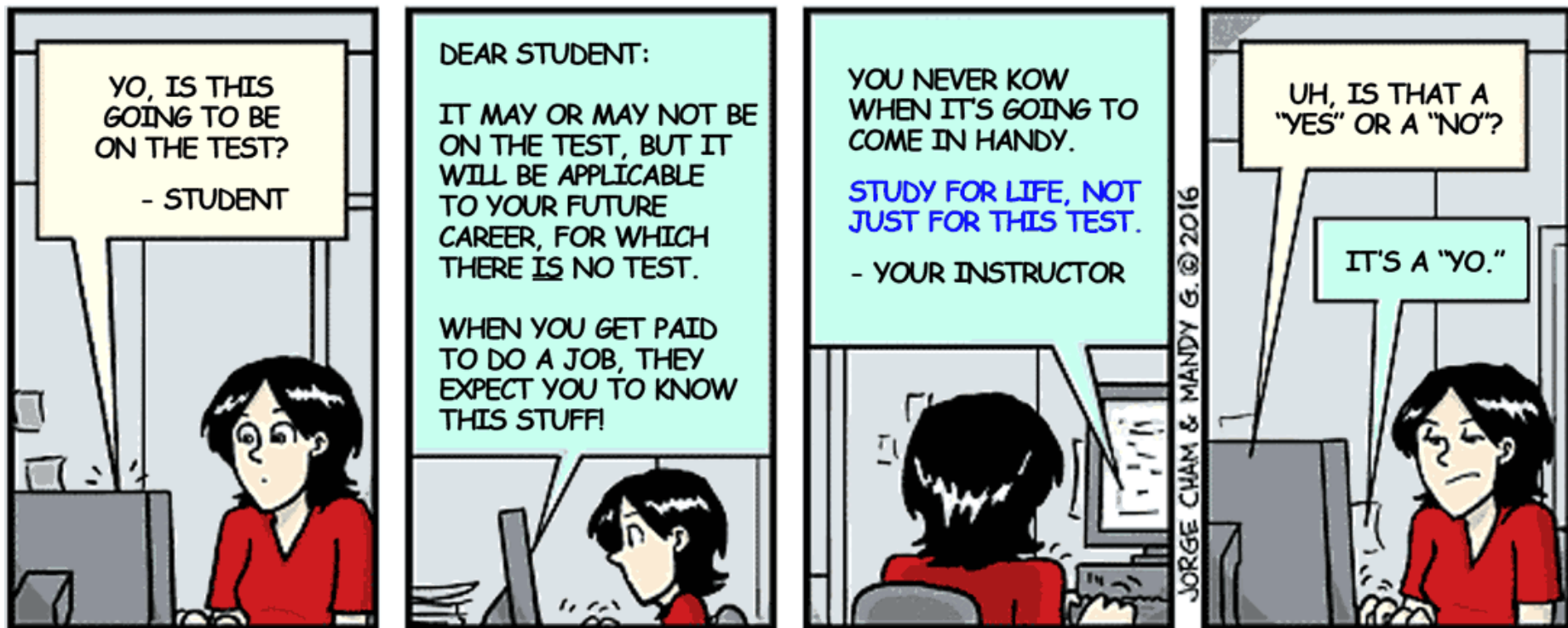
„Kadett, Sie kommen morgen mit dem Navigationsbuch zu mir!“

*Weiterlesen?: Stanisław Lem: „Test“; in: „Pilot Pirx“ bzw. „Die Entdeckung der Virtualität“. (Erstdruck auf Polnisch in „Inwazja z Aldebarana“, 1959.)*



# Hinweise zur Prüfungsvorbereitung (1)

- Relevant ist der gesamte Inhalt der Vorlesung und Übungen
  - Nicht alleine diese Präsentationskopie
  - Auch Themen der letzten Vorlesungsstunden (Parallelität, Threads, Simulation etc.) und generell „theoretische“ Themen (z.B. Verifikation oder Komplexität) sind sehr wichtig und daher prüfungsrelevant



# Hinweise zur Prüfungsvorbereitung (2)

- Die wöchentl. **Übungsaufgaben** sind eine gute Vorbereitung
  - Oft erscheinen einige (leicht verändert) in der Prüfungsklausur
  - In der Klausur wird zwar etwas mehr Wert auf Konzepte gelegt, jedoch sind bei einigen Aufgaben Java-Programmfragmente anzugeben
- Vorbereitung in kleinen **Lerngruppen** ist empfehlenswert
- Das angegebene **Lehrbuch** liefert eine zweite Sicht
  - Das Wissen kann so gefestigt werden
- **Alte Prüfungen** vergangener Jahre sind nicht mustergültig

„Es ist schlimm“, rief Eduard, „dass man jetzt nichts mehr für sein ganzes Leben lernen kann. Unsre Vorfahren hielten sich an den Unterricht, den sie in ihrer Jugend empfangen; wir aber müssen jetzt alle fünf Jahre umlernen, wenn wir nicht ganz aus der Mode kommen wollen“.

*Johann Wolfgang von Goethe: Die Wahlverwandtschaften (1809)*

# Hinweise zur Prüfungsvorbereitung (3)

- Diese Präsentationskopie enthält „**bonus** off-syllabus slides“ (Folien, die nicht in der Vorlesung gezeigt wurden) als **zusätzliches Material**, um interessierten Studierenden einen breiteren Kontext zu bieten. Dieses Zusatzmaterial kann die Prüfungsvorbereitung unterstützen; ganz neue Themen (z.B. historische Bezüge) sind aber kein Prüfungsstoff.
- Es sind in den Text auch einige „**Denkübungen**“ eingestreut: Kleine Probleme, deren Lösung mit Absicht nicht angegeben werden – sie können leicht oder auch schwierig sein und sollen zur intensiveren Beschäftigung mit dem Thema anregen. Denn nicht die Lösung selbst, sondern der Weg (also das Nachdenken über das Problem) ist das Ziel!

The book *Dynamic Programming* by Richard Bellman is an important, pioneering work in which a group of problems is collected together at the end of some chapters under the heading “Exercises and Research Problems,” with extremely trivial questions appearing in the midst of deep, unsolved problems. It is rumored that someone once asked Dr. Bellman how to tell the exercises apart from the research problems, and he replied: “If you can solve it, it is an exercise; otherwise it’s a research problem.” -- Donald Knuth

# Hinweise zur Prüfungsvorbereitung (4)

- Prüfungszeit ist kurz → Prüfung ist daher eher eine Stichprobe; sie umfasst längst nicht alle Teilthemen der Vorlesung
  - Jeweils andere Stichprobe → frühere Prüfungen nicht mustergültig
  - Themengewichtung der Prüfung nicht „proportional“ zur Vorlesung
  - Auch scheinbar (!) „unwichtige“ Teilthemen sind nicht unwahrscheinlich; man hüte sich davor, nur „Schwerpunktthemen“ zu lernen
- *Kommentare von Studierenden nach der Prüfung im O-Ton:*
  - *...Multi-Threading. Bei der Vorlesung handelte es sich um eine oder höchstens zwei Vorlesungen über Parallelität... Aber dann in der Info II Prüfung gab es 1/6 der Punkte in einer Aufgabe über Multi-Threading*
  - *Bytecode analysieren? So viel zu Threats und parallele Prozesse? Sehr unerwartet.*
  - *Ich hätte als Dozent mehr Wert auf wichtige Konzepte nämlich - Stack, List, Bäume, Spieltheorien - gelegt, und nicht auf Parallelität und solches...*
  - *Letzte Übungsserie wurde in der Übungsstunde nur sehr schlecht besprochen trotz schwierigen Themen, in der Prüfung war aber sehr viel davon zu sehen*
  - *Die Aufgabe über Induktionsbeweis war unfair und entspricht eigentlich nicht den Lernzielen der Vorlesung*

# Hinweise zur Prüfungsvorbereitung (5)

- Klausur umfangreich → alle Aufgaben kaum (perfekt) lösbar
  - Für eine bestmögliche Note benötigt man aber nicht 100% der Punkte
  - Konzentration auf Lösung zu Aufgaben, deren Themen man beherrscht
- *Kommentare von Studierenden nach der Prüfung im O-Ton:*
  - *...dass es sehr schwierig bis fast nicht möglich war, alle Aufgaben innerhalb der kurzen Zeit zu bearbeiten.*
  - *Ich habe nicht ein so grosser Zeitdruck erwartet*
  - *Die Zeit ist bei info 1 schon knapp, aber bei Info 2 war es die Hölle...fast jede einzelne Teilaufgabe musste begründet werden*
  - *Da die Themenauswahl sehr spezifisch war, bzw ein Grossteil des Vorlesungsinhaltes nicht abgefragt wurde, waren diejenigen Studenten bevorzugt, die "zufällig" das richtige Randthema gelernt hatten.*
  - *Die Fragen waren völlig übertrieben schwierig und kamen völlig unerwartet. Ich weiss, so sollten Prüfungen auch sein, aber es gibt dazu noch Grenzen.*
  - *...eine sehr amüsante Stunde, die ich mit Lösen, Raten und Lustige Antworten schreiben verbrachte.*

# Hinweise zur Prüfungsvorbereitung (6)

- Java ist wesentlicher Bestandteil von Vorlesung und Übung
  - Man sollte kleine Java-Programmfragmente (vergleichbar den Beispielen aus der Vorlesung) „von Hand“ programmieren können
- *Kommentare von Studierenden nach der Prüfung im O-Ton:*
  - *...ein überraschend grosser Schwerpunkt auf Programmierfähigkeiten gelegt*
  - *...kommen 3 Programmieraufgaben wo man in einer Stunde lösen musste. Ich hätte mich besser aufs Programmieren vorbereiten sollen in der Lernphase*
  - *Ich hätte nicht 1/3 programmieren erwartet*
  - *Klar machen, dass bei der Prüfung Codes auf Papier geschrieben werden müssen können, ohne Hilfe von Eclipse oder anderen.*
  - *Ich finde es schade, dass die Assistenten uns gesagt haben, wir müssten kaum programmieren. Aber da die Programmieraufgaben nicht extrem schwer waren, war dies völlig in Ordnung!*
  - *Im Vergleich mit den alten Prüfungen fand ich es in dieser sehr viel besser, dass Theorie und Programmieraufgaben ausgeglichener waren*

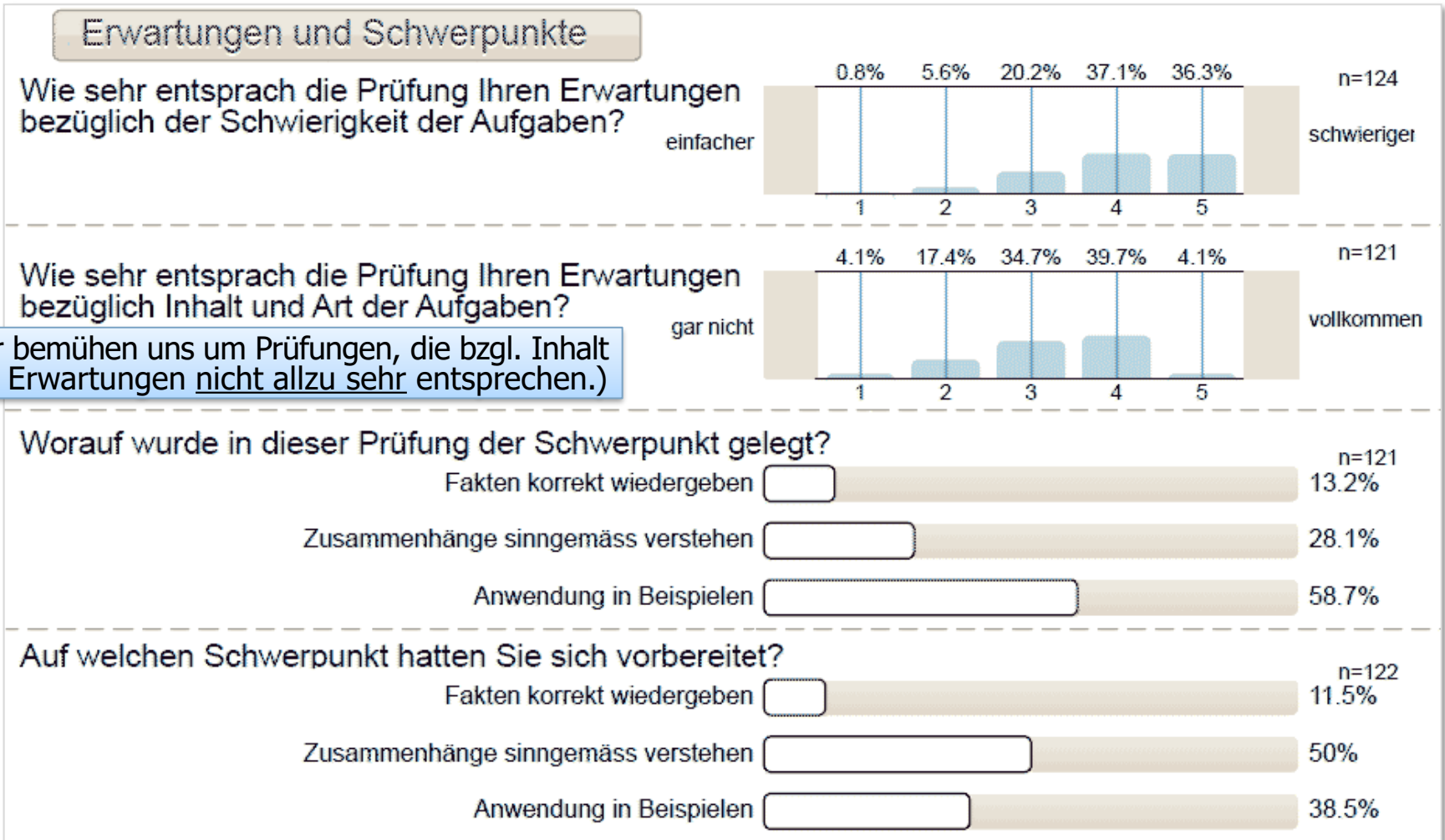
# Hinweise zur Prüfungsvorbereitung (7)

- **Verständnis der Konzepte ist wichtiger als „Codieren“**
  - Oberflächlich mag manches „spitzfindig“ aussehen (z.B. der Unterschied von „&“ und „&&“), dahinter steckt aber manchmal ein tiefsinniges Konzept (im Beispiel: Shortcut-Evaluation → Basis des  $\alpha$ - $\beta$ -Verfahrens)
- ***Kommentare von Studierenden nach der Prüfung im O-Ton:***
  - *Bei den Übungen war zu viel programmieren gefragt (= sehr zeitintensiv) und dann an der Prüfung nur wenig coden.*
  - *Ich war ein wenig enttäuscht, ich habe so viel gelernt, und konnte alle möglichen Codes und Dinge, die man sich mit Fleiss und Disziplin aneignen kann und dann kommt eine Prüfung, die nur auf Verständnis setzt.*
  - *Dachte, würde mehr eine Lernprüfung werden als hauptsächlich Verständnis*
  - *Die Aufgabe mit dem and und andand: wenn man den Unterschied an der Prüfung wusste, war sie einfach zu lösen, aber wenn nicht, konnte man eine ganze Aufgabe vergessen. Dies finde ich doch sehr spitzfindig*
  - *Ich erwartete bei Info 2 weniger Beweise*
  - *Wenn man Laufzeitkomplexitäten nicht gut verstanden hat, war diese Prüfung sehr schwer zu lösen*



# Hinweise zur Prüfungsvorbereitung (8)

## Feedback nach der Prüfung Informatik I / II



# Hinweise zur Prüfungsvorbereitung (9)

- „Wo lagen die grössten **Unterschiede zu Ihren Erwartungen?**“
  - *In der Schwierigkeit der Aufgaben*
  - *Prüfung war einfacher als befürchtet*
  - *Prüfung zu Informatik 2 entsprach völlig meinen Erwartungen*
  - *Völlig anders bezüglich der letzten Jahre*
  - *Ich habe die Bäume unterschätzt*
  - *Schwerpunkte waren v.a. auf die letzten Themen der Vorlesung gelegt*
  - *Ich habe mehr Programmieraufgaben erwartet*
  - *Die Aufgaben waren auf den Stoff der Vorlesung abgestimmt und das fand ich sehr gut*
  - *Ich würde die Studierenden besser darauf aufmerksam machen, dass Programmieren in Java einen erheblichen Teil der Prüfung ausmacht*
  - *Es wurde Wert auf Aufgaben gelegt, auf die in der Vorlesung nur kurz hingedeutet wurden und nicht auf die Kernthemen*
  - *...Gefühl, dass ich mir das Lernen hätte sparen können, weil ich die Dinge die ich könnte auch ohne Lernen gekonnt hätte und die anderen sowieso nicht*

Hippokrates, der berühmte Arzt des Altertums, mahnte vor langer Zeit:

*Das Leben ist kurz  
Die Kunst lang  
Die Gelegenheit flüchtig  
Die Erfahrung trügerisch  
Das Urteil schwierig.*

Rund 2200 Jahre später tritt in Goethes Faust der etwas naive Famulus (also Hilfsassistent) Wagner „im Schlafrocke und der Nachtmütze“ auf, und unwissentlich zitiert er diese Aussage in Form eines stilistisch eleganten Stosseufzers:

*Ach Gott! Die Kunst ist lang,  
Und kurz ist unser Leben.*

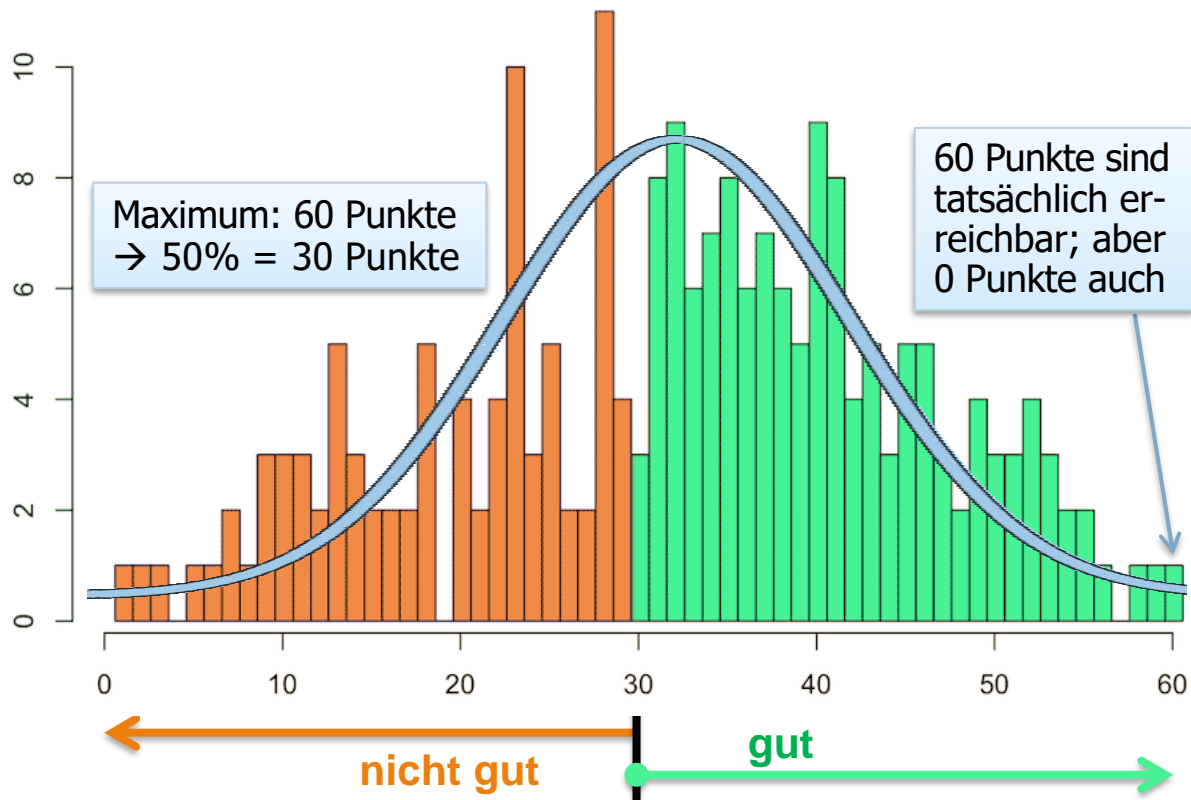
Wir stimmen in die Klage ein: Der Prüfungsstoff ist üppig, die Vorbereitungszeit kurz. Hippokrates, Goethe und Wagner werden zu Leidensgenossen.



**Ein guter Tipp zum Schluss:** Herauszufinden, ob man gut im Lösen von Prüfungsaufgaben ist, ist nicht der Prüfungszweck. (Alles klar?)

# Punkteverteilung der Prüfung Informatik II

## [Beispiel einer früheren Prüfung]



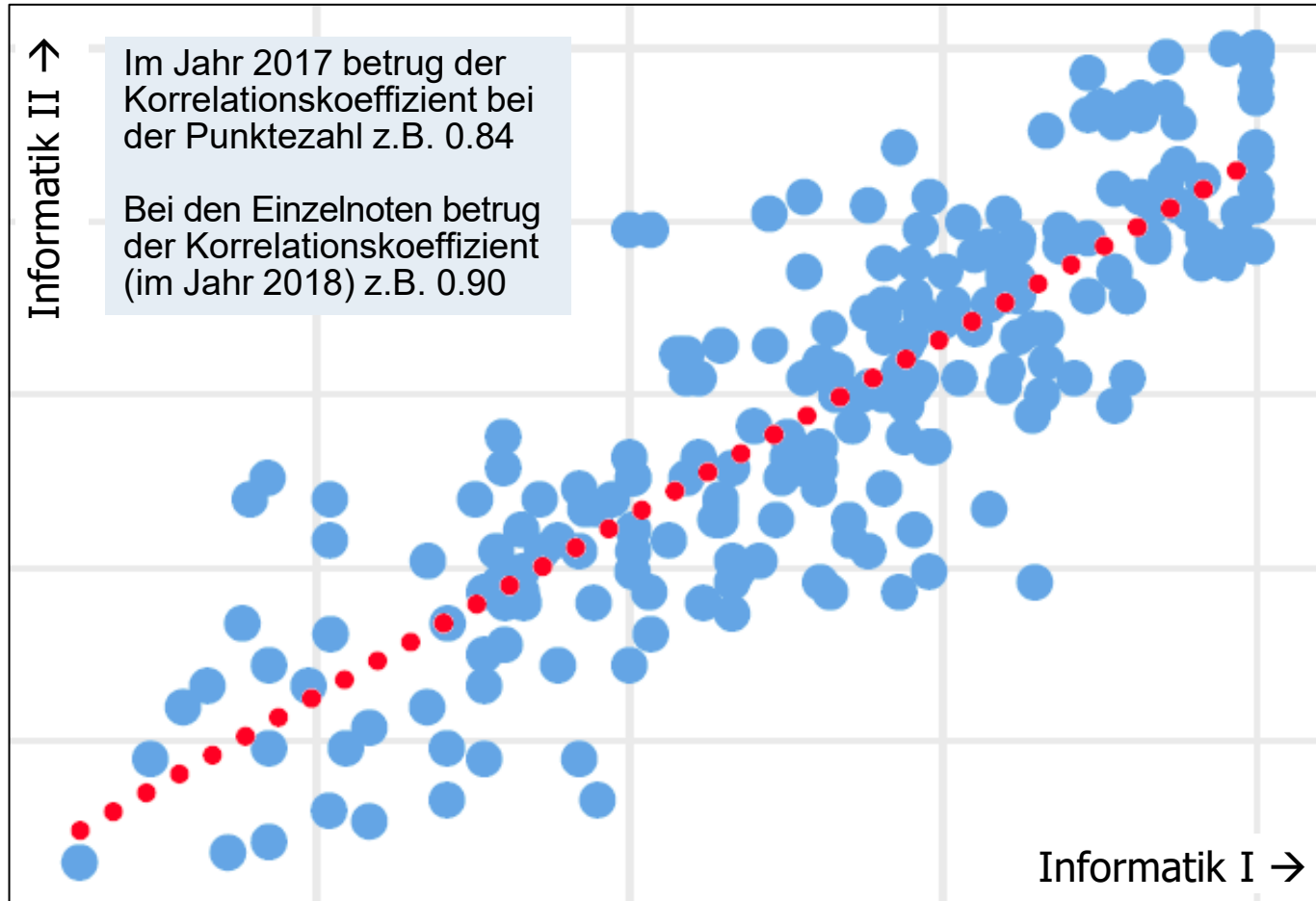
NORMAL DISTRIBUTION



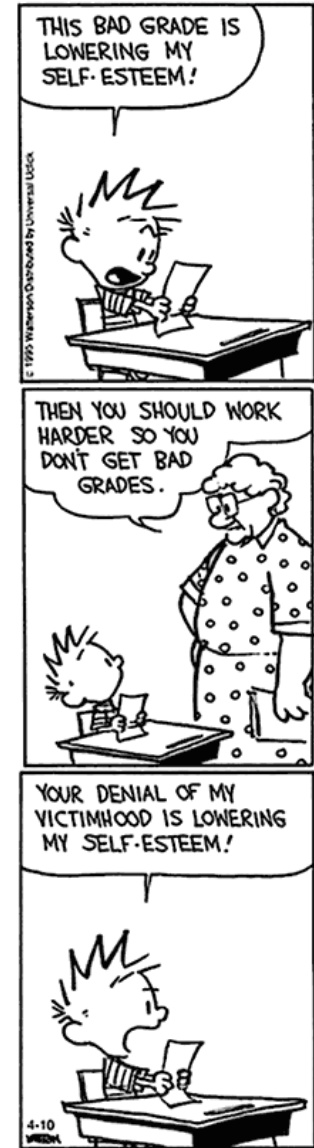
PARANORMAL DISTRIBUTION

# Korrelation Punkte Informatik I / Informatik II

## [Beispiel einer früheren Prüfung]



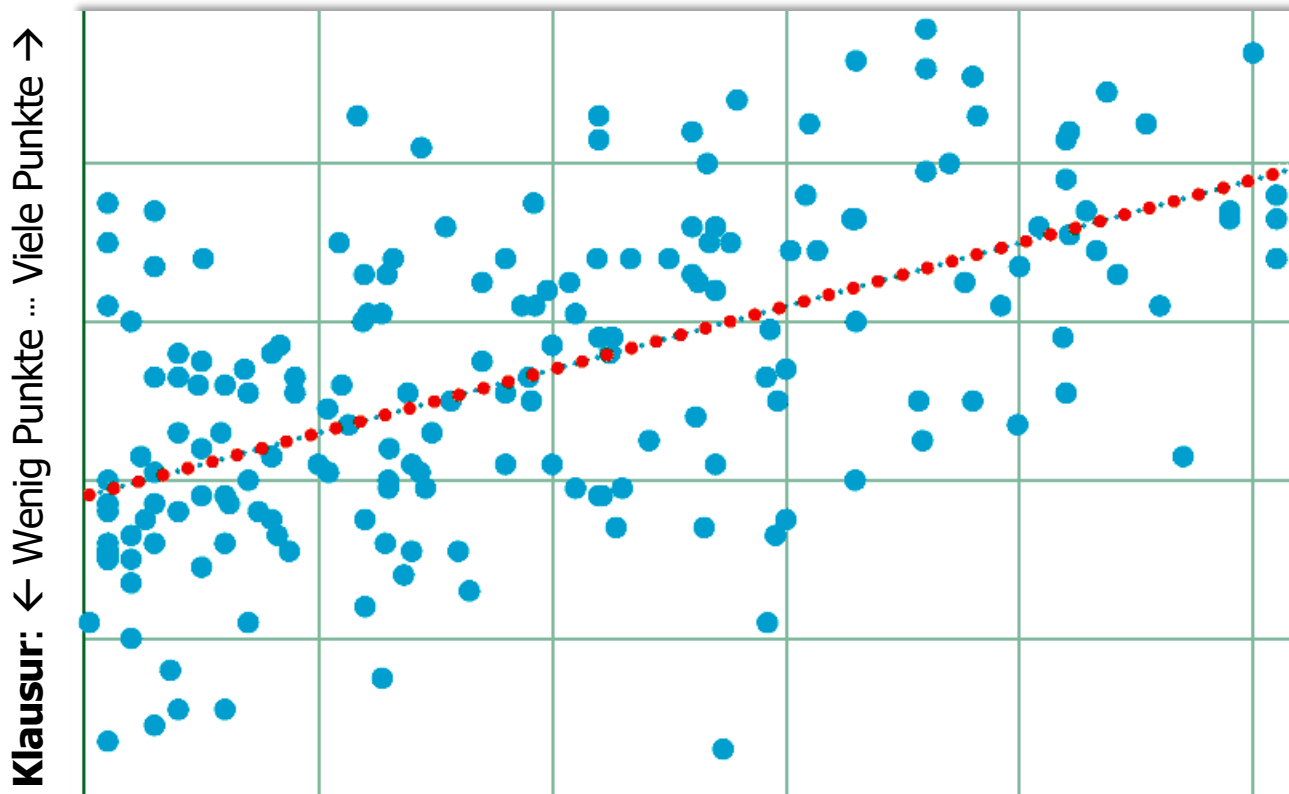
← Wenig Punkte (schlechte Note)    ...    Viele Punkte (gute Note) →



# Korrelation Punkte Übungen / Prüfungsklausur

## [Beispiel einer früheren Prüfung]

Für abgegebene Programmieraufgaben liefert das automatische Bewertungssystem einen „score“ entsprechend der Zahl der erfolgreich durchgeführten Tests. Der Korrelationskoeffizient beträgt hier 0.57 – lohnt sich also das Üben?



Patrick Winston (Prof. am MIT) bemerkte zu seiner Vorlesung dazu:

*We were pleased to see there was a positive correlation...*

*Of course, we reminded ourselves not to confuse correlation with cause. A likely explanation for the positive slopes is that those who take the subject more seriously are more engaged in general, and that more engaged means spending more time studying as well as dragging oneself out of bed for an early morning lecture.*

Codeboard-Score: ← Wenig Punkte ... Viele Punkte →

# Er schreibt in der Klausur... was sollt er alles wissen nur

An seinem Pult der Assistent  
guckt auf den stud. inf. permanent.  
Student! Er schreibt in der Klausur...  
was sollt er alles wissen nur.  
Er wünscht sich gänzlich unbeachtet,  
statt dass der Assi ihn betrachtet.  
Er fühlt so schwer des Blickes Bürde:  
„Wenn der mich durchfall'n lassen würde?“  
Ihm ist, als ob er kaum noch schriebe...  
wenn er auch diesmal sitzen bliebe?!  
Doch jener dreht nach einer Stunde  
den Kopf aus irgend einem Grunde,  
vielleicht auch ohne tiefern Sinn,  
(wer weiss) nach 'ner Studentin hin.

Aus: Karl Nickel (1924–2009, alias „KLEN“):  
*Palmström als Programmierer*; nach *Der  
Sperling und das Känguru* aus Christian  
Morgensterns *Palma Kunkel*

*In seinem Zaun das Känguru -  
es hockt und guckt dem Sperling zu.*

*Der Sperling sitzt auf dem Gebäude -  
doch ohne sonderliche Freude.*

*Vielmehr, er fühlt, den Kopf geduckt,  
wie ihn das Känguru beguckt.*

*Der Sperling sträubt den Federflaus -  
die Sache ist auch gar zu kraus.*

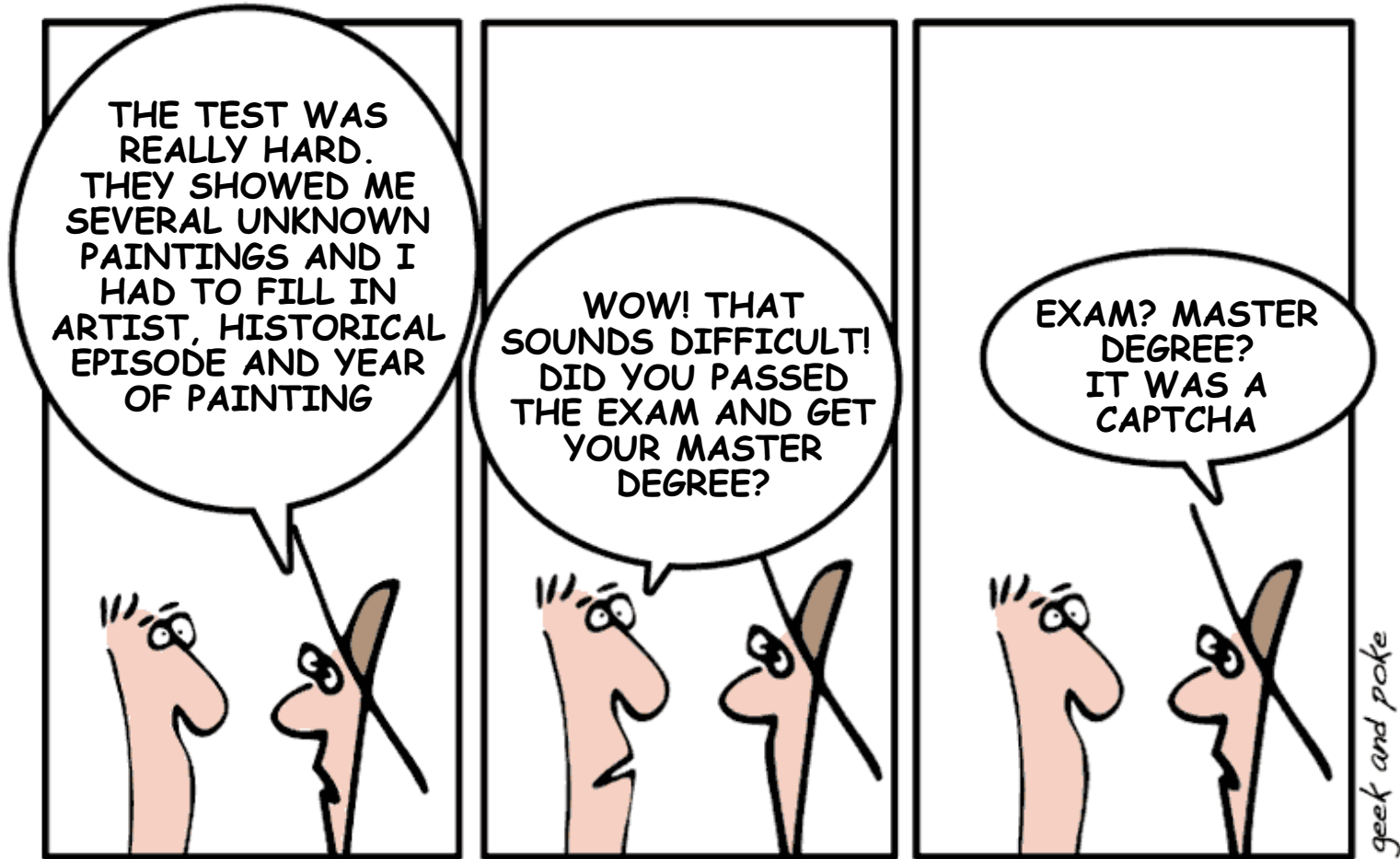
*Ihm ist, als ob er kaum noch sässe ...  
Wenn nun das Känguru ihn frässe?!*

*Doch dieses dreht nach einer Stunde  
den Kopf aus irgend einem Grunde,  
vielleicht auch ohne tiefern Sinn,  
nach einer andern Richtung hin.*

**Viel Erfolg bei  
der Prüfung!**



# The test was really hard

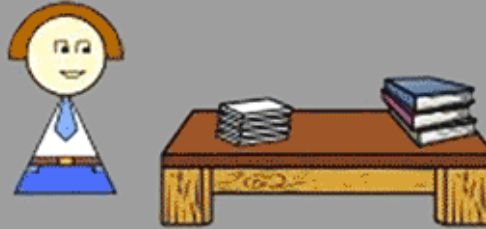


# Feedback zur Vorlesung

spikedmath.com © 2010

## TEACHING EVALUATIONS

OKAY, LET'S SEE WHAT THE STUDENTS WROTE ABOUT ME!



too much material



good amount of material



goes fast



goes slow



textbook is useless



textbook is a valuable resource



not an asshole



UH... I GUESS THAT IS A GOOD THING.



THANKS.

<http://spikedmath.com/comics/344-teaching-evaluations.png>

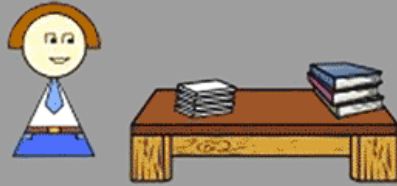


# Feedback (2)

spikedmath.com © 2010

## TEACHING EVALUATIONS

OKAY, LET'S SEE WHAT THE STUDENTS WROTE ABOUT ME!



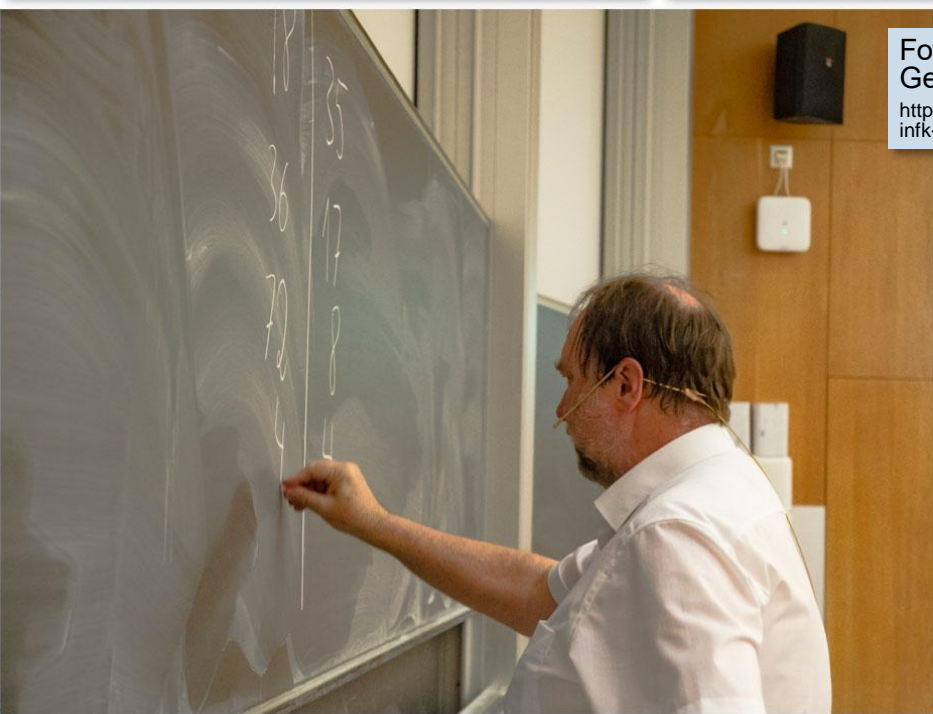
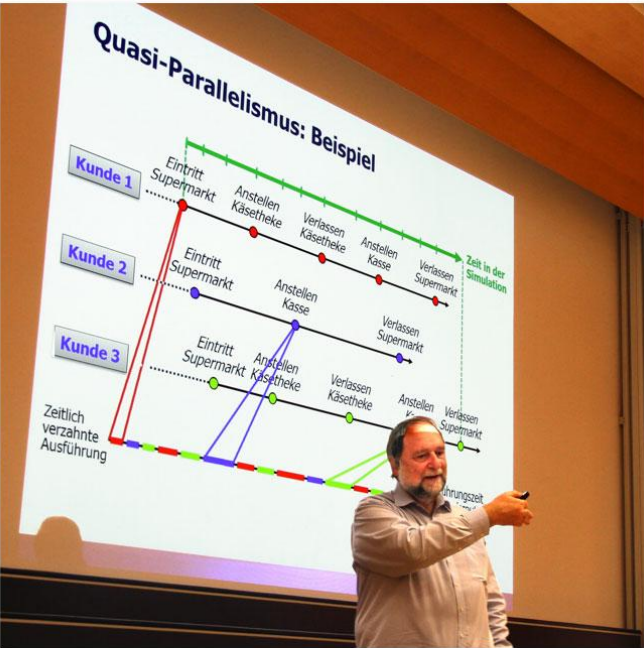
- Professor Deutscher aber ok.
- Mehr Frauen in der Vorlesung!
- Lehrbuch ist zu anstrengend.
- Benötigter Zeitaufwand stieg exponentiell von Serie zu Serie.
- Reversi ist ziemlich unnötig.
- Wenn man in jedem Fach so viel Zeit wie in Informatik investieren müsste, dann könnte man ja nie schlafen gehen.
- Der Professor tut oft so, als wäre das, was er da gerade erklärt, einfach. Dem ist aber überhaupt nicht so!
- Ich bin persönlich zwischen Info-I / Info-II verloren; es wäre besser, wenn der Stoff leichter + einfacher wäre.
- Ohne Google hätte ich keine Übung lösen können.
- Oft ein bisschen langweilig, da zu viel Theorie, z.B. über irgendeinen Algorithmus oder so.
- Es war sehr schade, dass gewisse Übungsstunden mitten am Nachmittag stattfanden, dadurch ist der ganze Nachmittag schon verbraucht.
- Reversi spannend und interessant.
- Ich weiss nicht so recht, was Spielbäume mit Elektrotechnik zu tun haben.

UH... I GUESS THAT IS A GOOD THING.



THANKS.

- Vorlesung ist nicht schlecht doch vollkommen unnötig.
- ...schaue ich keine Vorlesungen, benutze nicht das Skript und erlerne alles über Übungsstunden.
- Der Dozent erklärt sehr verständlich.
- Meistens werden einfachste Sachen übermässig erklärt, während die schwierigeren nur knapp erwähnt werden.
- ...Persönlichkeit macht die Vorlesung mega freundlich.
- Ich habe die ersten paar Vorlesungen geschaut danach jedoch aufgehört da ich die Vorlesung zum Lösen der Übungen nicht gebraucht habe.
- Grosses Dankeschön für die super Vorlesung! Das gewisse Extra an Randbemerkungen, Humor, Vorgeschichten, Stoff-Exkursen, Gedankenanstörungen usw. war in dieser Vorlesung hervorragend.
- Ich hätte nichts anzumerken, finde es gut so wie es ist.
- Die Bonusslides waren super - würde ich beibehalten.
- Die Folien könnten um ca. 90% gekürzt werden.
- Die Bonus-Slides helfen über den Tellerrand hinaus zu denken und Informatik im Kontext besser zu verstehen daher ist es gut dass sie Teil des Unterrichts sind.
- Dummerweise ist das Skript so gut, dass ich es nicht bemängeln kann.
- Meiner Meinung nach ist es die beste Vorlesung, die wir im Moment haben.
- Vorlesung ist von sehr schlechter Qualität. Dennoch machte Herr Mattern keine Anstalten, dies zu ändern.
- Sie können sehr gut erklären. Wenn man am morgen zwei Stunden aufmerksam zuhört und mitdenkt, sind komplizierte Dinge plötzlich sehr einfach. Praktisch in jeder Vorlesung kommen mehrere Aha-Erlebnisse.
- Der Dozent bot einen sehr anschaulichen, gut verständlichen Unterricht. Er versucht so zu erklären, als ob er den Stoff selbst zum ersten Mal hörte, was ausgezeichnet hilft, das Erklärte zu verstehen. Meiner Meinung nach beste Vorlesung des Semesters.
- Der Professor hat keine Ahnung.



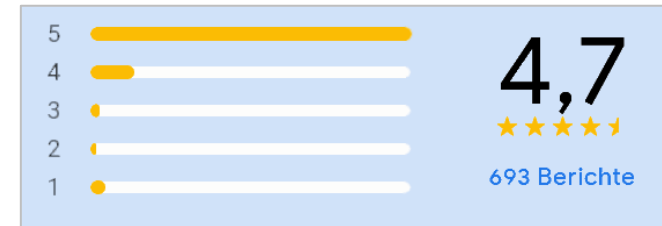
Fotos aus dem Artikel „Informatik ist auch Geschichte, Kultur und Gesellschaft“  
<https://inf.ethz.ch/de/news-und-veranstaltungen/spotlights/infk-news-channel/2020/12/critical-thinking-lecture.html>



# Feedback zur ETH

[www.google.ch/maps/place/Eidgen%C3%B6ssische+Technische+Hochschule+Z%C3%BCrich/@47.376658,8.547124,17.87z/data=!4m5!3m4!1s0x479aa0a6516e8cf7:0x4b61f428b6377607!8m2!3d47.3763888!4d8.5476281?hl=de](https://www.google.ch/maps/place/Eidgen%C3%B6ssische+Technische+Hochschule+Z%C3%BCrich/@47.376658,8.547124,17.87z/data=!4m5!3m4!1s0x479aa0a6516e8cf7:0x4b61f428b6377607!8m2!3d47.3763888!4d8.5476281?hl=de)

- \*\*\*\*\* Gut eingerichtete Universität. Prüfung war etwas schwierig für meinen Geschmack. Darum einen Stern Abzug.
- \*\*\*\*\* Tolle Universität mit schöner Aussicht.
- \*\*\*\*\* Hab meine Mütze verloren und am nächsten Tag in der Vorlesung wieder gefunden, tolle Gemeinschaft.
- \*\*\*\*\* Wer bereit ist, seine mentale Gesundheit und Nerven gegen eine top Ausbildung einzutauschen, der ist hier am richtigen Ort.
- \*\*\*\*\* Schlechte Atmosphäre, kein geeigneter Studienort. Kompliziert und viel Bürokratie. Auch das Essen ist sehr teuer. Besser man geht arbeiten und wird reich.
- \*\*\*\*\* Sehr schöne Universität. Angenehmer Arbeitsplatz und gutes Arbeitsklima.
- \*\*\*\*\* Sehr kompliziertes Gebäude mit eigensinnigen Studenten. Man ist eher eine Nummer als Mensch. Schade.
- \*\*\*\*\* Eindrückliche Gebäude, super Vorlesungen und den Studenten raucht der Kopf nach einer Woche Studium.
- \*\*\*\*\* Grossartige Bibliothek!
- \*\*\*\*\* 고풍스러운 건물ㅎ.
- \*\*\*\*\* I study computer science there. The building is gorgeous.
- \*\*\*\*\* 크리스마스연휴라 대학교가 문을 열지 않아서 아쉬웠습니다.
- \*\*\*\*\* Es un paraiso.
- \*\*\*\*\* No down side. ETH is on a hill.
- \*\*\*\*\* Great place for studying, amazing and clever people, a great respectful environment.
- \*\*\*\*\* Student center serves drinks, including wine and beer, to students and the visitors.
- \*\*\*\*\* For some unknown reason there was free wifi which was just what I needed at that point in time. Helpful students and the restroom was very clean.



# Feedback zur ETH (2)

1. Am Polytechnikum, o du mon Dieu, mon Dieu, da wird man steif und krumm, o du mon Dieu; :,: lernt manche schöne Theorie, braucht sie in seinem Leben nie. Sacre di bleu. :,:
2. Und im Kollegium, o du mon Dieu, mon Dieu, da sitzt man still und stumm, o du mon Dieu; :,: da wird gar mancherlei doziert, doch nur das wenigste kapiert! Sacre di bleu! :,:
3. Repetitorium, o du mon Dieu, mon Dieu, denk ich dein, wird's mir dumm, o du mon Dieu; :,: weiss man auch alles ganz gewiss, hat man doch oft erbärmlich Schiss. Sacre di bleu. :,:
4. Und hat man nicht studiert, o du mon Dieu, mon Dieu, ist man erst recht lackiert, o du mon Dieu; :,: steht da als wie ein Delinquent, höhnisch lacht dann der Repetent. Sacre di bleu! :,:

Aus dem Liederbuch zum Jubiläum „50 Jahre Eidgenössisches Polytechnikum“, 1905



Die letzte Strophe lautet:

*Geplagt ist der Student,  
o du mon Dieu, mon Dieu,  
Kreuzhimmeldonnensakrament,  
o du mon Dieu;  
der Teufel hol das Studium  
am Po- Po- Polytechnikum.*





Quelle unbekannt; evtl. „Teutsche“ im „stern“ ca. 1980.

This is not a publication. The material is intended for one-time educational use only. Pictures should not be copied or redistributed. No violations of copyrights are intended.